

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Київський національний університет будівництва і архітектури

АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМУВАННЯ

Методичні вказівки
до виконання лабораторних робіт
для студентів, які навчаються за напрямом підготовки
6.050101 «Комп'ютерні науки»

Київ 2015

УДК 681.3.06
ББК 32.973 – 01
А 45

Укладач Г.В. Красовська, канд. техн. наук, доцент

Рецензент О.В. Ізмайлова, канд. техн. наук, доцент

Відповідальний за випуск А.О. Білощицький, канд. техн. наук,
професор

Затверджено на засіданні кафедри інформаційних технологій,
протокол №7 від 24 листопада 2014 року.

Видається в авторській редакції.

Алгоритмізація та програмування: методичні вказівки до виконання
А 45 лабораторних робіт / уклад. Г.В. Красовська. – К.: КНУБА, 2015. – 64 с.

Містять загальні положення, методичні вказівки до виконання лабораторних робіт, рекомендації щодо розробки та налагодження програм, вимоги до оформлення звітів, список рекомендованої літератури.

Призначено для студентів, які навчаються за напрямом підготовки 6.050101 «Комп'ютерні науки» під час вивчення дисципліни «Алгоритмізація та програмування».

ЗМІСТ

Загальні положення	4
Вимоги до оформлення звітів з лабораторних робіт.....	4
ЛАБОРАТОРНА РОБОТА № 1. Програмування лінійного обчислювального процесу.....	5
ЛАБОРАТОРНА РОБОТА № 2. Програмування розгалуженого обчислювального процесу	11
ЛАБОРАТОРНА РОБОТА № 3. Програмування циклічного обчислювального процесу.....	26
ЛАБОРАТОРНА РОБОТА № 4. Програми з підпрограмами.....	34
ЛАБОРАТОРНА РОБОТА №5. Обробка масивів даних	40
ЛАБОРАТОРНА РОБОТА №6. Обробка записів (структур) даних.....	49
ЛАБОРАТОРНА РОБОТА №7. Обробка типу множина мови Паскаль.....	54
ЛАБОРАТОРНА РОБОТА №8. Динамічний розподіл пам'яті.....	57
Рекомендована література	63

Загальні положення

Дисципліна «Алгоритмізація та програмування» («Основи програмування та алгоритмічні мови») є базовою професійно-орієнтованою дисципліною підготовки бакалаврів за напрямком 6.050101 «Комп'ютерні науки». Вивчення цієї дисципліни передбачено на першому курсі в першому та другому семестрах.

Методичні вказівки до виконання лабораторних робіт складені відповідно до робочої програми навчальної дисципліни «Алгоритмізація та програмування» і є невід'ємною частиною методичного комплексу, що включає конспекти лекції [9; 10], методичні вказівки до вивчення дисципліни [12], до практичних робіт [13], виконання індивідуальних завдань та курсової роботи [14].

Дане видання методичних вказівок є оновленим та доповненим виданням [11], яке розроблялося протягом багатьох років на кафедрі інформаційних технологій під час викладання дисципліни «Алгоритмізація та програмування» та «Основи програмування та алгоритмічні мови» для студентів напрямку підготовки КН.

При розробці варіантів завдань, що наводяться до кожної лабораторної роботи, використовувалися посібники [1; 2; 3; 5]. Завдання сформульовані таким чином, що можуть виконуватися із застосуванням різних мов програмування та середовищ розробки програм, але приклади виконання окремих завдань наведені у порівнянні мовами Паскаль та Сі.

Вимоги до оформлення звітів з лабораторних робіт

Звіт з лабораторної роботи повинен мати в своєму складі:

1. Завдання до лабораторної роботи та варіант завдання;
2. Математичну постановку задачі, яка містить:
 - опис вхідних та вихідних даних (змінних), їх типів та у разі необхідності опис обмежень на дані;
 - математичну модель задачі та опис проміжних змінних;
 - схему алгоритму з коментарями до неї;
3. Тестовий приклад для перевірки правильності та повноти отриманого рішення (необхідно навести варіант вирішення задачі вручну та програмні розрахунки);
4. Аналіз результатів роботи програми;
5. Опис програмного інтерфейсу з користувачем;
6. Текст(и) програм(и).

ЛАБОРАТОРНА РОБОТА № 1.

Програмування лінійного обчислювального процесу

Завдання 1.1

Скласти програму для обчислення математичного виразу згідно з варіантом завдання¹.

Рекомендації до виконання роботи

1. Для програмованого обчислення виразів здебільшого випадків необхідно використовувати математичні функції зі стандартних бібліотек мов програмування. Якщо в бібліотеці немає необхідної математичної функції, треба її виразити через ті, що визначені. Наприклад, стандартом мови Паскаль не передбачена функція $\arccos(x)$, але існує $\arctg(x)$. За тригонометричною формулою

$$\arccos x = \arctg \frac{1}{\sqrt{1-x^2}}, \quad (1)$$

де $|x| \leq 1$.

Стандартом мови Паскаль не передбачена операція піднесення до степеня x^a , де $a > 2$. Для обчислення степеня можна використати формулу:

$$x^a = e^{a \cdot \ln x}. \quad (2)$$

Обчислення логарифмів:

$$\log_a x = \frac{\ln x}{\ln a}. \quad (3)$$

2. Якщо вираз складний, або в ньому є частини, що повторюються, для спрощення запису цього виразу необхідно ввести проміжні змінні. Наприклад:

$$y = \frac{\sqrt{x^2 + 4y^2}}{\arctg \frac{x}{(x^2 + 4y^2)^3}}.$$

В заданому виразі частина $x^2 + 4y^2$ повторюється двічі, тому доцільно ввести додаткову проміжну змінну :

$$t = x^2 + 4y^2.$$

Вхідний вираз набуває вигляду:

$$y = \frac{\sqrt{t}}{\arctg \frac{x}{t^3}}$$

¹ варіант завдання обирається студентом з наведеного нижче списку «Варіанти завдань 1.1» відповідно до номеру в списку академічної групи.

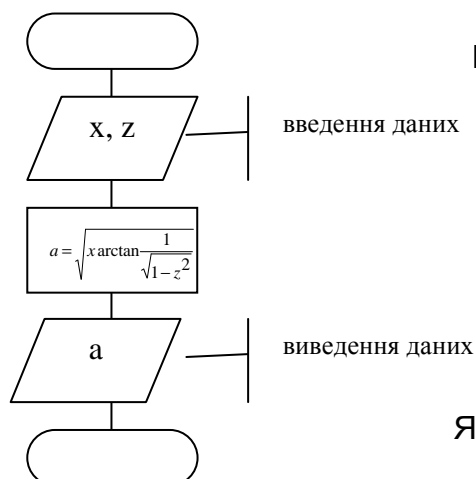
Приклад виконання роботи

Завдання: скласти програму для обчислення

$$a = \sqrt{x \arccos(z)}.$$

1. Математична постановка задачі.

Вхідні дані	Дії	Вихідні дані
x, z дійсного типу. Обмеження: $ z \leq 1; x \geq 0$.	$a = \sqrt{x \arctan \frac{1}{\sqrt{1-z^2}}}$	a – дійсне.



Враховуючи те, що мовою Паскаль не передбачено обчислення функція \arccos , здійснено перетворення даного виразу за формулою 1:

$$a = \sqrt{x \arctan \frac{1}{\sqrt{1-z^2}}}$$

2. Тестовий приклад.

Як приклад розглянемо 3 випадки:

- ♦ $x=0.5$, $z=-0.7$;
 $a = \sqrt{0.5 \arccos(-0.7)} = \sqrt{0.5 \cdot 2.34619} = 1.08395$
- ♦ $x=2.5, z=0.5$; $a = \sqrt{2.5 \arccos(0.5)} = \sqrt{2.5 \cdot 1.0472} = 1.618022$
- ♦ $x=7, z=0$; $a = \sqrt{7 \arccos(0)} = \sqrt{7 \cdot 1.5708} = 3.315958$.

3. Аналіз результатів роботи програми.

Вхідні дані	Вихідні дані		
	Розрахунок вручну	Паскаль	Сі
$x=0.5, z=-0.7$	1.08395	1.0839	1.0840
$x=2.5, z=0.5$	1.618022	1.6180	1.6180
$x=7, z=0$	3.315958	3.3160	3.3160

Похибка, що отримана під час програмного розрахунку не перебільшує 0.001, тому можна вважати програмні розрахунки правильними.

4. Опис програмного інтерфейсу з користувачем.

Після запуску програми на екран виводиться повідомлення-запрошення про введення даних. Необхідно ввести значення x та z , враховуючи обмеження на дані, що визначені в математичній постановці задачі.

Після розрахунку на екран виводиться значення a .

Для завершення роботи програми необхідно натиснути будь-яку клавішу на клавіатурі.

5. Тексти програм мовами Паскаль та Сі.

<pre> Uses Crt; Var a, x, z : real; begin ClrScr; Write('Введіть значення змінних x, z : '); ReadLn(x, z); a:=Sqrt(x*arctan(1/sqrt(1-sqr(x)))); WriteLn('a= ', a:9:4) ; readkey; End.</pre>	<pre> #include <stdio.h> #include <conio.h> #include <math.h> main() { float a, x, z; clrscr(); printf("Введіть значення змінних x, z: "); scanf("%f%f", &x, &z); a= sqrt(x*acos(z)); printf("\n a= %9.4f \n", a); getch(); }</pre>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Варіанти завдань 1.1

$$1. z = \frac{\sqrt{|x-1|} - \sqrt[3]{y}}{1 + \frac{x^2}{2} + \frac{y^2}{4}}$$

$$2. a = x(\arcsin \frac{r}{r^2+1} + d^{(x+3)})$$

$$3. a = \frac{3 + e^y - 1}{1 + x^2 |y - \operatorname{tg}(z)|}$$

$$4. b = \frac{\operatorname{tg}(cx^3 + dx^2 - x)}{e^{(x^2 - 3x - |cd|)} / \operatorname{ctg}(x)}$$

$$5. a = \sqrt{\frac{|\sin 8h| + 17}{\left(1 - \sin 4h \cos(h^2 + 18)\right)^2}}$$

$$6. b = \frac{1 + \sin^2(x + y)}{2 + \left| \frac{x - 2x}{(1 + x^2 y^2)} \right|}$$

$$7. a = (1 + y) \frac{x + \frac{y}{x^2 + 1}}{e^{-x-2} + \frac{1}{x^2 + 1}}$$

$$8. b = 1 + |y - x| + \frac{(y - x)^2}{2} + \frac{(y - x)^3}{3}$$

$$9. a = y + \frac{x}{y^2 + \left| \frac{x^2}{y + \frac{x^3}{3}} \right|}$$

$$10. a = e^{x^2} + \operatorname{tg}^2 \frac{\log z}{2^x}$$

$$11. a = 1 + \frac{z^2}{3 + z^2/5}$$

$$12. a = \cos^2(\operatorname{arctg}^{\log_a b} / z^x)$$

$$13. a = \ln \left| \left(y - \sqrt{|x|} \right) \left(x - \frac{y}{z + \frac{x^2}{4}} \right) \right|$$

$$14. a = x - \frac{x^2}{3!} + \frac{x^5}{5!}$$

$$15. a = \frac{\sin^3 |cx^3 + dx^2 - cd|}{\sqrt{(cx^3 + dx^2 - cd)}}$$

$$16. b = \frac{1 + \cos(y - 2)}{\frac{x^4}{2} + \sin^2 z}$$

$$17. a = \frac{2 \cos(x - \frac{\pi}{6})}{\frac{1}{2} + \sin^2 x}$$

$$18. a = 1 - \sqrt{\frac{3}{3 + |tgah^2 - \sin ah|}}$$

$$19. a = \frac{\left(\frac{\sin x \cos y}{\operatorname{arctg} \frac{z}{z}} \right)^2}{\frac{\operatorname{tg} x + y^2}{\sqrt{z^3}}}$$

$$20. a = \frac{2 \cos x + \sqrt{3} \sin 4a - 1}{2 \sin a^2 + \cos 42 - e^{\sqrt[3]{x}}}$$

$$21. a = \frac{\frac{x^2 + 3}{\sin x} + 3(x + y)^4 + \sqrt{b}}{\operatorname{tg}(x + y) + |x|}$$

$$22. a = \frac{4x^3 + \cos^2 x - \sin x^3}{\log_2 x + 2\sqrt{3}}$$

$$23. a = \frac{\cos x^2 + \sin^2 x}{e^{5x} + \lg 5 + \left| (x - e^{2x})^3 \right|}$$

$$24. a = \frac{(2x - 3)(7 + 12b)^2 + x}{\cos x \sqrt{b} - |x| + \log_3 4x^4}$$

$$25. a = \frac{4x^3 + \sqrt[8]{2y} - \sin^3 3c}{(x + \sqrt{y})^2 (\sqrt{3d} - e^{4x})}$$

Завдання 1.2

Розробити математичну постановку задачі та скласти програму для розрахунку значень згідно з варіантом завдання.

Приклад виконання роботи

Завдання: скласти програму для визначення площі та периметра прямокутника.

1. Математична постановка задачі.

Для визначення площі та периметра прямокутника необхідно знати довжини двох його сторін. Введемо позначення:

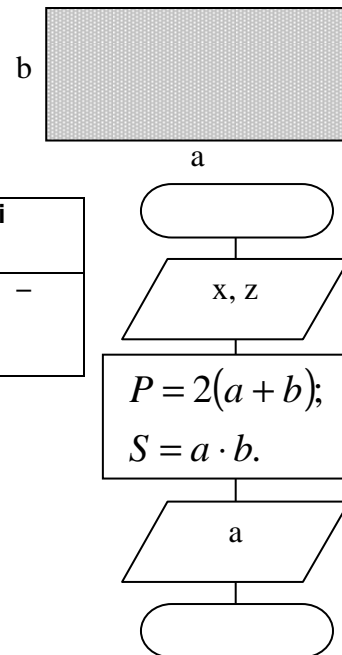
a – довжина прямокутника, задається в метрах (м);

b – ширина прямокутника (м);

P – периметр прямокутника (м);

S – площа прямокутника (m^2).

Вхідні дані	Дії	Вихідні дані
a, b дійсного типу. Обмеження: $a, b > 0$.	$P = 2(a + b);$ $S = a \cdot b.$	P, S – дійсні.



2. Тестовий приклад.

Як приклад розглянемо 3 випадки:

1. $a=5$ м, $b=10$ м. $P=30.0$ м, $S=50.0$ m^2 .

2. $a=0.5$ м, $b=0.7$ м. $P=2.4$ м, $S=0.35$ m^2 .

3. $a=3$ м, $b=1.5$ м. $P=9.0$ м, $S=4.5$ m^2 .

3. Аналіз результатів роботи програми.

Вхідні дані	Вихідні дані					
	Вручну		Паскаль		Сі	
	P	S	P	S	P	S
$a=5, b=10$	30.0	50.0	30.0	50.0	30.0	50.0
$a=0.5, b=0.7$	2.4	0.35	2.4	0.35	2.4	0.35
$a=3, b=1.5$	9.0	4.5	9.0	4.5	9.0	4.5

Результати розрахунків співпадають, похибки не виникло, тому можна вважати програмні розрахунки правильними.

4. Опис програмного інтерфейсу з користувачем.

Після запуску програми на екран виводиться повідомлення-запрошення про введення сторін прямокутника. Необхідно ввести значення довжини a прямокутника та ширини b , що виражені в метрах.

Після розрахунку на екран виводяться значення P, S .

Для завершення роботи програми необхідно натиснути будь-яку клавішу на клавіатурі.

5. Текст програм мовами Паскаль та Сі.

Uses Crt;	#include <stdio.h>
-----------	--------------------

<pre> Var a, b, P, S: real; begin ClrScr; Write('Введіть довжину a (м) та ширину b (м): '); ReadLn(a, b); P:=(a+b)*2; S:=a*b; WriteLn('Периметр = ', P:9:4, 'Площа = ', S:9:4); readkey; End.</pre>	<pre> #include <conio.h> main() { float a, b, P=0, S=0; clrscr(); printf("Введіть довжину a (м) та ширину b (м) : "); scanf("%f%f", &a, &b); P=(a+b)*2; S=a*b; printf("\n Периметр = %9.4f \n Площа = %9.4f \n", P, S); getch(); }</pre>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Варіанти завдань 1.2

Скласти програму для обчислення:

1. Координат точки перетину двох прямих;
2. Площі підлоги та площі стін кімнати;
3. Перерахунку координат точки з полярної системи до декартової;
4. Площі поверхні та об'єму кулі;
5. Площі поверхні та об'єму конуса;
6. Площі поверхні та об'єму циліндра;
7. Площі поверхні та об'єму прямокутного паралелепіпеда;
8. Площі та периметру трикутника за заданими довжинами сторін;
9. Площі та довжини кола заданого радіусу;
10. Площі та периметра прямокутного трикутника за заданими катетами;
11. Площі та периметра рівнобічної трапеції;
12. Площі та периметра прямокутної трапеції;
13. Площі та периметра паралелограма за заданими двома сторонами та кутом між ними;
14. Визначення за заданими x та a значень x^a , x^{-a} , $\sqrt[a]{|x|}$;
15. Швидкості V_1 та V_2 двох потягів, які почали рухатися на зустріч один одному з відстані S , якщо відомо, що швидкість першого потяга більше на $V_{\text{км/годину}}$, а зустрілися вони через t годин;
16. Катетів прямокутного трикутника за заданими довжиною гіпотенузи та кутом;
17. Довжин гіпотенузи та катета за заданими довжиною іншого катета та прилеглим до нього кутом;
18. Довжини гіпотенузи та двох кутів за заданими довжинами катетів;
19. Площі та довжини дуги сектора за заданим радіусом та кутом;
20. Площа та довжини хорди сегмента за заданим радіусом кола та кутом;
21. Координат центру відрізка за заданими координатами кінців;
22. Координат центру вписаного в квадрат кола, якщо відомі координати лівої верхньої та правої нижньої вершини квадрата;
23. Довжини сторін прямокутника, якщо відомі координати його лівої верхньої та правої нижньої вершини;
24. Довжини основи та площі рівнобічного трикутника за заданими довжиною бокової сторони та кута між нею та основою;
25. Площі та довжини сторони ромба за заданими довжинами його діагоналей.

ЛАБОРАТОРНА РОБОТА № 2.

Програмування розгалуженого обчислювального процесу

Завдання 2.1

- ◆ Скласти програму для визначення значення y згідно з варіантом завдання;
- ◆ після налагодження програми виконати *трасування програми*, задавши набір вхідних даних для перевірки «істинної» та «хибної» гілок оператора умови;
- ◆ встановити *точку останову* на операторі if;
- ◆ відкрити вікно для перегляду значення змінних програми (*watches*);
- ◆ повторно запустити програму на виконання.

Приклад виконання роботи

Завдання: скласти програму для обчислення:

$$y = \begin{cases} x^2, & \text{для } 0 \leq x \leq 5 \\ \cos x & \text{у протилежному випадку} \end{cases}$$

1. Математична постановка задачі(МПЗ).

Вхідні дані	Дії	Вихідні дані
x дійсного типу.	Якщо виконується $0 \leq x \leq 5$, тоді $y = x^2$, інакше $y = \cos x$	y – дійсне.

2. Тестовий приклад.

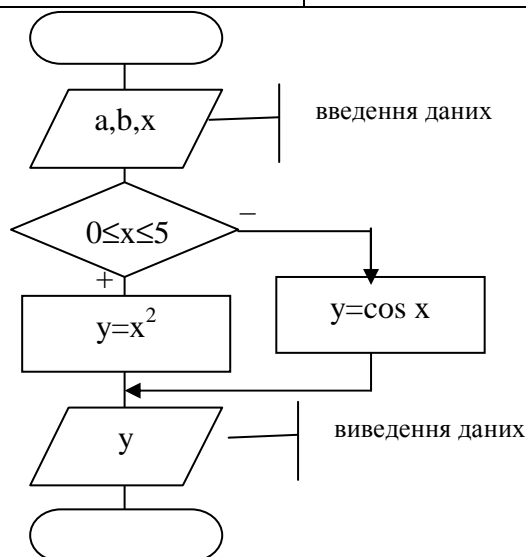
Для перевірки правильності МПЗ та складеної програми розглянемо такі випадки:

1. Умова $a \leq x \leq b$ виконується.

- ◆ $x=3 \rightarrow y=9$;
- ◆ $x=0 \rightarrow y=0$;
- ◆ $x=5 \rightarrow y=25$;

2. Умова $a \leq x \leq b$ не виконується.

- ◆ $x=-4 \rightarrow y=-0,653644$;
- ◆ $x=9 \rightarrow y=-0,911130$;



3. Аналіз результатів роботи програми.

Вхідні дані	Розрахунок вихідних даних		
	вручну	Паскаль	Cі
$x=3$	9	9.0	9.0
$x=0$	0	0.0	0.0
$x=5$	25	25.0	25.0
$x=-4$	-0,653644	-0,653644	-0,653644
$x=9$	-0,911130	-0,911130	-0,911130

По всіх тестових прикладах розрахунки, що були отримані вручну та програмно співпадають, це доводить правильність роботи програми.

4. Опис програмного інтерфейсу з користувачем.

Після запуску програми на екран виводиться повідомлення-запрошення про введення значення x . Необхідно ввести дійсне значення x . Обмежень на значення x не встановлюється.

Після розрахунку на екран виводиться значення y .

Для завершення роботи програми необхідно натиснути будь-яку клавішу на клавіатурі.

5. Текст програм мовами Паскаль та Сі.

Примітка! Для обчислення в програмах мовою Паскаль степеневих функцій необхідно перетворювати вираз (див. Л.р.№1, з.1.1.). Виникає обмеження на значення основи степеня (більше за 0), що вимагає додаткових перевірок та перетворень вхідного виразу. Додаткові перевірки також необхідні при обчисленні значення функцій tg та ctg . Враховуючи це алгоритм роботи програми буде відрізнятися від алгоритму розрахунку в математичній постановці задачі. Необхідно навести цей алгоритм в звіті.

<pre> Uses Crt; Var x, y : real; begin ClrScr; Write('Введіть значення x : '); ReadLn(x); if (x>=0) and (x<=5) then y=sqr(x) else y=cos(x); WriteLn('y= ', y:9:6) ; readkey; End.</pre>	<pre> #include <stdio.h> #include <conio.h> #include <math.h> main() { float x, y; clrscr(); printf("Input x:"); scanf("%f", &x); if (x>=0 && x<=5) y=pow(x, 2); else y=cos(x); printf("\ny=%9.6f", y); getch(); }</pre>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Варіанти завдань 2.1

$$1. \quad y = \begin{cases} x^{4\cos c}, & \text{якщо } -10 \leq x \leq 10; \\ \cos \pi x^2 & \text{інакше.} \end{cases}$$

$$2. \quad y = \begin{cases} e^{ax^2}, & \text{якщо } x \leq 0, a > 0; \\ \log_a x & \text{інакше.} \end{cases}$$

$$3. \quad y = \begin{cases} \arccos x, & \text{якщо } 0 < x \leq 1; \\ x^2 - \operatorname{tg} \pi x^2 - 1 & \text{інакше.} \end{cases}$$

$$4. \quad y = \begin{cases} \frac{\log_x k}{x^2 - k}, & \text{якщо } x > k, k > 0; \\ \sqrt{|xk|} & \text{інакше.} \end{cases}$$

$$5. \quad y = \begin{cases} |x^b|, & \text{якщо } -5 \leq x \leq -2; \\ x^6 + \cos b & \text{інакше.} \end{cases}$$

$$6. \quad y = \begin{cases} (\operatorname{tg} a - d)^2, & \text{якщо } d > a, a \leq 2 \\ \cos(ad)^3 & \text{інакше.} \end{cases}$$

$$7. \quad y = \begin{cases} \log_n x, & \text{якщо } n \neq 0, x \geq 0; \\ |x^{n+2}| & \text{інакше.} \end{cases}$$

$$8. \quad y = \begin{cases} \sin px, & \text{якщо } 0 < x \leq \pi; \\ \operatorname{ctg} px & \text{інакше.} \end{cases}$$

$$9. \quad y = \begin{cases} x^{2z} + z^x, & \text{якщо } -5 \leq x \leq 5; \\ \sqrt{e^z} - 2x & \text{інакше.} \end{cases}$$

$$10. \quad y = \begin{cases} a^{\sin^2 x}, & \text{якщо } -5 \leq x \leq 0; \\ a + x \cos a^2 & \text{інакше.} \end{cases}$$

$$11. \quad y = \begin{cases} x^{2a} + |a - 12|, & \text{якщо } a > 0, x < 0; \\ \frac{e^x}{\sqrt{1+a^2}} & \text{інакше.} \end{cases}$$

$$12. \quad y = \begin{cases} \cos^2 x + 2b, & \text{якщо } 0 < x < 2; \\ b - \operatorname{tg} x^2 & \text{інакше.} \end{cases}$$

$$13. \quad y = \begin{cases} a \operatorname{ctg}^2 x, & \text{якщо } 0 < a < 1; \\ a - a \cdot \cos^2 x & \text{інакше.} \end{cases}$$

$$14. \quad y = \begin{cases} \max(a, b), & \text{якщо } x > 0, a > 0; \\ \min(a, b) & \text{інакше.} \end{cases}$$

$$15. \quad y = \begin{cases} \frac{\log_x 10}{|x^2 + 3|}, & \text{якщо } -9 < x \leq 9; \\ e^{x^3} & \text{інакше.} \end{cases}$$

$$16. \quad y = \begin{cases} x^2 - \lg a, & \text{якщо } -2 \leq x \leq 2; \\ x^{\cos a^5} & \text{інакше.} \end{cases}$$

$$17. \quad y = \begin{cases} \operatorname{ctg} x + x^3, & \text{якщо } -3 < x < 3; \\ \cos x^2 + 3^x & \text{інакше.} \end{cases}$$

$$18. \quad y = \begin{cases} e^{-ax}, & \text{якщо } -7 \leq x < 0; \\ \sin^3 x^2 & \text{інакше.} \end{cases}$$

$$19. \quad y = \begin{cases} \frac{2-x}{\sqrt{|x+1|}}, & \text{якщо } a \leq x \leq b; \\ 10^7 \sin^5 ax & \text{інакше.} \end{cases}$$

$$20. \quad y = \begin{cases} \frac{\sin x}{\sqrt[7]{z^2 + 16}}, & \text{якщо } 0 < x \leq z; \\ bx^2 - zx & \text{інакше.} \end{cases}$$

Завдання 2.2

- ◆ Скласти програму з використанням оператора case (switch);
- ◆ встановити точку останову на операторі case (switch);
- ◆ виконати трасування програми з різними вхідними даними, відкривши вікно для перегляду поточного значення змінної-перемикача.

Рекомендації до виконання роботи

Як вже було вказано в завданні, при виконанні цієї роботи необхідно використовувати оператор множинного розгалуження (в мові Паскаль – оператор case – of – else, в мові Сі – оператор switch – case - default).

При написі програми на одній гілці оператора множинного розгалуження можна об'єднати декілька констант. При цьому в Паскаль можна задати діапазон цілочисельних або символьних значень [1; 2; 4-8; 9; 10; 13].

Бажано при написі програми використовувати також гілку else (default) для перевірки та видачі повідомлення про «неможливі» значення вхідних даних, навіть якщо це не передбачено в умові варіанта завдання.

Приклад виконання роботи

1. Скласти програму для виведення на екран повідомлення “молодша школа”, “середня школа”, “старша школа” залежно від заданого номера класу.
 1. Математична постановка задачі(МПЗ).

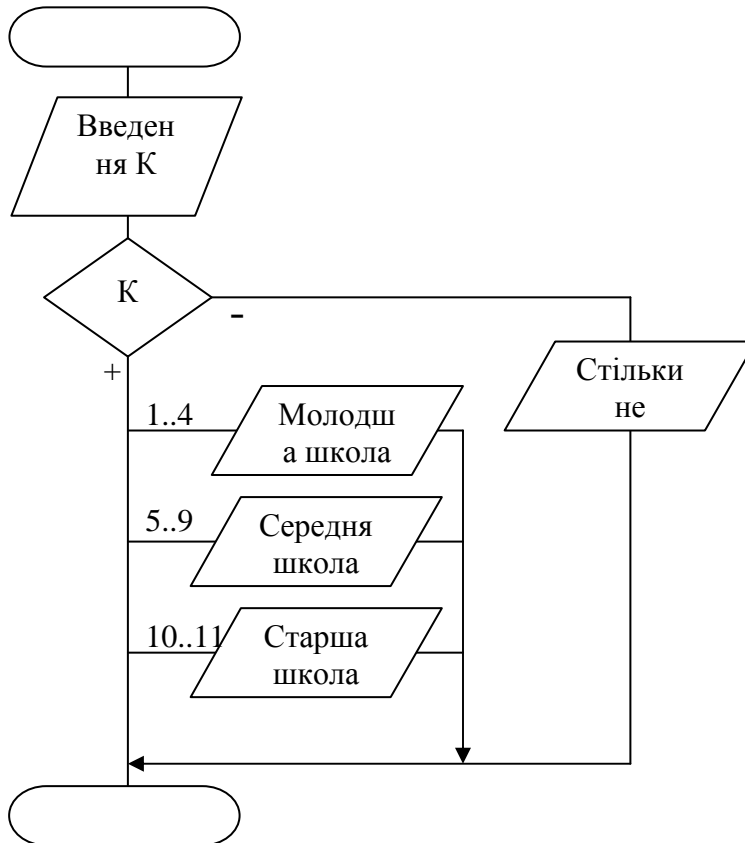
Вхідні дані	Дії	Вихідні дані
2. K – ціле	5. Якщо K співпадає з:	7.
3. Обмеження:	6. 1..4, тоді вивести	8. молодша школа
4. K>0.	9. 5..9, тоді вивести	10. середня школа
	11. 10, 11, вивести	12. старша школа
	13. інакше вивести	14. стільки не вчаться!!!

2. Тестовий приклад та аналіз результатів роботи програми.

Вхідні дані	Розрахунок вихідних даних	
	Паскаль	Сі
K=2	15. молодша школа	16. молодша школа
K=7	17. середня школа	18. середня школа
K=10	19. старша школа	20. старша школа
K=22	21. стільки не вчаться!!!	22. стільки не вчаться!!!

По всіх тестових прикладах отримані коректні вихідні повідомлення.

2. Опис програмного інтерфейсу з користувачем.



Після запуску програми на екран виводиться повідомлення-запрошення про введення значення К. Необхідно ввести цілочисельне додатне значення К. Після цього на екран виводиться повідомлення про шкільний рівень.

Для завершення роботи програми необхідно натиснути будь-яку клавішу на клавіатурі.

3. Текст програм мовами Паскаль та Сі.

<pre> Uses Crt; Var K:byte; begin ClrScr; Write('Введіть клас : '); ReadLn(K); case K of 1..4: WriteLn('молодша школа'); 5..9: WriteLn('середня школа'); 10..11: WriteLn('старша школа'); else WriteLn('стільки не вчаться!!!'); end; </pre>	<pre> #include <stdio.h> #include <conio.h> main() { short k; clrscr(); printf("Введіть клас :"); scanf("%hd", &k); switch (k) { 24. case 1:case 2:case 3:case 4: printf("молодша школа "); break; 27. case 5:case 6:case 7:case 8:case 9: printf("середня школа "); break; 29. case 10:case 11: printf("старша школа "); break; 30. default: printf("стільки не вчаться!!!"); } } </pre>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

```
readkey;  
End.
```

```
getch();  
}
```

Варіанти завдань 2.2

1. За заданим номером місяця вивести кількість днів в ньому.
2. За заданим номером місяця вивести номер кварталу.
3. За заданим номером місяця вивести пору року.
4. За заданим номером місяця вивести перелік державних свят в ньому.
5. За заданим номером дня в тижні вивести його назву.
6. За заданим номером дня в тижні вивести повідомлення „робочий, вихідний”.
7. За заданим номером кварталу вивести перелік місяців.
8. За заданим номером дня в тижні вивести кількість пар за розкладом вашої групи.
9. Ввести пергу букву з ім'я людини, вивести перелік з трьох можливих імен. Наприклад, введено 'а', виводиться –“Андрій, Анна, Архип”.
10. В одному елітному будинку 20 квартир чотирьох типів: однокімнатна, двокімнатна, трикімнатна, чотирикімнатна. Було складено перелік:

№ квартири	Тип квартири
1,2,3,4, 16,17	однокімнатна
5,7,9,12,13	двокімнатна
6,8,10,14,15	трикімнатна
11,18,19,20	чотирьохкімнатна

Отже, за введеним номером квартири вивести її тип.

11. За першою буквою з назви дисципліни вивести її напрямок підготовки (технічні, природничі, гуманітарні).
12. За першою буквою з назви оператора мобільного зв'язку вивести його повну назву.
13. За кодом пакету мобільного зв'язку вивести назву оператора. Наприклад, (0) 50, (0) 66 – UMC.
14. За першою буквою з назви оператора мови Паскаль, вивести його повний синтаксис. Наприклад, задана буква 'i', вивести – 'if умова then дія1 else дія2;'.
'
15. За першою буквою з оператора мови Сі вивести, до якого типу обчислювального процесу він відноситься (розгалужений та циклічний).
16. За першою буквою з назви міста вивести країну. Перелік міст: Київ, Москва, Тула, Харків, Париж, Ліон, Генуя, Рівно, Норильськ, Венеція, Одеса.
17. За першою буквою з назви області України вивести в якій стороні світу відносно Києва вона знаходиться. Перелік областей: Львівська, Житомирська, Крим, Луганська, Донецька, Рівненська, Харківська, Одеська.
18. За введеним знаком арифметичної операції вивести її пріоритет.
19. За першою буквою з назви функції з бібліотеки math.h вивести її клас (степенева, логарифмічна, тригонометрична, визначення модуля).
20. За першою буквою з назви марки автомобіля вивести країну виробника.

21. За номером кольору у веселці вивести повідомлення, чи відноситься він до теплих чи холодних кольорів (таплі – червоний, жовтий і т.п., холодні – синій, голубий і т.п.).

Завдання 2.3

Скласти програму для перевірки, чи належить точка з координатами (x, y) до заданої області згідно варіанта завдання.

Рекомендації до виконання роботи.

В математичній постановці задачі необхідно визначити:

1. Рівняння прямих та кола, що обмежують задану область.

Для прямої рівняння визначається за формулою:

$$\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1},$$

де x_1, y_1, x_2, y_2 – координати двох точок, що належать прямій.

Для кола:

$$(x - x_c)^2 + (y - y_c)^2 = R^2,$$

де x_c, y_c – координати центра, R – радіус.

2. Скласти логічний вираз істинний, коли точка з координатами (x, y) належить заданій області.

Приклад виконання роботи

В якості приклада буде розглянуто математичну постановку задачі (МПЗ) для двох варіантів областей, що задані на рис. 1. та рис.2. Інші пункти прикладу виконання роботи будуть продовжувати МПЗ для рис. 1.

1. Математична постановка задачі.

Приклад 1 (рис. 1)

Задана на рис. 1 область обмежена колом, центр якого знаходиться в точці $(3, 2)$ і радіусом 2 та прямою, що проходить через пару точок $(3, 4)$, $(5, 2)$.

1. Рівняння кола буде:

$$(x - 2)^2 + (y - 3)^2 = 2^2$$

2. Рівняння прямої буде:

$$\frac{x - 3}{5 - 3} = \frac{y - 4}{2 - 4}$$

$$(x - 3) \cdot (-2) = (y - 4) \cdot 2$$

Після перетворень отримуємо: $y = 7 - x$.

3. Отже, для того щоби точка з координатами (x, y) потрапила в задану область, вона повинна, по-перше бути всередині кола, тобто її координати повинні задовольняти умові:

$$(x - 2)^2 + (y - 3)^2 \leq 4.$$

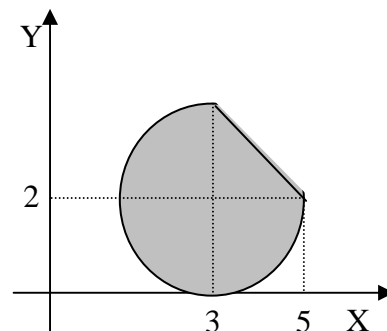


Рис. 1

По-друге, для того щоби потрапити в задану область точка повинна бути нижче прямої, отже, її координати повинні задовольняти умові:

$$y \leq 7 - x.$$

4. Враховуючи те, що наведені умови повинні виконуватися одночасно, для побудови загального логічного виразу їх необхідно об'єднати логічним "І" (математична операція – кон'юнкція \wedge).

Приклад 2 (рис. 2)

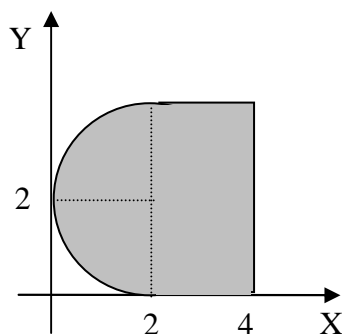


Рис. 2

Задана на рис. 2 область обмежена трьома прямими, які описуються рівняннями:

$$y=2, y=0, x=4,$$

та півколом з координатами центру (2,2) і радіусом 2.

Рівняння кола буде:

$$(x-2)^2 + (y-2)^2 = 2^2.$$

Точка з координатами (x, y) буде належати заданій області, якщо вона потрапляє в коло, тобто виконується умова:

$$(x-2)^2 + (y-2)^2 \leq 2^2,$$

або до прямокутника, для якого виконуються умови:

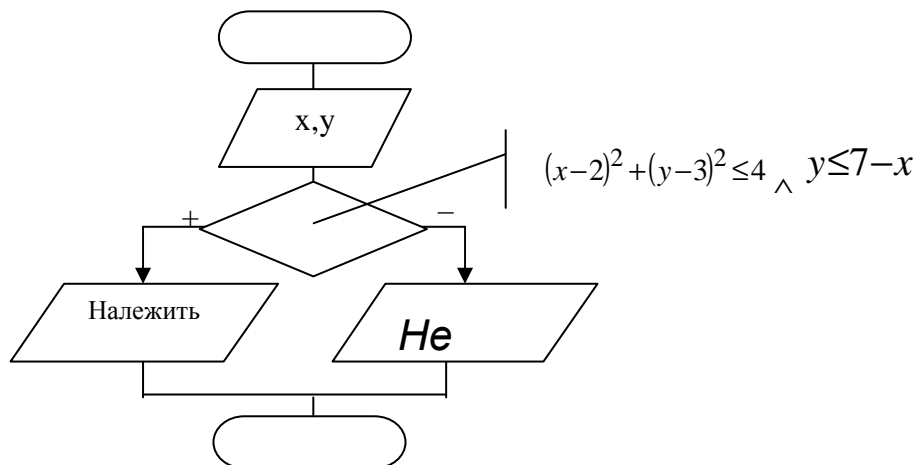
$$x \geq 2 \wedge x \leq 4 \wedge y \geq 0 \wedge y \leq 4.$$

Ці чотири умови об'єднані кон'юнкцією \wedge (логічне "І") тому, що кожна з них повинна виконуватися для того, щоби точка за координатами (x, y) потрапляла всередину прямокутника.

Отже, точка належить області заданій на рис. 2, якщо вона потрапляє АБО до кола, АБО до прямокутника. Ці дві умови можна об'єднати в один логічний вираз операцією диз'юнкції \vee (логічне "АБО"):

$$(x-2)^2 + (y-2)^2 \leq 2^2 \vee (x \geq 2 \wedge x \leq 4 \wedge y \geq 0 \wedge y \leq 4).$$

Вхідні дані	Дії	Вихідні дані
x, y – координати точки дійсного типу.	Якщо виконується $(x-2)^2 + (y-2)^2 \leq 4 \wedge y \leq 7 - x$, тоді вивести на екран інакше вивести на екран	Належить Не належить



Тестовий приклад.

Для перевірки правильності МПЗ та складеної програми розглянемо такі випадки:

Умова виконується.

- ◆ $x=3, y=1$;
- ◆ $x=3, y=4$;
- ◆ $x=4, y=2$.

Умова не виконується.

- ◆ $x=4.5; y=3$;
- ◆ $x=1, y=1$;
- ◆ $x=6, y=5$.

2. Аналіз результатів роботи програми.

По всіх тестових прикладах результати, що були отримані вручну та програмно співпадають, це доводить правильність роботи програми.

Вхідні дані	Розрахунок вихідних даних		
	вручну	Паскаль	Сі
$x=3, y=1$	Належить	Належить	Належить
$x=3, y=4$	Належить	Належить	Належить
$x=4, y=2$	Належить	Належить	Належить
$x=4.5; y=3$	Не належить	Не належить	Не належить
$x=1, y=1$	Не належить	Не належить	Не належить
$x=6, y=5$	Не належить	Не належить	Не належить

3. Опис програмного інтерфейсу з користувачем.

Після запуску програми на екран виводиться повідомлення-запрошення про введення значення координат точки (x, y) . Необхідно ввести дійсне значення. Обмежень на значення x не встановлюється.

На екран виводиться повідомлення "належить" чи "не належить" залежно від результатів перевірки.

Після виведення результатів на екрані з'являється повідомлення:

"Для виходу з програми натисніть ESC".

Для продовження роботи програми необхідно натиснути будь-яку клавішу на клавіатурі, для виходу з програми – <ESC>.

6. Текст програм мовами Паскаль та Сі.

<pre>Uses Crt; Var x, y : real;</pre>	<pre>#include <stdio.h> #include <conio.h> #include <math.h> main() { float x, y;</pre>
------------------------------------------	-------------------------------------------------------------------------------------------------------------

```

begin
repeat
  ClrScr;
  Write('Введіть координати (x, y): ');
  ReadLn(x, y);

  if (sqr(x-3)+sqr(y-4)<=4) and (y<=7-x)
  then WriteLn('Належить')
  else WriteLn('Не належить');
WriteLn('Для виходу нажміть ESC')
until readkey=#27;
End.

```

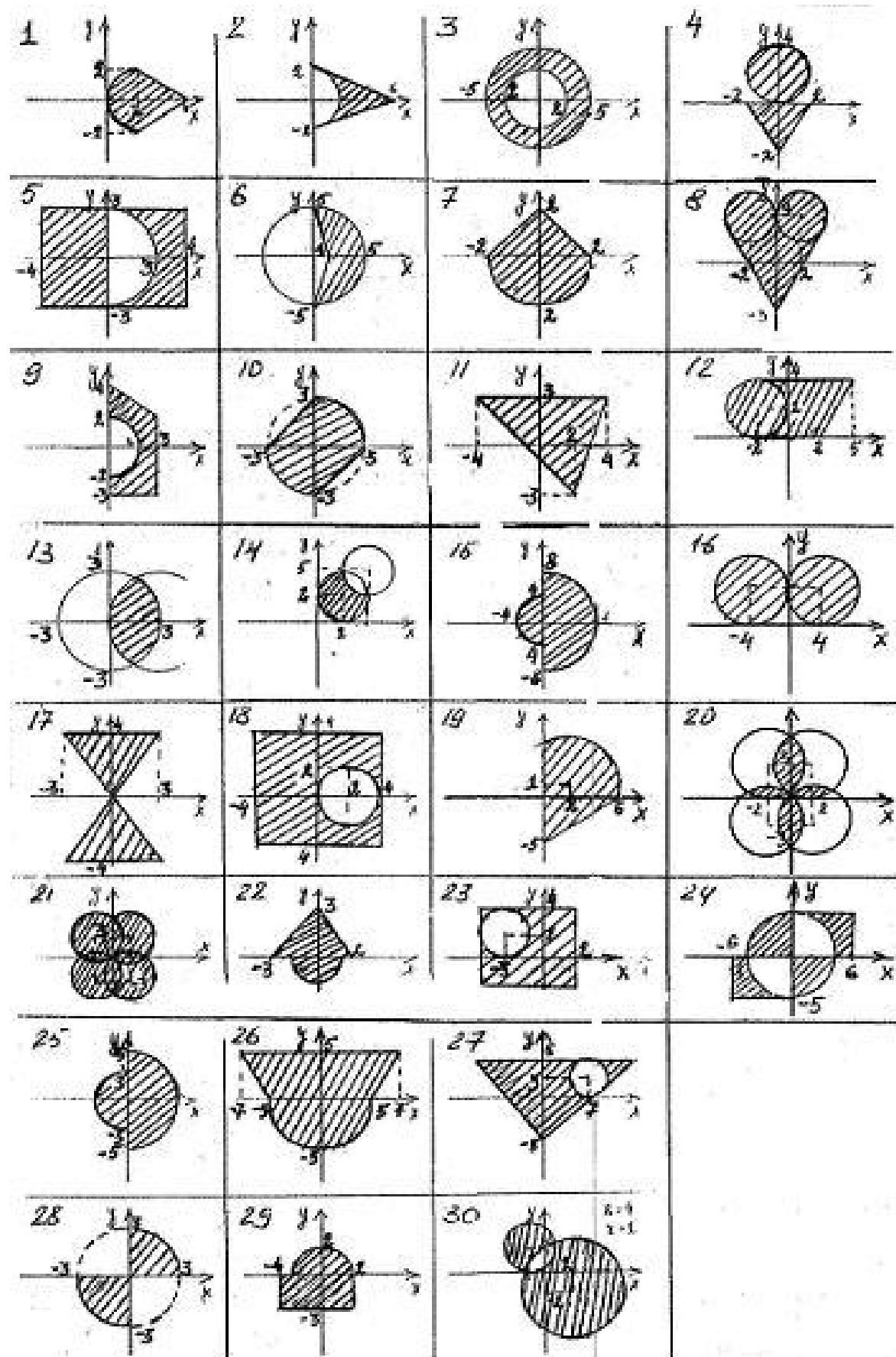
```

do
{
  clrscr();
  printf("Введіть координати (x, y): ");
  scanf("%f%f", &x, &y);

  if (pow(x-3, 2)+pow(y-4, 2)<= 4 && x<=5)
    printf("Належить\n");
  else printf("Не належить\n");
  printf("Для виходу нажміть ESC");
}
while (getch() != 27);
}

```

Варіанти завдань 2.3



Завдання 2.4

- ◆ Скласти програму для обробки натиснення клавiш на клавiатурi:
- ◆ вивести на екран повідомлення ‘ESC’, ‘Enter’, ‘Прогалина’, ‘TAB’ залежно від натиснутої клавiшi. Бажано, в програмi ввести константи для кодiв вiдповiдних клавiш: ESC=#27, ENTER=#13, SPACE=#32, TAB=#9.
- ◆ вивести на екран коди алфавiтно-цифрових клавiш i перевiрити, чи спiвпадають вони з кодами символiв в таблицi ASCII – кодiв.
- ◆ вивести на екран повідомлення “Простий код” або “Розширений код” та безпосередньо код в залежностi від натиснутої клавiшi.

Рекомендації до виконання роботи

Залежно від типу клавiатури на нiй можуть бути розташованi 83/84, 101/102, 122 клавiшi. Але принцип розташування та призначення основних клавiш на всiх типах клавiатури однаковi.

При натисненнi або вiдпусканнi клавiшi на клавiатурi генерується електричний сигнал, який розпiзнається контролером клавiатури. Контролер клавiатури передає код вiдповiдної клавiшi(скен-код) для подальшої обробки. Залежно від типу клавiшi генеруються прямi або розширенi коди.

Прямий скен-код клавiшi розмiром один байт i його значення спiвпадає з кодом вiдповiдного символу в таблицi ASCII-кодiв.

Розширений скен-код займає два байти: старший – нульовий, молодший – код клавiшi.

старший байт	молодший байт
00	код клавiшi

Алфавiтно-цифрова група клавiш та клавiшi Esc, Tab, BackSpace, Enter, пробiл(Space), клавiшi правої цифрової групи (крiм Del) повертають прямi скен-коди (рис. 3).

Функцiональнi клавiшi та клавiшi управлiння курсором повертають розширенi скен-коди.

Зсувнi клавiшi Shift, Ctrl, Alt, клавiшi CapsLock (фiксування режиму прописних букв) та NumLock(фiксування режиму введення цифр), PrintScrn(друку екрану), ScrollLock(блокування прокрутки екрану), Pause/Break(переривання команди) не повертають кодiв.

Але якщо зсувнi клавiшi використовуються в комбiнацiях з клавiшами алфавiтно-цифрової клавiатури, вони повертають розширений скен-код. Наприклад, якщо натиснути на клавiатурi кнопку з латинською буквою ‘a’, буде повернений код 97, при натисканнi Shift+a буде повернений код 65 – це символ ‘A’.



Рис. 3. Загальна схема розташування клавіш

Клавіші управління курсором (розширений код)			Права цифрова група клавіш (прямий код)			
Print Scrn 0	Scroll Lock 0	Pause Break 0	Num Lock 0	/	*	–
Insert 82	Home 71	PgUp 73	Home 55	↑ 56	PgUp 57	+ 43
Delete 83	End 79	PgDn 81	← 52		→ 54	
	↑ 72		End 49	↓ 50	PgDn 51	Enter 13
← 75	↓ 80	→ 77	Ins 48		Del 00:83	

Рис. 4. Розташування та коди клавіш управління курсором та правих цифрових.

Для обробки в програмі сигналів від клавіатури можна скористатися такими функціями:

Мова прогр.	Функція	Призначення	Бібліотека
Паскаль	KeyPressed : boolean;	Призупиняє виконання програми до натиснення користувачем клавіші на клавіатурі. Повертає значення true, якщо була натиснута будь-яка клавіша	CRT
Паскаль	ReadKey : char;	Призупиняє виконання програми до натиснення користувачем клавіші на клавіатурі. Повертає символ*, який відповідає коду натиснутої клавіші	CRT
Ci	char getch();	_____	conio.h

*– не для всіх клавіш на клавіатурі є відповідні символи, наприклад ESC(простий скен-код 27, в ASCII-таблиці не позначений), тому для перетворення числового коду в його символічний еквівалент в Паскаль-програмі використовується такий запис:

#код

Отже, символне подання *ESC=#27*, *ENTER=#13*, *TAB=#9*, *SPACE=#32* і т.п.

Сі-програма розглядає символні дані, як числові, тому, наприклад, символ 'а' та його код 97 з точки зору Сі-програми не розрізняються.

Для отримання інформації про натиснуті клавіші у програмі необхідно оголосити символну змінну і присвоювати їй значення, що повертає функція ReadKey(або getch()).

var c:char; begin c:= ReadKey; ...{подальша обробка коду} end;	#include <conio.h> main() { char c; c=getch(); }
----------------------------------------------------------------------------	--------------------------------------------------------------

Але це спрацює, якщо клавіша повертає простий скен-код. Якщо клавіша повертає розширений скен-код, необхідно двічі звертатися до функції ReadKey (або getch ()): перший раз для зчитування старшого байта #0 (або просто 0 в Сі), вдруге для зчитування молодшого байта – коду. Узагальнений фрагмент програми для зчитування і простих, і розширених кодів буде:

Uses CRT; var c:char; begin c:= ReadKey; if c=#0 then c:= ReadKey; ...{подальша обробка коду} end;	#include <conio.h> main() { char c; c=getch(); if (c= =0) c=getch(); .../* подальша обробка коду */ }
----------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------

Зазвичай тестування будь-якої програми потребує багатократного запуску, що вщент виснажує розробника :-). Можна додати до тексту програми оператор циклу з пост-умовою, в якому наприкінці виводити приблизно таке повідомлення: 'Для виходу натисніть ESC, для продовження будь-яку клавішу', а в рядку-завершенні циклу звертатися до ReadKey(або getch()) і перевіряти код на ESC.

{необхідні розділи Паскаль-програми} begin repeat {текст програми} Writeln('Для виходу натисніть ESC, для продовження будь-яку клавішу') Until readkey = ESC; End. /*препроцесування*/ main() { do { /* текст програми */ printf("\n Для виходу натисніть ESC, для продовження будь-яку клавішу \n"); } while (getch() != 27); }
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Приклади програм

<p>Завдання:</p> <ul style="list-style-type: none"> ♦ вивести на екран повідомлення 'ESC', 'Enter', 'Прогалина', 'TAB' в залежності від натиснутої клавіші. Бажано, в програмі ввести константи для кодів відповідних клавіш: ESC=#27, ENTER=#13, SPACE=#32, TAB=#9. ♦ вивести на екран коди алфавітно-цифрових клавіш і перевірити, чи співпадають вони з кодами символів в таблиці ASCII – кодів. 	<p>Завдання:</p> <ul style="list-style-type: none"> ♦ вивести на екран повідомлення "Простий код" або "Розширений код" та безпосередньо код в залежності від натиснутої клавіші.
<pre> Uses Crt; Const TAB=#9; ENTER=#13; ESC=#27; SPACE=#32; Var c:char; begin repeat clrscr; WriteLn('Натисніть клавішу'); c:= ReadKey; case c of TAB : WriteLn('Табуляція'); ENTER: WriteLn('Введення'); SPASE : WriteLn('Прогалина'); ESC : WriteLn('Ескейп') else WriteLn('Символ ', c, ' з кодом ', ord(c)); end; WriteLn('Для виходу натисніть ESC, для продовження будь-яку клавішу') until readkey = ESC; End.</pre>	<pre> #include <stdio.h> #include <conio.h> main() { int c; clrscr(); do { printf("Натисніть клавішу\n"); c=getch(); if (c==0) { c=getch(); printf("Розширений код>>>%d", c); } } else printf("Код клавіші >>>%d\n", c); printf("\n <ESC> для виходу або будь-яку клавішу для продовження \n"); } while (getch() != 27); }</pre>

ЛАБОРАТОРНА РОБОТА № 3.

Програмування циклічного обчислювального процесу

Завдання 3.1

- ◆ Скласти програму для порозрядної обробки цілочисельних (дійсних) значень згідно з варіантом завдання. Розглянути два варіанта алгоритму розв'язку: з циклом з перед-умовою та циклом з після-умовою. У звіті навести дві схеми алгоритму та проаналізувати переваги та недоліки кожного з них для розв'язку цього завдання;
- ◆ встановити точку останову на заголовку оператора циклу;
- ◆ відкрити вікно для перегляду значення вхідної змінної, змінної-параметра циклу та значення результуючої змінної;
- ◆ повторно запустити програму на виконання та виконати трасування тіла циклу, контролюючи значення параметру циклу та виконання (невиконання) умови циклу.

Рекомендації до виконання роботи

У програмуванні використовуються декілька систем числення: десяткова, двійкова, вісімкова, шістнадцяткова. Кожна з цих систем є позиційною (порядковою), тобто значущість цифри в записі числа залежить від її розташування в цьому числі. Наприклад, в числі 12 одиниця більш значуща (має більшу "питому вагу"), оскільки розташована в розряді десятків, а двійка в розряді одиниць [2; 7; 9].

Будь-яке ціле число в позиційній системі обчислення можна подати у вигляді суми :

$$N = \sum_{i=0}^{k-1} m_i \cdot W^i,$$

де N – подане ціле число; k – кількість цифр в числі (кількість розрядів); m – кількісний еквівалент цифри; W – основа системи числення; i – номер розряду цифри [2;7].

Розряди нумеруються з кінця числа, наприклад, для числа 25735:

Розряди	4	3	2	1	0
Число	2	5	7	3	5

Як суму в десятковій системі числення це число можна записати так:

$$2 \cdot 10^4 + 5 \cdot 10^3 + 7 \cdot 10^2 + 3 \cdot 10^1 + 5 \cdot 10^0.$$

Отже, для того, щоб у програмі розбити число на цифри, необхідно послідовно ділити його на W – основу системи числення, брати остачу від ділення:

$$m = \left\{ \frac{N}{W} \right\}$$

та залишати цілу частину від частки:

$$N = \left[\frac{N}{W} \right].$$

Цей процес повторюється доки ціла частина від частки більша за 0.

Приклад виконання роботи

Завдання: скласти програму, після виконання якої на екрані будуть надруковані цифри цілого числа N та номери розрядів, у яких вони розташовані.

1. Математична постановка задачі (МПЗ).
- 2.

Вхідні дані	Дії	Вихідні дані
N цілого типу.	1. $k=0; N= N ;$ 2. Доки $N \neq 0$, виконувати 3-5, інакше – пункт 6. 3. $m = \left\{ \frac{N}{W} \right\}.$ 4. Вивести $m, k.$ 5. $k=k+1; N = \left[\frac{N}{W} \right].$ Перехід до пункту 2. 6. Останов.	m – цифра числа N , ціле; k – номер розряду, ціле;

Схема алгоритму розв'язку задачі наведена на рис.1 (в звіті необхідно побудувати також алгоритм для циклу з після-умовою та провести його тестування).

3. Тестовий приклад (розглянуто тільки цикл з перед-умовою, в звіті додати ще аналіз циклу з пост-умовою).

Наприклад, для введення подано число 379.

$N = 379; k = 0;$.

На початку циклу відбувається перевірка : $N <> 0 ? (379 <> 0 ?)$ Результат - TRUE.

1^{ша} ітерація.

$m = 379 \bmod 10 = 9, k = 1, N = 379 \div 10 = 37$

2^{га} ітерація. $37 <> 0 \rightarrow \text{TRUE}$

$m = 37 \bmod 10 = 7, k=2, N=3.$

3^{тя} ітерація. $3 <> 0 \rightarrow \text{TRUE}$

$m = 3 \bmod 10 = 3, k=3, N=3 \div 10 = 0.$

4^{та} ітерація. $0 <> 0 \rightarrow \text{FALSE}$. Цикл закінчено.

Управління передається до наступного оператора після циклу, що завершує програму.

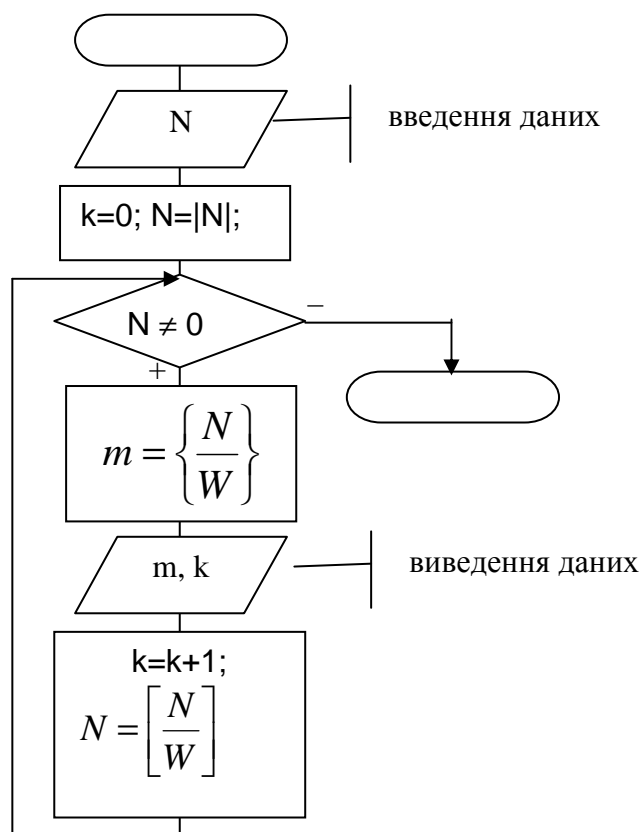


Рис. 1. Схема алгоритму розв'язку задачі (цикл з перед-умовою)

3. Аналіз результатів роботи програми (для циклу з після-умовою тестування провести самостійно).

Вхідні дані	Розрахунок вихідних даних				
	вручну	Паскаль		Сі	
N=379	m=3, k=0 m=7, k=1 m=3, k=2	Перед-умова	Після-умова	Перед-умова	Після-умова
		3 0	?	3 0	?
		7 1		7 1	
		9 2		9 2	
N=0	m=0, k=0	0 0	?		?
N=-12	m=2, k=0 m=1, k=1	2 0	?	2 0	?
		1 1		1 1	

По всіх тестових прикладах результати, що були отримані вручну та програмно повністю співпадають, це доводить правильність роботи програми.

4. Опис програмного інтерфейсу з користувачем.

Після запуску програми на екран виводиться повідомлення-запрошення про введення значення x. Необхідно ввести дійсне значення x. Обмежень на значення x не встановлюється.

Після розрахунку на екран виводиться значення y.

Для завершення роботи програми необхідно натиснути будь-яку клавішу на клавіатурі.

5. Текст програм мовами Паскаль та Сі.

<pre> Uses CRT; Const W=10; {основа системи числення} Var N: longint; m, k : byte; Begin ClrScr; WriteLn('Введіть число N : '); ReadLn (N); k:=0; N:=abs(N); While N<>0 Do begin m := N mod W; WriteLn(' Цифра ',m:2,' розташована в розряді № ',k:2); k:=k+1; N:=N div W; end; readkey; End.</pre>	<pre> #include <stdio.h> #include <conio.h> #include <stdlib.h> /* для abs */ const int w=10; main() { long n; short m, k=0; clrscr(); printf("\nВведіть n: "); scanf("%ld", &n); n=abs(n); while (n != 0) { m= n% w; printf("\n Цифра %hd розташована в розряді № %hd", m, k); k++; n=n/10; } getch(); }</pre>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Варіанти завдань 3.1

1. Визначення суми парних цифр цілого числа.
2. Визначення добутку непарних цифр цілого числа.
3. Визначення суми перших 5 цифр з дробової частини дійсного числа.
4. Визначення чи дорівнює сума перших двох цифр цілого числа сумі останніх двох цифр.
5. Визначення чи дорівнює сума цифр з цілої частини дійсного числа сумі такої ж кількості цифр з дробової частини.
6. Визначення скільки нулів в записі цілого числа.
7. Визначення, чи дорівнює сума цифр цілої частини дійсного числа сумі такої ж кількості цифр в дробовій частині.
8. Вилучення з запису цілого числа нулів.
9. Перевірити, чи виконується для заданого цілого ознака ділення на 3.
10. Виведення на екран перших (з ліва) К цифр цілого числа (К задає користувач).
11. Виведення на екран Р цифр з дробової частини дійсного числа (Р задає користувач).
12. Перевірити, чи дорівнює квадрат числа кубу суми його цифр.
13. Визначення, чи є серед С цифр дробової частини дійсного числа цифра 0.
14. Визначення кількості цифр Т в записі цілого числа (Т задає користувач).

15. Замінити місцями цілу та дробову частини дійсного числа.
16. Визначення чи симетричні цифри дійсного числа відносно крапки.
17. Визначення кількості цифр, в записі цілого числа, що більші за 3 та менші 7.
18. Визначення різниці між сумою цифр цілої та такої ж кількості цифр дробової частини дійсного числа.
19. Перевірити, чи виконується для заданого числа ознака ділення на 9.
20. Визначення, чи дорівнює ціле число квадрату суми його цифр.
21. Визначення добутку цифр цілого числа.
22. Визначення, чи дорівнює квадрат суми цифр числа сумі квадратів його цифр.
23. Перевірити, чи є число степенем числа 3.
24. Замінити першу та останню цифри цілого числа на цифру P (P задає користувач).
25. Замінити місцями першу та останню цифри цілого числа.
26. *Перевірити, чи є число паліндромом, тобто таким, що зліва направо та справа наліво читається однаково.
27. *Визначити найбільший спільний дільник чисел A та B.
28. *Вивести на екран всі "досконалі" числа в діапазоні від 0 до 100000. Число "досконале", якщо воно дорівнює добутку своїх цифр.

Завдання 3.2

- ◆ Скласти програму для обчислення суми скінченного ряду згідно з варіантом завдання;
- ◆ встановити точку останову на заголовку оператора циклу;
- ◆ відкрити вікно для перегляду значення вхідної змінної, змінної-параметра циклу та значення результуючої змінної;
- ◆ повторно запустити програму на виконання та виконати трасування тіла циклу, контролюючи значення параметру циклу та виконання (невиконання) умови циклу.

Рекомендації до виконання роботи

Розглянемо приклад. Обчислити суму N членів ряду:

$$S = 1 + \underbrace{\frac{1+x}{2} + \frac{1+2x}{3} + \frac{1+3x}{4} + \dots}_{n \text{ елементів}}$$

Елементи цього ряду мають рекурентне співвідношення, тому що кожний наступний член ряду можна визначити через попередній.

Кожний член ряду можна подати як дріб виду $\frac{a}{b}$ з початковими значеннями $a=1$, $b=1$.

Для отримання наступного значення a необхідно до попереднього додати x , а для отримання наступного значення b до попереднього необхідно додати 1. Отже,

$$a_1 = 1, a_2 = a_1 + x, \dots, a_i = a_{i-1} + x, \dots$$

$$b_1 = 1, b_2 = b_1 + 1, \dots, b_i = b_{i-1} + 1, \dots$$

Виходячи з цього, загальним членом ряду буде $\frac{1+(i-1)x}{i}$, де $i = \overline{1, N}$, а S

буде обчислюватися за формулою:

$$S = \sum_{i=1}^N \frac{(i+1)x}{i}.$$

Виходячи з математичної постановки задачі, члени ряду обчислюються в циклі, кількість ітерацій циклу заздалегідь відома, крок зміни параметру циклу 1. Тому при розробці схеми алгоритму та програми доцільним буде використати цикл з параметром (рис. 2, а).

Якщо важко визначити загальний член ряду, можна скористатися проміжними змінними для визначення члену ряду і подати обчислення S за такою схемою:

$$1. S = 0, a = 1, b = 1;$$

$$2. S = S + \frac{a}{b}, \quad a = a + x, \quad b = b + 1; \text{ для } i = \overline{1, N}.$$

Враховуючи те, що значення b збігається зі значенням i, то можна вираз спростити:

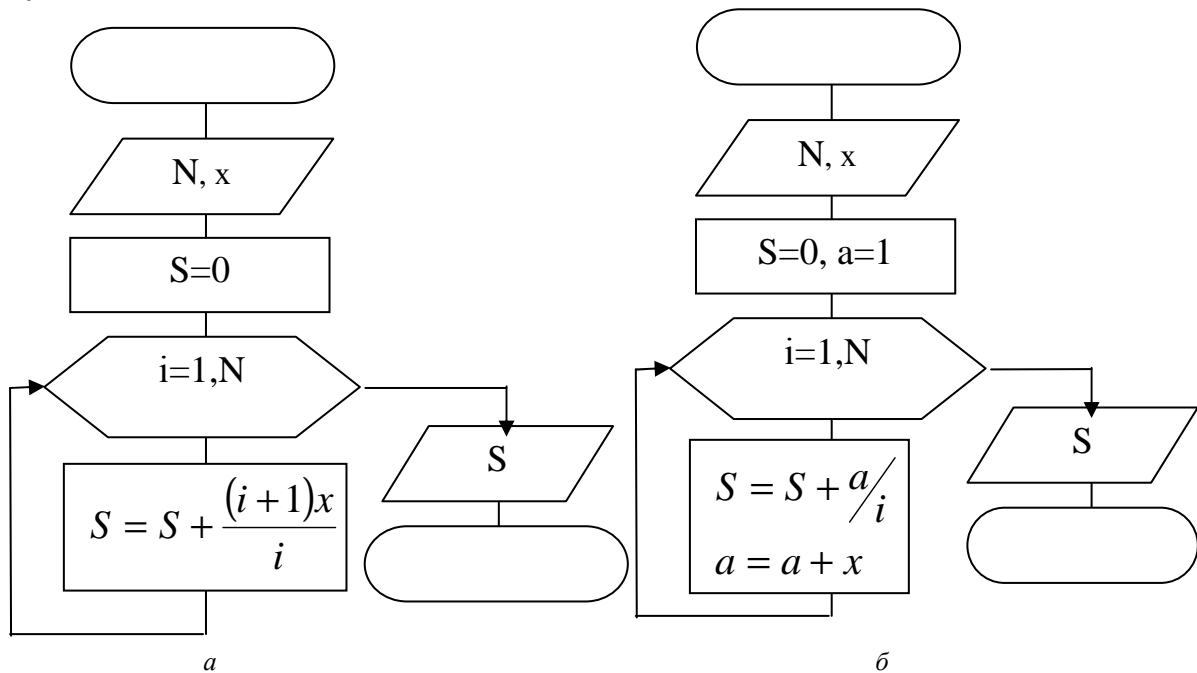


Рис. 2. Схема алгоритму обчислення суми ряду.

$$S = S + \frac{a}{i}, \quad a = a + x; \text{ для } i = \overline{1, N}.$$

Схема алгоритму для такого способу обчислення наведена на рис.2, б.

Примітка. Під час виконання лабораторної роботи можна обрати будь-який спосіб обчислення суми ряду. Звіт з лабораторної оформляється згідно до загальних вимог до оформлення звітів.

Варіанти завдання 3.2

$$1. \quad y = \underbrace{\frac{1}{2} + \frac{3}{2} + \frac{3}{4} + \frac{5}{4} + \frac{5}{6} + \dots}_{n \text{ елементів}}$$

$$2. \quad y = \underbrace{\sqrt{2 + \sqrt{2 + \sqrt{2 + \dots + \sqrt{2}}}}}_{n\text{-коренів}}$$

$$3. \quad y = \frac{1}{\sin 1} + \frac{1}{\sin 1 + \sin 2} + \dots + \frac{1}{\sin 1 + \dots \sin N}$$

$$4. \quad y = \sqrt{3 + \sqrt{6 + \sqrt{9 + \dots + \sqrt{99}}}}$$

$$5. \quad y = 1 - \underbrace{\frac{x}{2} + \frac{x^2}{4} - \frac{x^3}{8} + \dots}_{m \text{ елементів}}$$

$$6. \quad y = 1 + \underbrace{\frac{1}{\sqrt{3}} + \frac{1}{\sqrt{6}} + \frac{1}{\sqrt{10}} + \frac{1}{\sqrt{15}} + \dots}_n$$

$$7. \quad y = \cos x + \cos x^2 + \dots + \cos x^n$$

$$8. \quad y = x + \underbrace{\frac{x^2}{2} + \frac{x^3}{3} + \dots}_h$$

$$9. \quad y = 1 - \underbrace{\frac{a}{2} + \frac{3a}{4} - \frac{5a}{8} + \dots}_n$$

$$10. \quad y = \frac{2}{3!} + \frac{4}{5!} + \dots + \frac{2n}{(2n-1)!}$$

$$11. \quad y = 1 + 2^3 + \frac{3^3}{2!} + \dots + \frac{n^3}{(n-1)!}$$

$$12. \quad y = x - \underbrace{\frac{x^3}{3} + \frac{x^5}{5} - \dots}_n$$

$$13. \quad y = \cos(1 + \cos(2 + \dots + \cos(39 + \cos(40)))) \dots)$$

$$14. \quad y = x - \underbrace{\frac{x^3}{3!} + \frac{x^5}{5!} - \dots}_v$$

$$15. \quad y = x - \underbrace{\frac{x^2}{2} + \frac{x^3}{3} - \dots}_n$$

$$16. \quad p = \underbrace{\left(x - \frac{1}{2^2}\right) \left(x^2 - \frac{1}{3^2}\right) \dots}_n$$

$$17. \quad y = \underbrace{\sin(x) + \sin(\sin(x)) + \sin(\sin(\sin(x))) + \dots}_{n \text{ елементів}}$$

$$18. \quad y = \underbrace{\frac{5w}{1 \cdot 2} + \frac{w}{2 \cdot 3} + \frac{5w}{3 \cdot 4} + \frac{w}{4 \cdot 5} + \dots}_{n \text{ елементів}}$$

$$19. \quad y = 1 + \underbrace{\frac{3}{2!} + \frac{6}{5!} + \dots}_{K \text{ елементів}}$$

$$20. \quad y = \underbrace{\frac{3}{2!} - \frac{7}{3!} + \frac{11}{4!} - \dots}_{K \text{ елементів}}$$

$$21. \quad y = \underbrace{\frac{x}{1 \cdot 3} + \frac{2x}{3 \cdot 5} + \frac{3x}{5 \cdot 7} + \dots}_n$$

$$22. \quad y = x + \underbrace{\frac{x}{1!} + \frac{x^2}{2!} + \dots}_v$$

$$23. \quad y = 1 - \underbrace{\frac{x^2}{2!} + \frac{x^4}{4!} - \dots}_k$$

$$24. \quad y = 1 - \underbrace{\frac{3}{2^2} + \frac{5}{3^2} - \frac{7}{4^2} + \dots}_n$$

$$25. \quad y = 1 + \underbrace{\frac{2}{3^2} + \frac{3}{5^2} + \dots}_n$$

ЛАБОРАТОРНА РОБОТА № 4.

Програми з підпрограмами

Завдання 4.1

- Скласти програму для табулювання функції згідно з варіантом завдання на заданому проміжку (стовпець “Функція та проміжок” в таблиці варіантів завдань);
- обчислення значення функції оформити в програмі як підпрограму-функцію, описати в звіті як “чорну скриню”, розробити для неї МПЗ, скласти схему алгоритму;
- розробити МПЗ, перевіряючи чи визначена задана функція на вказаному проміжку, та СА для головної програми. Під час табулювання функції провести обчислення, що вказані в стовпці “Завдання” таблиці варіантів завдань;
- результати табулювання функції вивести на екран у вигляді таблиці:

X	Y
x_1	y_1
x_2	y_2
...	...
x_n	y_n

- для виведення заголовка стовпців таблиці скласти підпрограму без параметрів;
- провести трасування програми, дослідивши переходи між головною програмою та функцією.
-

Рекомендації до виконання роботи

Табулювання функції

Під терміном “табулювання функції” розуміють обчислення n значень певної функції $y=f(x)$ для x , значення якого належать проміжку $[a,b]$ і змінюється з кроком Δx (рис.1) [1; 2; 7]. Отже, необхідно обчислити значення:

$$y_1 = f(x_1), \quad \text{де } x_1 = a,$$

$$y_2 = f(x_2), \quad \text{де } x_2 = x_1 + \Delta x,$$

...

$$y_n = f(x_n), \quad \text{де } x_n = b.$$

Якщо задано n – кількість точок на проміжку, тоді $\Delta x = \frac{b-a}{n-1}$.

Процес обчислення значень функції носить циклічний характер, тіло циклу складають дії:

$$y = f(x), \quad x = x + \Delta x.$$

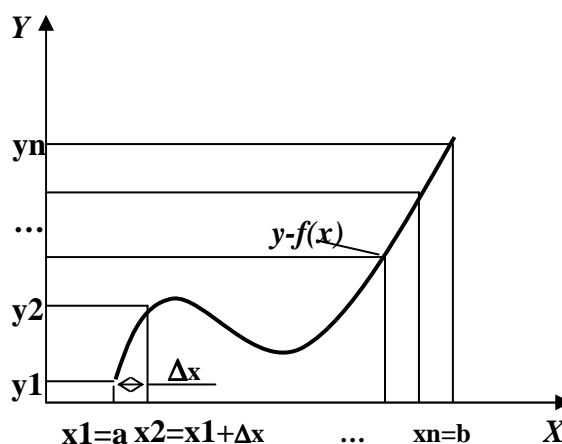


Рис. 1

Параметром циклу є змінна x з початковим значенням $x = a$, зміна параметру відбувається в тілі циклу. Умовою закінчення циклу буде $x > b$ (або зворотна, як умова продовження циклу).

Крім того в програмі в тілі циклу **до** зміни x необхідно вивести на екран значення x та y .

Формування таблиці

Результати обчислення виводяться на екран у вигляді таблиці, яка формується у програмі за допомогою оператора WriteLn (printf). Наприклад, для виводу шапки необхідно задати таку послідовність:

WriteLn('_____');	printf("_____\n");
WriteLn(' X Y ');	printf(" X Y \n");
WriteLn('_____').	printf("_____\n").

Або з використанням символів псевдографіки, що беруться з таблиці ASCII-кодів символів(деякі коди наведено у табл.1). Для того щоб задати в програмі оператор WriteLn (printf), який виводить символи псевдографіки необхідно праву цифрову групу клавіш перемкнути в режим набору цифр(ввімкнути NumLock) і, утримуючи клавішу ALT, набрати код необхідного символу. Наприклад, для виведення верхньої границі таблиці необхідно задати символи з кодами 218, 196, 196, ... 194, 196, ..., 191:

WriteLn (' _____');	printf (" _____\n");
---------------------	----------------------

Оформлення звіту з лабораторної роботи

Звіт з лабораторної роботи оформлюється згідно з загальними вимогами до оформлення звітів.

Таблиця 1 – Символи псевдографіки			
Код	Символ	Код	Символ
179		195	┤
180	├	196	─
191	└	197	┘
192	┌	217	┐
193	└─	218	┌─
194	┐─		

Варіанти завдання 4.1

№	Функція та проміжок	Завдання
1	$y = x^3 - 4 x + 3$ при $x \in [-2, 3]$	найбільше за модулем значення функції.
2	$y = \frac{x^3 - 4x^2}{x^2 + 1}$ при $x \in [-1, 6]$	середнє значення функції на заданому проміжку.
3	$y = \frac{\sqrt{ x^3 - 3 }}{x + 3}$ при $x \in [-2, 5]$	кількість значень функції більших за К.
4	$y = \frac{(x - 2)^2}{x - 1}$ при $x \in [-5, 1)$	максимальне значення функції та при якому значенні воно досягається.
5	$y = x^3 - 3 + x - 5 $ при $x \in [0, 6]$	максимальне та мінімальне значення функції.
6	$y = \left(\frac{5}{2}\right)^x + 2x$ при $x \in [-2, 5]$	визначити скільки додатних значень функції.
7	$y = \sqrt{x + 1} - x^2$ при $x \in [-1, 10]$	мінімальне за модулем значення функції.
8	$y = \sqrt[3]{x^2} + x$ при $x \in [-2, 9]$	середнє арифметичне серед тих значень у, що більше за Р.
9	$y = -\frac{1}{2}x^2 - \frac{1}{2} x + 5x$ при $x \in [-6, 6]$	скільки від'ємних значень приймає функція на проміжку.
10	$y = e^{x^2 + 4}$ при $x \in [-6, 6]$	значення х при якому функція приймає найбільше значення.
11	$y = \frac{e^{x^2 - 5}}{x - 1}$ при $x \in (1, 4]$	значення х при якому функція приймає найменше значення.
12	$y = \lg(6x^3 + 2x + 5)$ при $x \in (-0.8, 5]$	середнє арифметичне додатних значень функції.
13	$y = \ln(x^2 - 5x + 5)$ при $x \in (-10, 1.3]$	середнє арифметичне від'ємних значень функції.
14	$y = \sin^2 \frac{x}{2}$ при $x \in [-\pi, \pi]$	кількість значень функції, що більші за 0.5.
15	$y = \cos \frac{x^3}{2}$ при $x \in [-\pi, \pi]$	кількість значень функції, що потрапляють в проміжок [0.2, 0.4].

№	Функція та проміжок	Завдання
16	$y = \frac{x^2 + \sin^2 x - 2x \sin x}{x^3}$ при $x \in (0, 2\pi]$	суму значень функції менших за задане В.
17	$y = \lg(a^2 x + 1)$ при $x \in (0, 2]$, a – ціле.	суму від'ємних значень функції.
18	$y = \sin\left(\frac{3}{2}x\right) - \ln(10x)$ при $x \in [1.5, 7.5]$	кількість значень функції більших за -1 .
19	$y = 3\sin^2(5x + 0.5)$ при $x \in [0, 0.2]$	найменше значення функції.
20	$y = e^x \sin x^2$ при $x \in [0, 3.0]$	добуток від'ємних значень функції.
21	$y = \frac{1-x}{1-\operatorname{tg} x^2}$ при $x \in [0, \pi/2)$	кількість значень функції, у яких дробова частина більша за 0.5.
22	$y = \sin x \cos ax^2$ при $x \in [0, 6]$	кількість значень функції, які за модулем більші ніж попереднє значення.
23	$y = \cos^2 x - \operatorname{tg} x$ при $x \in [0.2, 8]$	значення x , в яких функція не визначена на заданому проміжку.
24	$y = \frac{\sin x}{5 - \cos x}$ при $x \in [0, 3]$	середнє арифметичне значень y , що менші за значення аргументу x .
25	$y = \cos ax \cdot e^x$ при $x \in [0, 5]$	кількість значень y , що більші за задане С.

Завдання 4.2

- Використавши свій варіант завдання 1.2 (л.р.№1), переробити програму, оформивши обчислення з цього варіанта завдання як підпрограму. Описати цю підпрограму як “чорну скриню”, визначити, які дані будуть передаватися за значенням, а які за посиланням, розробити для підпрограми МПЗ, скласти схему алгоритму;
- розробити МПЗ та СА для головної програми.
- Оформити звіт згідно до загальних вимог.

Рекомендації до виконання роботи

Як приклад розглянемо найпростішу програму для визначення площі та периметру прямокутника, що заданий довжинами двох сторін.

Математична постановка задачі.

Для визначення площі та периметру прямокутника необхідно знати довжини двох його сторін. Введемо позначення:

a – довжина прямокутника, задається в метрах(м);

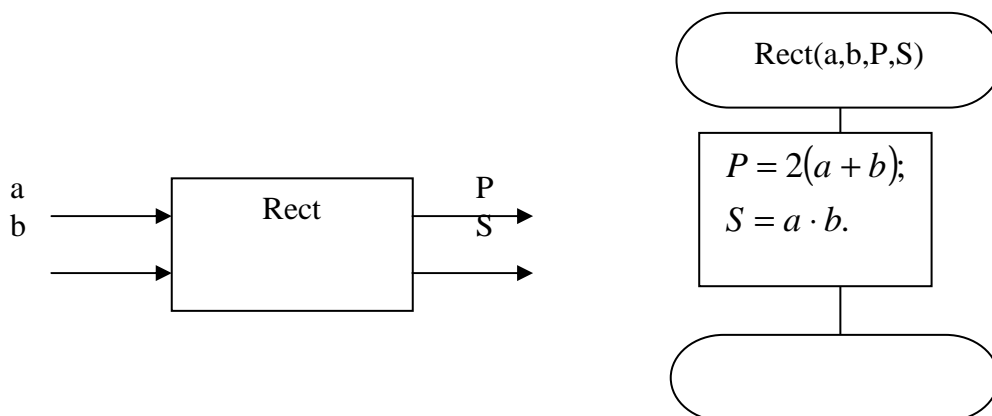
b – ширина прямокутника (м);

P – периметр прямокутника (м);

S – площа прямокутника (m^2).

Вхідні дані	Дії	Вихідні дані
a, b дійсного типу. Обмеження: $a, b > 0$.	$P = 2(a + b);$ $S = a \cdot b.$	P, S – дійсні.

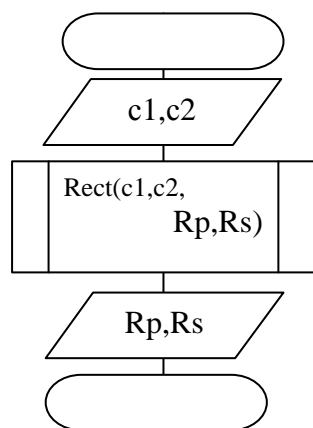
Якщо оформляти цю програму як підпрограму і подати її як “чорну скриню”, маємо:



Враховуючи те, що значення P, S є результатами роботи підпрограми їх необхідно описати як параметри-змінні.

Текст підпрограми буде таким:

<pre> Procedure Rect (a, b:real; Var P, S: real); begin P:=(a+b)*2; S:=a*b; End;</pre>	<pre> void rect (float a, float b, float *P, float *S) { *P=(a+b)*2; *S=a*b; }</pre>
------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------



6. Головна програма.

Вхідні дані:

- ◆ c1, c2 – довжини сторін прямокутника.

Вихідні дані:

- ◆ Rp – периметр прямокутника;
- ◆ Rs – площа прямокутника.

<pre> Uses Crt; var c1, c2, Rp, Rs:real; {опис підпрограми} ... Begin WriteLn('Введіть дані:'); ReadLn(c1, c2); Rect (c1, c2, Rp, Rs); WriteLn('Периметр=', Rp, 'Площа=', Rs); End.</pre>	<pre> #include <stdio.h> /*опис підпрограми*/ ... main() { float c1, c2, Rp, Rs; printf("Введіть дані:"); scanf("%f%f", &c1, &c2); rect (c1, c2, &Rp, &Rs); printf("Периметр=%f\tПлоща=%f\n", Rp, Rs); }</pre>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Завдання 4.3

Скласти програму з використанням підпрограм зі стандартних бібліотек для управління виведенням на екран у складі двох підпрограм:

1. Перша функція – в текстовому режимі роботи екрана виводить кольоровим шрифтом в центрі екрана інформацію про розробника програми та привітання з найближчим святом для викладача;
2. Друга – в графічному режимі роботи екрана виводить емблему розробника, яка складається як мінімум з:
 - ◆ трьох контурних та трьох заповнених певним стилем геометричних примітивів;
 - ◆ ПІБ розробника, виведеного на екран в горизонтальному та вертикальному напрямках різними стилями шрифту, розміром та кольором.

В головній програмі викликати першу підпрограму, затримавши зображення до натиснення певної клавіші, потім встановити графічний режим та викликати другу підпрограму.

В звіті навести опис всіх використаних бібліотечних підпрограм та опис з зазначенням екранних координат, кольорів та стилів зображень, що виводить програма.

Завдання для самостійного виконання

В головній програмі створити текстове меню, в якому 3 пункти: текстова заставка, графічна емблема, вихід. Вибір пункту меню здійснюється через натиснення клавіші з номером пункту.

* - це завдання для самостійного виконання за бажанням!!!

ЛАБОРАТОРНА РОБОТА №5. Обробка масивів даних

Завдання 5.1²

- Скласти математичну постановку задачі, схему алгоритму та програму для обробки одновимірного масиву даних згідно з варіантом завдання;
- описати дві підпрограми, передавши в них масив як параметр: перша – для введення, друга – для виведення елементів масиву на екран по 10 елементів в екранний рядок;
- перед виведенням результатів роботи необхідно, очистити екран, вивести вхідний масив, а потім результати роботи програми;
- * описати програмний модуль (бібліотеку), в яку перенести підпрограми введення/ виведення масиву; передати в ці підпрограми відкритий масив; перевірити працездатність на масивах різних розмірів.

Рекомендації до виконання роботи

Введення/виведення масиву

В мові Паскаль для передачі масивів як параметрів підпрограм необхідно [1; 2; 4; 9]:

- описати тип для масиву в розділі type;
- при передачі в підпрограму описати масив як параметр-змінну.

Наприклад, заданий цілочисельний масив з 50-ти елементів [4].

Const N = 50;

Type

Ar_Type = array [1..N] of integer;

Var

A, B : Ar_Type; {описано два масиви A та B}

Procedure Input_Ar(var X:Ar_Type);

{в процедури передається параметр-змінна масив X}

Var I : byte;

begin

WriteLn('Введення елементів масиву');

for I := 1 to N do

begin

Write('Введіть елемент ', i:3, ' :');

Read(X[i]);

end;

end;

Begin

Input_Ar(A);

Input_Ar(B);

end.

² - Звіт з лабораторної оформлюється згідно до вимог щодо оформлення звітів для програм з підпрограмами (п.р.№4).

Під час виведення елементів у випадках, коли всі елементи масиву не вміщуються в екранний рядок, до циклу доцільно додати додаткову перевірку:

```
WriteLn('Виведення елементів масиву');
for I := 1 to N do
begin
    Write(X[i]:5);
    if I mod 10 = 0 then WriteLn;
end;
```

В наслідок роботи такого циклу елементи масиву будуть виводитися по десять в екранний рядок.

Мовою Сі введення/ виведення масиву можна організувати так:

```
#include <stdio.h>
#define N 50
void input_ar (int* x) /* передача масиву як параметра-змінної */
{   int i;
    printf("Введення елементів масиву \n");
    for (i=0; i<N; i++)
        { printf("Введіть елемент %d :", i);
          scanf("%d", &x[i]);
        }
}
```

Загальний синтаксис передачі масиву у функцію мови Сі такий [6-8; 10] :

тип_результату_функції ім'я_функції

(тип_елементів_масиву ім'я_масиву)*

Звернення до параметра-масиву в тілі функції здійснюється як і до звичайного масиву:

ім'я_масиву [індекс_елемента]

Відкриті масиви

На жаль процедура Input_Ar не є досить універсальною, вона призначена тільки для введення масиву одного типу Ar_Type. Якщо в програмі виникне потреба ввести цілочисельний масив з іншою кількістю елементів, процедурою Input_Ar неможливо буде скористатися для цього. Більш універсальною ця процедура стане, якщо вона буде призначена для введення цілочисельного масиву довільного розміру. Для цього необхідно скористатися відкритим масивом. [1; 2; 4; 9]

Передача масиву в функцію мови Сі більше схожа на відкриті масиви мови Паскаль, тому що під час передачі масиву його розмір фактично нічим не обмежений. Якщо бажано, щоби функція обробляла масиви різні за розміром, другим параметром в таку функцію доцільно передати кількість елементів в масиві [6-8; 10].

Варіанти завдання 5.1

1. Масив зі 100 цілих чисел замінити всі від'ємні елементи на їх модулі.
2. Масив зі 100 цілих чисел знайти суми всіх додатних та всіх від'ємних елементів.
3. У масиві зі 100 цілих чисел за умови рівності елемента 0 замінити його "сусідів" зліва та справа відповідно на -1 та 1.
4. У масиві зі 100 цілих чисел знайти суму найбільшого та найменшого з від'ємних елементів.
5. Відповідному елементу масиву У присвоїти true, якщо елемент масиву Х додатний.
6. У масиві зі 100 цілих чисел знайти суму мінімального та максимального елементів.
7. У масиві зі 100 цілих чисел замінити на 0 всі елементи, індекс яких дорівнює числу Фібоначчі.
8. У цілочисельному масиву X(100) замінити значення елементів на значення з протилежним знаком.
9. У масиві з 50 дійсних значень визначити суми S_p S_{np} парних та непарних елементів.
10. У масиві зі 100 елементів кожний додатний елемент подвоїти, а для кожного від'ємного зменшити вдвічі його абсолютне значення.
11. У масиві зі 100 цілих чисел поміняти місцями сусідні елементи (1-й з 2-м, 3-й з 4-м і т.д.).
12. Записати в масив K(20) елементи ряду Фібоначчі.
13. Перевірити чи утворюють елементи масив X(10) ряд Фібоначчі.
14. В масиві зі 100 елементів визначити суму елементів, розташованих між макимальним та мінімальним значеннями.
15. Визначити чи утворюють елементи масиву B(75) послідовність, що спадає.
16. Визначити чи всі елементи масиву B(100) не перебільшують діапазону [a, b].
17. Знайти добуток елементів масиву K(70), що знаходяться зліва від максимуму.
18. Поміняти місцями елементи масиву з індексами однієї парності (1-й та 3-й, 2-й та 4-й і т.д.).
19. Записати в масив K(10) цифри заданого цілого числа Т.
20. Отримати значення цілочисельної змінної К, цифри якої записані порозрядно в масив Х.
21. Визначити кількість елементів масиву Т(50), що більше за своїх сусідів зліва та справа.
22. Елементи масиву зі 100 дійсних чисел розташувати у зворотному порядку.
23. Визначити кількість елементів цілочисельного масиву А(100), парність яких співпадає з парністю їх індексів.
24. Визначити кількість змін знаку в масиві дійсних чисел K(100). Під зміною знаку розуміється розташування поряд від'ємного та додатного значень, наприклад, в послідовності 17, 4, -5, -7, 8, -3 буде три зміни знака.
25. В цілочисельному масиві В(100) визначити кількість елементів, що дорівнюють макимальному.

Завдання 5.2

- ◆ Скласти математичну постановку задачі, схему алгоритму та програму для обробки двовимірного масиву даних згідно з варіантом завдання;
- ◆ описати підпрограму для введення елементів масиву за допомогою генератора випадкових чисел;
- ◆ описати підпрограму для виведення масиву на екран;
- ◆ * додати до програмного модуля (бібліотеки) з завдання 5.1 підпрограми введення/виведення матриці; під час введення передбачити можливість введення матриці вручну та за допомогою генератора.

◆

Рекомендації до виконання роботи

Використання генератора випадкових чисел

Генератор випадкових чисел – спеціальна бібліотечна підпрограма, яка повертає випадкове значення. В мові Паскаль робота генератора можлива в двох режимах: генерація випадкового цілого в межах від 0 до заданого значення (у прикладі 1 для змінної k буде згенероване число від 0 до 1000) або генерація дійсного в межах від 0 до 1 (приклад 1, мова Паскаль, змінна r) [2; 4]

В мові Сі на відміну від Паскаль є дві функції-генератора, але вони дають тільки цілі числа: функція `int random(int num)` дає випадкове ціле в межах від 0 до `num-1`; функція `int rand(void)` дає випадкове ціле в межах від 0 до `RAND_MAX=32767` [6-8].

Слід зауважити, що для використання функцій-генераторів у мові Сі необхідно підключити бібліотеку `<stdlib.h>`.

Перед тим як почати генерацію випадкових чисел необхідно провести ініціалізацію генератора (ввімкнути його), використавши процедуру мови Паскаль `randomize` [2; 4] або функцію `randomize()` мови Сі [6-8].

В програмі можна використати `randomize` тільки один раз!!!

Приклад 1. Використання генератора для отримання цілого та дійсного значень.

	<code>#include <stdlib.h></code>
<code>var k:integer;</code>	<code>int k; float r;</code>
<code> r:real;</code>	<code>main()</code>
<code>begin</code>	<code>{</code>
<code> randomize;</code>	<code> randomize();</code>
<code> k:=random(1000);</code>	<code> k= random(1000);</code>
<code> r:=random;</code>	<code> r= (float) rand() / 100.0;</code>
<code>end.</code>	<code>}</code>

Але ж виникає запитання: як отримати від'ємне значення? Для цього можна ще раз застосувати генератор. Справедливо вважати, що існує однакова вірогідність отримати додатне або від'ємне значення, тому використавши

* - це завдання для самостійного виконання за бажанням!!!

генератор ще раз, можна перевірити, якщо отримане значення більше 0,5, вважаємо число додатним, ні – від’ємним (або навпаки).

Приклад 2. Використання генератора для отримання від’ємних та додатних значень.

	#include <stdlib.h>
var k:integer;	int k;
	main()
begin	{
randomize;	randomize();
k:=random(1000);	k= random (1000);
if random < 0.5 then k:=-k ;	if (random (100) < 50) k=-k ;
end.	}

Варіанти завдання

1. В матриці $M(N, M)$ знайти суму елементів парних рядків.
2. В матриці $M(N, M)$ знайти суму від’ємних елементів.
3. В матриці $M(N, M)$ замінити елементи парних рядків на цифри від 0 до 9.
4. В матриці $M(N, M)$ поміняти місцями елементи 1-го та 2-го рядків, 3-го та 4-го і т.д.
5. В матриці $M(N, M)$ знайти суму елементів непарних рядків.
6. В матриці $M(N, M)$ всі елементи рядка, в якому знаходиться мінімальний елемент матриці замінити на протилежне значення за знаком.
7. В матриці $M(N, M)$ знайти суму елементів головної та бічної діагоналей.
8. В матриці $M(N, M)$ поміняти місцями елементи головної та бічної діагоналей.
9. В матриці $M(N, M)$ знайти суму від’ємних елементів непарних стовпчиків.
10. В матриці $M(N, M)$ поміняти місцями елементи рядків та стовпчиків.
11. Визначити чи $M(N, M)$ симетрична матриця $M(10 \times 10)$ відносно головної діагоналі.
12. В матриці $M(N, M)$ знайти суму елементів парних стовпчиків.
13. Для кожного рядка матриці $M(N, M)$ вивести повідомлення, якщо його елементи впорядковані за зростанням.
14. Для кожного стовпця матриці $M(N, M)$ вивести повідомлення, якщо його елементи впорядковані за спаданням.
15. Для кожного рядка матриці $M(N, M)$ вивести повідомлення, якщо його елементи симетричні.
16. Для кожного стовпця матриці $M(N, M)$ вивести повідомлення, якщо його елементи утворюють арифметичну прогресію.
17. Для кожного стовпця матриці $M(N, M)$ вивести повідомлення, якщо його елементи утворюють геометричну прогресію.
18. В матриці $P(N, M)$ знайти добуток елементів кожного рядка.
19. В матриці $P(N, M)$ знайти суму елементів кожного стовпця.
20. Визначити добуток двох матриць розміром $N \times N$.
21. Визначити добуток матриці $N \times N$ та масиву з N елементів.
22. Транспонувати матрицю $A(N, M)$.
23. Визначити визначник матриці $K(3, 3)$.
24. В матриці $X(N, N)$ визначити чи є вона магічним квадратом.
25. Визначити номер рядка матриці $C(N, N)$, сума елементів якого найбільша.

Завдання 5.3. Обробка рядків символів мови Паскаль

- ◆ Скласти математичну постановку задачі, схему алгоритму та програму для обробки рядків згідно з варіантом завдання;
- ◆ кожне завдання описати як окрему підпрограму без параметрів;
- ◆ в головній програмі організувати меню для вибору завдань або виходу з програми.

Рекомендації до виконання роботи

Рядок символів мови Паскаль описується типом *string* і являє собою спакований масив символів, тобто packed array [0..255] of char. Таким чином, кожен символ рядка має свій порядковий номер (індекс). Нульовий елемент містить загальний розмір (довжину) рядка (рис 5.1). [1; 2; 4; 9]

Var s: string;

S

s[0]	s[1]	s[2]	s[3]	s[4]	...	s[254]	s[255]
Chr(255)	a	b	c	@		\$	d

Рис. 5.1

Якщо в задачі немає необхідності використовувати всі 256 символів рядка, можна при описі задати іншу довжину:

Var s: string [10]; {рядок довжиною в 10 символів}

Не зважаючи на те, що рядок це масив, над рядками в мові Паскаль припустимо виконання таких операцій [1; 2; 4; 9]:

- присвоєння рядків однакової довжини (за описом);
- порівняння рядків однакової довжини;
- додавання (склеювання, конкатенація) рядків.

Крім того рядки вводяться та виводяться як прості змінні, а не в циклі як масиви.

Введення рядка	Виведення рядка
ReadLn(s);	WriteLn(s);

Певні проблеми можуть виникнути при введенні двох рядків поспіль. Якщо їх вводити в одному операторі ReadLn, наприклад, ReadLn(s1, s2), то в результаті вся інформація потрапить в перший рядок. Для того, щоб цього уникнути необхідно кожний рядок вводити в окремому операторі ReadLn (але не в Read):

WriteLn('Input s1 : '); ReadLn(s1);

WriteLn('Input s2 : '); ReadLn(s2);

В мові Паскаль існують декілька процедур та функцій обробки рядкової інформації, які детально описані в [1; 2; 4].

Треба зауважити, що функція пошуку підрядка в рядку **Pos** знаходить тільки перше входження підрядка в рядок. Тому для виконання 4-го завдання у варіантах необхідно:

1. Знайти перше входження підрядка;
2. Видалити його з рядка (та виконати інші необхідні дії згідно з варіантом завдання);
3. Повернутися до пошуку наступного входження.

Зрозуміло, що йдеться про організацію циклічного процесу:

While pos (.....,.....) <> 0 do { *доки pos знаходить підрядок* }

Begin End;

Варіанти завдань 5.3

№	Завдання
1	Заданий рядок S. Перетворити S наступним чином: 1) видалити з S перше входження буквосполучення 'read'; 2) після першого буквосполучення 'tr' вставити 'ue'; 3) в рядок X скопіювати 5 символів з рядка S, починаючи з 2 позиції; 4) всі входження 'ed' замінити на 'ing'.
2	Заданий рядок S. Перетворити S наступним чином: 1) видалити з S перше входження буквосполучення 'do'; 2) після першої букви 'e' вставити буквосполучення 'nd'; 3) в рядок X скопіювати 4 символи з рядка S, починаючи з середини рядка 4) замінити всі буквосполучення 'it ' на 'is';
3	Заданий рядок S. Перетворити S наступним чином: 1) видалити з S перше входження буквосполучення 'are'; 2) після першої букви 'a' вставити буквосполучення 'bc'; 3) в рядок X скопіювати 6 символи з рядка S, починаючи з позиції наступної за першою буквою 'a' 4) видалити з S кожен символ '*'
4	Заданий рядок S. Перетворити S наступним чином: 1) видалити з S перше входження буквосполучення 'we' 2) після першого буквосполучення 'el' вставити 'se'; 3) в рядок X скопіювати 3 символи з рядка S, починаючи з 5 позиції 4) всі входження 'km' замінити на 'cn'
5	Заданий рядок S. Перетворити S наступним чином: 1) видалити з S перше входження буквосполучення 'red'; 2) після першого буквосполучення 'th' вставити 'e'; 3) в рядок X скопіювати 5 символів з рядка S, починаючи з 6 позиції; 4) видалити з рядка усі коми та крапки.
6	Заданий рядок S. Перетворити S наступним чином: 1) видалити з S перше входження буквосполучення 'var' 2) після першого буквосполучення 'th' вставити 'en'; 3) в рядок X скопіювати всі символи з рядка S, починаючи з середини рядка 4) підрахувати кількість буквосполучень 'sh'
7	Заданий рядок S. Перетворити S наступним чином: 1) видалити з S перше входження буквосполучення 'they'; 2) після першої букви 's' вставити буквосполучення 'tring'; 3) в рядок X скопіювати 6 останніх символів з рядка S; 4) після кожної букви 'w' додати букву 'q'.

№	Завдання
8	Заданий рядок S. Перетворити S наступним чином: 1) видалити з S перше входження буквосполучення 'for'; 2) після першої букви 'r' вставити 'eal'; 3) в рядок X скопіювати 4 символи з рядка S, починаючи з 5 позиції; 4) видалити з S всі входження 'abc'.
9	Заданий рядок S. Перетворити S наступним чином: 5) видалити з S перше входження буквосполучення 'while'; 6) після першої букви 'c' вставити 'har'; 7) в рядок X скопіювати 5 символів з рядка S, починаючи з 3 позиції; 8) видалити з S всі входження 'qwerty'.
10	Заданий рядок S. Перетворити S наступним чином: 1) видалити з S перше входження 'true'; 2) після першої букви 'i' вставити буквосполучення 'ng'; 3) в рядок X скопіювати 5 символів з рядка S, починаючи з 6 позиції ; 4) замінити всі 'x' на буквосполучення 'ks'.
11	Заданий рядок S. Перетворити S наступним чином: 1) видалити з S перше входження 'case'; 2) після першої букви 'p' вставити букву 'e'; 3) в рядок X скопіювати 4 символи з рядка S, починаючи з середини рядка; 4) після кожної букви 'i' додати буквосполучення 'am'.
12	Заданий рядок S. Перетворити S наступним чином: 1) видалити з S перше входження буквосполучення 'no'; 2) після першої 'begin' вставити 'end'; 3) в рядок X скопіювати 6 символів з рядка S, починаючи з 2 позиції ; 4) видалити з рядка всі входження буквосполучення 'xyz'.
13	Заданий рядок S. Перетворити S наступним чином: 1) видалити з S перше входження буквосполучення 'end'; 2) після першої букви 't' вставити буквосполучення 'rue'; 3) в рядок X скопіювати 6 символів з рядка S, починаючи з першої букви w ; 4) всі входження 'ed' замінити на 'ing'.
14	Заданий рядок S. Перетворити S наступним чином: 1) видалити з S перше входження буквосполучення 'begin'; 2) після першої букви 'e' вставити буквосполучення 'nd'; 3) в рядок X скопіювати 4 символи з рядка S після першого пропуску ; 4) замінити всі буквосполучення 'er' на 's'.
15	Заданий рядок S. Перетворити S наступним чином: 1) видалити з S перше входження буквосполучення 'const'; 2) після першої букви 'v' вставити буквосполучення 'ar'; 3) в рядок X скопіювати 7 символів з рядка S, починаючи з 10 позиції ; 4) замінити всі буквосполучення 'were' на 'are'.
16	Заданий рядок S. Перетворити S наступним чином: 1) видалити з S перше входження буквосполучення 'while'; 2) після першої букви 'f' вставити буквосполучення 'or'; 3) в рядок X скопіювати 6 символів з рядка S після першої крапки; 4) підрахувати кількість буквосполучень 'ing'.

№	Завдання
17	Заданий рядок S. Перетворити S наступним чином: 1) видалити з S перше входження 'type'; 2) після першої букви 'w' вставити буквосполучення 'hile'; 3) в рядок X скопіювати 3 символи з рядка S, починаючи з 4 позиції; 4) видалити з S всі входження букв 'm' та 'f'.
18	Заданий рядок S. Перетворити S наступним чином: 1) видалити з S перше входження буквосполучення 'int'; 2) після першого буквосполучення 'un' вставити 'til'; 3) в рядок X скопіювати 3 символи з рядка S після першої коми ; 4) повторити кожен символ 'a'.
19	Заданий рядок S. Перетворити S наступним чином: 1) видалити з S перше входження буквосполучення 'er'; 2) після першого буквосполучення 'un' вставити 'til'; 3) в рядок X скопіювати 4 символи з рядка S після першої 'q'; 4) підрахувати кількість буквосполучень 'sh'.
20	Заданий рядок S. Перетворити S наступним чином: 1) видалити з S перше входження буквосполучення 'tr'; 2) після першого буквосполучення 'func' вставити 'tion'; 3) в рядок D скопіювати 3 символи з рядка S після першої 'k'; 4) підрахувати кількість буквосполучень 'th'.
21	Заданий рядок S. Перетворити S наступним чином: 1) видалити з S перше входження буквосполучення 'in'; 2) після першого буквосполучення 'co' вставити 'de'; 3) в рядок Z скопіювати 5 символи з рядка S після першої 't'; 4) підрахувати кількість буквосполучень 'ro'.
22	Заданий рядок S. Перетворити S наступним чином: 1) видалити з S перше входження буквосполучення 'ng'; 2) після першого буквосполучення 'tu' вставити 'rn'; 3) в рядок X скопіювати 2 символи з рядка S після першої 'q'; 4) підрахувати кількість буквосполучень 'ght'.
23	Заданий рядок S. Перетворити S наступним чином: 1) видалити з S перше входження буквосполучення 'not'; 2) після першого буквосполучення 'so' вставити 'on'; 3) в рядок X скопіювати 4 символи з рядка S після першої 'j'; 4) підрахувати кількість буквосполучень 're'.
24	Заданий рядок S. Перетворити S наступним чином: 1) видалити з S перше входження буквосполучення 'fun'; 2) після першого буквосполучення 'li' вставити 'fe'; 3) в рядок X скопіювати 4 символи з рядка S після першої 'd'; 4) підрахувати кількість буквосполучень 'vo'.
25	Заданий рядок S. Перетворити S наступним чином: 1) видалити з S перше входження буквосполучення 're'; 2) після першого буквосполучення 'go' вставити 'od'; 3) в рядок X скопіювати 3 символи з рядка S після першої 's'; 4) підрахувати кількість буквосполучень 'le'.

ЛАБОРАТОРНА РОБОТА №6.

Обробка структурованих типів даних

Завдання

- ◆ скласти математичну постановку задачі, схему алгоритму та програму для обробки записів (структур даних) згідно з варіантом завдання;
- ◆ в програмі описати тип запису (структуру) для свого завдання (стовбець «Дані») та масив з N записів (структур) цього типу. Розробити набір тестових даних до 10 записів.
- ◆ кожне завдання (варіанти завдань, стовбець «Завдання») описати як окрему підпрограму (процедуру або функцію), передавши масив записів як параметр;
- ◆ вивести на екран дані з масиву в табличному вигляді (оформити як підпрограму з масивом - параметром). Границі таблиці на екрані можна задати символами ' * ' або символами псевдографіки;
- ◆ після виведення таблиці з вхідними даними вивести результати роботи програми;
- ◆ *додати до програми підпрограми для збереження даних в файл та зчитування даних з файлу (Паскаль – типізований файл, Сі – звичайний текстовий).

◆

Рекомендації до виконання роботи

До структурованих типів даних (СТД) належать тип запис `record` мови Паскаль та тип `struct` мови Сі.

Розглянемо приклад опису та обробки СТД для списку студентів. По кожному студенту задається прізвище, номер залікової книжки та оцінка з певної дисципліни в балах від 0 до 100. Подамо необхідні дані у вигляді таблиці:

Прізвище студента	Номер залікової книжки	Оцінка
Іваненко П.П.	1402	97

Шапка цієї таблиці складається з трьох стовпців (полів), тобто і СТД буде мати три поля. Залишилося визначитися з ідентифікатором та типом кожного поля:

	Прізвище студента	Номер залікової книжки	Оцінка
Ідентифікатор	<i>PIB</i>	<i>Nomer</i>	<i>Ocinka</i>
Тип (Pascal)	<i>string [20]</i>	<i>word</i>	<i>byte</i>
Тип (C)	<i>char[20]</i>	<i>unsigned int</i>	<i>unsigned short</i>

Найчастіше при роботі із записом описують спочатку тип запису, а потім цей тип використовують для опису змінних.

* - завдання виконується самостійно за бажанням.

<i>Pascal</i>	<i>C</i>
Type	
StudentType = record	struct StudentType {
PIB : string[20];	char PIB [20];
Nomer : word;	unsigned int Nomer;
Ocinka : byte ;	unsigned short Ocinka;
end ;	};

Тепер можна описати змінну:

<i>Pascal</i>	<i>C</i>
Var	
Student : StudentType;	struct StudentType student;

Для того щоб полям змінної STD присвоїти значення, необхідно звертатися окремо до кожного поля, використовуючи оператор розкриття запису (структури) '.' (крапка). Загальний синтаксис:

зміннаSTD. поле

<i>Pascal</i>	<i>C</i>
Student. PIB:='Іваненко П.П.';	strcpy(student.PIB, "Іваненко П.П.");
Student. Nomer:= 1402;	student. Nomer = 1402;
Student. Ocinka:= 97;	student. Ocinka = 97;

Тепер змінна Student зберігає дані про студента Іваненко П.П., у якого номер заліковки 1402, і який отримав оцінку 97 балів.

Слід зауважити, що в мові Сі для присвоєння рядку певного значення необхідно застосовувати спеціальну функцію:

strcpy (рядок-приймач, рядок-джерело);

бібліотеки <strings.h>, яка виконує копіювання вмісту з рядка-джерела до рядка-приймача.

Присвоєння даних запису Student з використанням оператора приєднання мови Паскаль (в мові Сі оператора приєднання не існує!) буде виглядати так:

with Student **do**

begin

 PIB:='Іваненко П.П.';

 Nomer:= 345;

 Ocinka:= 5;

end;

Використання масиву записів

З використанням вищеописаного типу можна задати масив записів для збереження даних про студентську групу:

Варіанти завдань

№	Дані	Завдання
1	Дані про вартість міжнародних телефонних переговорів (прізвище абонента, кількість хвилин, вартість)	загальну вартість розмов; прізвище абонента, який розмовляв за найнижчим тарифом
2	Відомості про товари на складі (назва товару, ціна за одиницю, кількість одиниць)	найдорожчу партію товару; загальну вартість товарів на складі
3	Дані для обліку ліків в аптеці (назва, кількість упаковок, інтенсивність продажу за день)	назви ліків, запас яких залишився на К днів; середню інтенсивність споживання ліків
4	Відомості про операторів мобільного зв'язку (назва оператора, код, вартість хвилини в мережі)	середню вартість хвилини розмови; відомості про операторів, вартість хвилини у яких не більша за Р
5	Результати здачі вступних іспитів до інституту (прізвище абітурієнта, оцінка з математики та диктанту)	прізвища абітурієнтів, середній бал яких більший за В; кількість абітурієнтів, що не склали іспити
6	Відомості про проведення спортивних змагань (команда1, очки, команда2, очки)	команди-переможці; загальну кількість набраних очок
7	Відомості про результати проведення олімпіади з програмування (прізвище студента, кількість розв'язаних задач, кількість набраних балів)	прізвища студентів, що зайняли призові місця
8	Відомості про пропуски студентами занять (прізвище студента, кількість пропущених лекцій та практичних)	Прізвища студентів, що пропустили кількість занять більшу за К; загальну кількість пропущених занять
9	Відомості про розклад руху потягів (№ потяга, пункт призначення, час відправлення)	номери потягів, що прямують до пункту Р; кількість потягів, що відправляються вночі
10	Відомості про графік проведення гуртків (назва гуртка, день тижня, час)	кількість гуртків, що проводяться в заданий день; назви гуртків, що проводяться до 12 годин дня
11	Відомості про обсяг навчального часу з дисципліни (назва дисципліни, кількість лекцій, лабораторних та практичних занять)	загальний обсяг навчального часу по дисциплінам; назви дисциплін, у яких кількість лекцій більша за загальну кількість практичних занять
12	Відомості про дисплейні класи (№ класу, прізвище старшого оператора, кількість комп'ютерів)	номери класів, кількість комп'ютерів в яких більша за К; прізвища операторів, що закріплені за декількома класами
13	Відомості про характеристики	характеристики машин, обсяг

№	Дані	Завдання
	пральних машин (назва, обсяг завантаження, частота обертів барабану, ціна)	завантаження та ціна яких не більше заданих; середню вартість машин
14	Відомості про спеціальності університету(назва спеціальності, кількість студентів, форма навчання(очна, заочна))	дані про спеціальність з кількістю студентів більшою за K; середню кількість студентів по спеціальностях
15	Дані метеорологічного спостереження за тиждень (день, температура повітря, атмосферний тиск)	середню температуру та тиск за тиждень; день, коли температура повітря чи тиск перевищували середнє значення
16	Дані про житлову площу квартир будинку(№ квартири, кількість кімнат, площа)	середню площу кімнати в будинку; дані про квартири, середня площа кімнат в яких більша за отримане середнє значення по будинку
17	Дані про маршрути громадського транспорту(номер маршруту, назва, інтервал руху)	середнє значення інтервалу; дані про маршрути, інтервал руху яких менший за середнє значення
18	Дані про розташування телевізійних ретрансляторів (номер ретранслятора, географічні координати, радіус дії)	дані про ретранслятори, область дії яких перетинається
19	Відомості про туристичні путівки (назва країни, кількість путівок, вартість однієї)	назви країн, вартість путівки до яких не перевищує заданого значення P; загальну вартість всіх путівок
20	Відомості про постачальників сировини одного виду (назва фірми-постачальника, вартість одиниці, обсяг поставки)	дані постачальників, вартість сировини у яких найменша; загальний обсяг та вартість придбаної сировини
21	Відомості про посадові оклади працівників(назва посади, кількість місць, оклад, коефіцієнт надбавки за шкідливість роботи)	назви посад, на яких з/п (з врахуванням надбавки) більша за T; загальний обсяг з/п
22	Дані про роботу Internet-кафе (назва, час відкриття та закриття, вартість послуг)	дані про кафе з найбільшою тривалістю роботи; середню вартість послуг
23	Відомості про ріки країни (назва, довжина, площа басейну)	дані про ріки, довжина яких більша за P; загальну площу водного басейну країни
24	Відомості про державні свята (назва свята, день, місяць, кількість вихідних днів)	загальну кількість святкових днів; дані про свята за заданим місяцем
25	Дані про студентів, що мають академічні заборгованості(прізвище студента, кількість нескладених	дані про студентів з найбільшою академ. заборгованістю; середню кількість незданих іспитів та заліків

№	Дані	Завдання
	іспитів та заліків)	

ЛАБОРАТОРНА РОБОТА №7. Обробка типу множина мови Паскаль

Завдання

- скласти математичну постановку задачі, схему алгоритму та програму мовою Паскаль для обробки множин;
- описати тип даних для множини згідно до варіанта;
- описати дві змінні-множини та провести їх ініціалізацію за допомогою конструктору множини;
- в головній програмі (або у підпрограми) для знаходження перетину, об'єднання, різниці двох множин, перевірки на включення однієї множини до іншої та перевірки на включення до множини заданого елемента;
- *описати підпрограми для введення та виведення елементів множин. На початку роботи програми надати користувачу можливість обрати введення елементів множин вручну або автоматично.

Рекомендації до виконання роботи

Для зручності розробки програми необхідно описати тип множини, а потім змінну-множину цього типу, наприклад:

Type

```
Тип_множина = set of діапазон_значень_елементів;
Digits = set of '0'..'9';
AlphaBet = set of 'A'..'Z';
Double_dig= set of 10..99;
```

Var

```
D:Digit;
A:AlphaBet;
Dig:Double_dig;
```

При описі типу елементів множини необхідно пам'ятати, що:

- базовим типом при завданні діапазону значень для елементів може бути беззнаковий цілий або символний тип даних;
- загальна кількість елементів множини не може перевищувати 256;
- при описі типу множини в якості діапазону значень елементів припустимо використовувати типи byte та char, наприклад:

Type

```
Set_1 = set of byte;
Set_2 = set of char;
```

Ініціалізація змінних-множин та операції над множинами детально описані в [1; 2; 4; 9].

* - це завдання виконується самостійно за бажанням.

Слід пам'ятати, що операції виконуються тільки над множинами одного типу!!! Наприклад:

Type

Set_B = **set of** byte;
Set_0 = **set of** 0..255;

Var

T, K : Set_B;
Q, P : Set_0;

Над множинами T і K (або Q і P) можливе застосування операцій, а над множинами T і Q (або K і P, K і Q, T і P) – **ні**, не зважаючи навіть на те, що за змістом опису типів вони однакові!

Введення/виведення елементів змінних-множин

Введення елементів змінних-множин можна організувати так:

1. Запросити у користувача кількість елементів в множині;
2. За заданою кількістю організувати цикл, в якому вводити змінну, тип якої відповідає базовому типу елемента множини. Якщо множина була описана певним діапазоном, то необхідно перевірити, чи належить значення введеної змінної заданому діапазону, якщо ні, то вивести про це повідомлення. Інакше включити введену змінну до множини операцією include.

Виведення елементів змінних-множин можна організувати так: в циклі по всіх можливих елементах множини перевіряти, якщо черговий елемент належить множині, вивести його на екран.

Оформлення звіту

У звіті наводиться:

- 1) завдання та варіант завдання;
- 2) математична постановка задачі для операцій над множинами з тестовим прикладом;
- 3) аналіз результатів роботи програми;
- 4) текст програми.

Варіанти завдань

№	Завдання
1	Описати тип для множини, що включає цифри від '0' до '9'
2	Описати тип для множини, що включає букви від 'A' до 'J'
3	Описати тип для множини, що включає букви від 'K' до 'S'
4	Описати тип для множини, що включає букви від 'T' до 'Z'
5	Описати тип для множини, що включає символи від '!' до '/'.
6	Описати тип для множини, що включає англійські букви від 'a' до 'i'
7	Описати тип для множини, що включає англійські букви від 'j' до 't'

№	Завдання
8	Описати тип для множини, що включає англійські букви від 's' до 'z'
9	Описати тип для множини, що включає українські букви від 'А' до 'З'
10	Описати тип для множини, що включає українські букви від 'К' до 'У'
11	Описати тип для множини, що включає українські букви від 'Ф' до 'Я'
12	Описати тип для множини, що включає українські букви від 'а' до 'з'
13	Описати тип для множини, що включає українські букви від 'к' до 'п'
14	Описати тип для множини, що включає українські букви від 'р' до 'щ'
15	Описати тип для множини, що включає числа від 1 до 31. Задати дві множини: одна – святкові дні січня цього року, друга – вихідні дні (субота, неділя) січня цього року.
16	Описати тип множини для трьохзначних цілих. Задати дві множини: одна – існуючі коди операторів мобільного зв'язку (не враховуючи перший 0), друга – коди операторів, якими користуються члени Вашої родини (друзі).
17	Описати тип множини для цілих. Задати дві множини: одна – степені 2 від 0 до 16-ї, друга – степені 4 від 0 до 8-ї.
18	Описати тип множини для цілих. Задати дві множини: одна – святкові дні травня цього року, друга – робочі дні травня (понед. – п'ятн.) не залежно від того випадає він на святковий день чи ні.
19	Описати тип множини для цілих. Задати дві множини: одна – результати таблиці множення на 3, друга – результати таблиці множення на 6.
20	Описати тип множини для цілих. Задати дві множини: одна – високосні роки 20 сторіччя (перший високосний 1904), друга – високосні роки 21 сторіччя (перший високосний 2004).
21	Описати тип множини для символів. Задати дві множини: одна – знаки арифметичних операцій мови Паскаль (операції mod, div позначити першими буквами), друга – знаки арифметичних операцій мови Сі.
22	Описати тип множини для символів. Задати дві множини: одна – всі дужки, що можуть використовуватися в мові Паскаль, друга – дужки мови Сі.
23	Описати тип множини для символів. Задати дві множини: одна – букви, з яких починаються прізвища студентів Вашої групи, друга – букви, з яких починаються прізвища студентів паралельної групи.
24	Описати тип множини для символів. Задати дві множини: одна – перші букви з міжнародних марок автомобілів східних виробників, друга – перші букви з міжнародних марок автомобілів західних виробників.
25	Описати тип множини для цілих. Задати дві множини: одна – двознакові парні, друга – степені двійки від 0 до 10.

ЛАБОРАТОРНА РОБОТА №8. Динамічний розподіл пам'яті

Завдання 8.1. Типізовані та нетипізовані покажчики

Скласти програму, в якій:

- ◆ описати типізовані покажчики на типи, що визначені варіантом завдання;
- ◆ виділити пам'ять під покажчики для збереження значень;
- ◆ записати у виділені області пам'яті певні значення (спочатку присвоєнням, а потім – введенням з клавіатури);
- ◆ обчислити суму заданих значень та зберегти в новій області пам'яті, визначивши типізований покажчик на неї;
- ◆ виконати аналогічні дії для нетипізованих покажчиків.

Рекомендації до виконання роботи

Покажчик – змінна, що зберігає адресу області (комірки) в оперативній пам'яті. Змінна-покажчик (ЗП) може бути типізованою чи нетипізованою.

Паскаль	Cі
<u>Типізований покажчик</u>	
Var	
ім'я_ЗП : ^тип;	тип* ім'я_ЗП;
p : ^integer;	int* p;
<u>Нетипізований покажчик</u>	
Var	
ім'я_ЗП : pointer;	void* ім'я_ЗП;
t : pointer;	void * t;

В наведеному вище прикладі змінна-покажчик p – типізований покажчик. Це означає, що вона може зберігати адресу області пам'яті, в яку записане тільки цілочисельне значення (при описі змінної заданий цілочисельний тип даних).

Змінна-покажчик t – нетипізований покажчик, тобто ця змінна зберігає адресу області пам'яті, в яку записані дані невизначеного типу. Але відомо, що всі дані повинні бути описані певним типом, тому при роботі з нетипізованим покажчиком необхідно завжди!!! явно приводити тип (див. відповідний розділ нижче).

Виділення пам'яті для покажчиків

Після опису змінної-покажчика (ЗП) її значення невизначене, адже під час опису, ніяку адресу в цю область не було записано. Отже, необхідно виділити область пам'яті та записати адресу цієї області в ЗП.

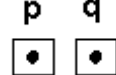
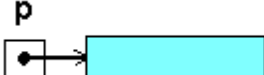

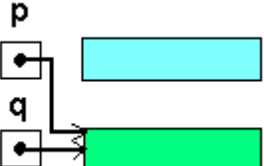
Паскаль	Cі
<u>Типізований покажчик</u>	
New (ім'я_ЗП);	ім'я_ЗП = (тип*)malloc(sizeof(тип));
New (p);	p = (int*)malloc(sizeof(int));
<u>Нетипізований покажчик</u>	
GetMem (ім'я_ЗП, розмір області);	ім'я_ЗП = malloc(розмір області);
GetMem (t, sizeof(real));	t = malloc(sizeof(float));

В мові Паскаль для виділення пам'яті під типізований та нетипізований покажчики використовуються різні процедури (New та GetMem). В мові С функція виділення пам'яті однакова для нетипізованих та типізованих покажчиків. При цьому функція malloc орієнтована на роботу з нетипізованими покажчиками, тому під час виділення пам'яті під типізовані покажчики необхідно явно приводити тип результату malloc до типу покажчика.

Крім того другим параметром malloc (а також GetMem в Pascal) є розмір області пам'яті, яку необхідно виділити. У наведеному вище прикладі виділяється область пам'яті розміром рівним до розміру дійсного значення типу real (float).

Розкриття посилання

Розкриття посилання забезпечує доступ до вмісту області пам'яті, на яку посилається покажчик. Розглянемо приклад:

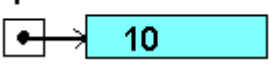
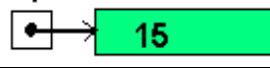
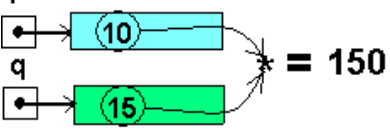
Паскаль	Сі	Графічна інтерпретація
Var p, q: ^integer;	int* p, q;	
begin	p = (int*)malloc(sizeof(int));	
New(p); New(q);	q = (int*)malloc(sizeof(int));	
p := q;	p = q;	
end.		

В наведеному прикладі описані два типізовані покажчики p та q. Для них виділені області пам'яті однакового розміру. Після цього відбувається присвоєння p = q, яке призведе до того, що до змінної p буде записана таж сама адреса, що зберігається в змінній q (при цьому адресу, що зберігалася в p буде втрачено!).

При роботі з покажчиками необхідно чітко розрізняти, коли ведеться робота з адресами виділених областей пам'яті, а коли з вмістом виділених областей пам'яті (тобто з даними, на які посилаються покажчики). Щоби отримати доступ до області пам'яті, на яку посилається покажчик, необхідно застосувати операцію розкриття посилання:

Паскаль	Сі
ім'я_ЗП^	*ім'я_ЗП
p^	*p

Розглянемо приклад, в якому в виділені області пам'яті для покажчиків p та q записуються певні значення, а їх добуток виводиться на екран.

Паскаль	Cі	Графічна інтерпретація
p:=10;	*p = 10;	p 
q:=15;	*p = 15;	q 
WriteLn('Добуток=', p*q);	printf("Добуток=%d", (*p)*(*q));	p q 

Приведення типів для нетипізованих покажчиків

При роботі з нетипізованими покажчиками під час розкриття посилання необхідно явно приводити тип (t – нетипізований покажчик з попереднього розділу):

Паскаль	Cі
тип(ім'я_ЗП^)	*(тип_ЗП)ім'я_ЗП
real(t^)	*(float*)t

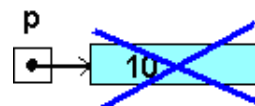
В мові Паскаль та Сі підхід до явного приведення типів нетипізованих покажчиків різняться:

- в мові Паскаль спочатку розкривається посилання (отримується доступ до вмісту області пам'яті), а потім вміст області пам'яті приводиться до необхідного типу;
- в мові Сі спочатку нетипізований покажчик приводиться до типізованого покажчика, а потім розкривається посилання.

Звільнення пам'яті для покажчиків

Після того, як «всі роботи» з використанням покажчика закінчені, **обов'язково необхідно звільнити** виділену область пам'яті:

Паскаль	Cі
<u>Типізований покажчик</u>	
Dispose (ім'я_ЗП);	free(ім'я_ЗП);
Dispose (p);	free(p);
<u>Нетипізований покажчик</u>	
FreeMem (ім'я_ЗП, розмір області);	free(ім'я_ЗП);
FreeMem (t, sizeof(real));	free(t);



Оформлення звіту

В звіті для цього завдання, а також для наступних завдань 8.2-8.3 необхідно навести:

- завдання до лабораторної роботи;
- текст програми з графічною інтерпретацією кожного оператора, в якому є звернення до показчиків.
-

Варіанти завдань

№	p1	p2	№	p1	p2
1	byte,	word,	2	integer,	byte,
3	word,	short,	4	longint,	word,
5	short,	integer,	6	real,	short,
7	integer,	longint,	8	byte,	longint,
9	longint,	real,	10	word,	real,
11	real,	byte,	12	short,	byte,
13	byte,	short,	14	integer,	word,
15	word,	integer,	16	longint,	short,
17	short,	longint,	18	real,	integer,
19	integer,	real,	20	byte,	real,
21	longint,	byte,	22	word,	byte,
23	real,	word,	24	short,	word,
25	byte,	integer,	26	integer,	short,
27	word,	longint,	28	longint,	integer,
29	short,	real,	30	real,	longint,

Завдання 8.2. Показчики та структуровані типи даних

Скласти програму, в якій описати типізований та нетипізований показчик на тип запис (структуру) з лабораторної роботи 6, виділити пам'ять, ініціалізувати поля запису (структури), вивести їх значення на екран.

Рекомендації до виконання роботи

Опис показчика на структурований тип даних (запис мови Паскаль або структуру мови Сі) виконується наступним чином:

Паскаль	Сі
<u>Опис структурованого типу</u>	
Type	
TSomeRecord = record	struct Some {
A:integer;	int A;
B:real;	float B
end;	};
<u>Типізований показчик</u>	
p : ^ TSomeRecord;	struct Some *p;

Виділення пам'яті для запису (структури) здійснюється аналогічно до будь-якого іншого стандартного типу даних (див. завдання 8.1).

Доступ до полів запису (структури) виконується наступним чином:

- спочатку розкривається посилання, в наслідок чого отримується доступ до області пам'яті, де зберігаються дані полів запису (структури);
- через операцію '.' вказується поле структури, значення якого береться.

Слід зауважити, що в мові Сі доступ до полів структур можна здійснити двома способами: через розкриття посилання та оператор '.' або за допомогою оператора '->' (стрілка).

Паскаль	Сі
<i>Типізований покажчик</i>	
p^.A := 10;	*p.A = 10; або так p->A = 10;
p^.B := 0.5;	*p.B = 0.5; або так p->B = 0.5;

При роботі з нетипізованими покажчиками доступ до полів здійснюється аналогічно але з обов'язковим явним приведенням типу покажчика (див. завдання 8.1).

Завдання 8.3.Обробка рядків символів в Сі-програмах

Скласти програму мовою Сі для обробки рядків символів, використавши завдання з ЛР№5.3. В роботі використати функції бібліотеки <strings.h>.

Рекомендації до виконання роботи

В мові Сі рядок – це масив символів, як і в мові Паскаль. А масиви в мові Сі нерозривно пов'язані з покажчиками. Тому при роботі з рядками в мові Сі необхідне явне використання покажчиків на символний тип.

Функції роботи з рядками бібліотеки <strings.h> детально розглянуто в [5-8;10]. Але в мові Сі не передбачено функцій видалення частини рядка (підрядка) або вставки підрядка в рядок. Тому виконання цих операцій необхідно здійснювати в декілька етапів та застосовувати змінні-покажчики.

Видалення підрядка з рядка

Нехай в програмі описані рядки s1, s2 та покажчик p:

```
char s1[10]="abcdef",    s2[10]="de";
char *p;
```

Для того щоби видалити підрядок s2 з s1 спочатку необхідно перевірити, чи входить підрядок s2 до s1, і зберегти адресу першого входження в покажчик p:

```
p=strstr(s1, s2);
```

Якщо функція strstr знайшла підрядок, повертається ненульове значення адреси. Отже для видалення підрядка s2 залишиться скопіювати частину рядка s1, що розташована після входження s2 («хвіст» рядка):

```

if (p) {
strcpy(p, p+strlen(s2));

printf("\n%s\n", s1);

} else printf("\nNo such substring!!!\n");

```



Слід зауважити, що для того щоби видалити з рядка всі входження підрядка, необхідно організовувати таке видалення в циклі аналогічно до використання функції **pos** мови Паскаль (див. завдання 5.3 відповідної лабораторної роботи).

Вставка підрядка в рядок

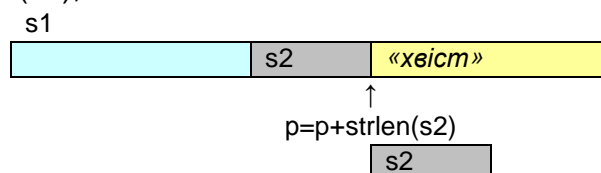
Розглянемо приклад, коли в рядок s1 після підрядка s2, він додається ще раз.

Для виконання вставки необхідно буде використати додатковий проміжний рядок s3:

```
char s3[10]="\0";
```

По-перше необхідно перевірити, чи входить підрядок s2 в s1 та визначити адресу, з якої починається «хвіст».

```
p=strstr(s1, s2)+strlen(s2);
```

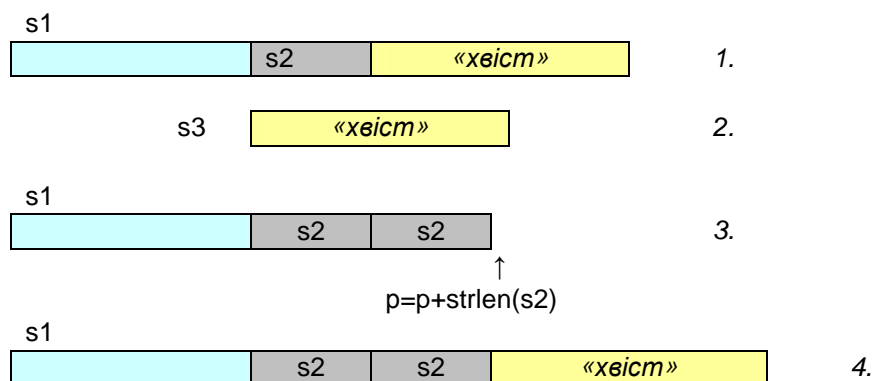


Якщо підрядок s2 знайдений, і нова довжина рядка з додаванням ще одного підрядка не перевищує задану максимальну довжину в 10 символів, можна виконувати додавання.

```

if (p && (strlen(s1)+strlen(s2))<10) {
strcpy(s3,p); /* 1.копіювання «хвоста» в проміжний рядок */
strcpy(p, s2); /* 2.копіювання на місце «хвоста» підрядка s2 */
p=p+strlen(s2); /*3.переведення покажчика в кінець нового підрядка s2 */
strcpy(p, s3); /*4. додавання «хвоста» */
printf("\n%s\n",s1); }
else printf("\nNo such string or length is very much!!!\n");

```



Слід зауважити, що наведені способи реалізації видалення та вставки підрідків не є єдиними можливими, але, на думку автора, є достатньо ефективними.

Рекомендована література

Мова Паскаль

1. Глинський Я.М., Анохін В.Є., Ряжська В.А. Паскаль. Turbo Pascal і Delphi: учеб. посібн.: 3-є вид. – Львів: Деон, 2002 – 144 с.
2. Ковалюк Т.В. Основи програмування. – К.: Видавнича група BHV, 2005. – 384 с.
3. Пильщиков В.Н. Сборник упражнений по языку Паскаль: учеб. пособие для вузов. - М.: Наука; Гл. ред. физ.-мат. лит., 1989. - 160 с.
4. Фаронов В.В. Турбо Паскаль 7.0. Начальный курс. учеб. пособие – М.: Нолидж, 1997- 616 с., ил.

Мова C/C++

5. Глинський Я.М., Анохін В.Є., Ряжська В.А. C++ і C++Builder. – Львів: Деон, 2003. – 193 с.
6. Керниган Б., Ритчи Д. Язык программирования Си/Пер. с англ.,— 4-е изд. — М.: Вильямс, 2007. — С. 304.
7. Ковалюк Т.В. Алгоритмізація та програмування: Підручник. – Львів: «Магнолія 2006», 2013. – 400 с.
8. Шпак З.Я. Програмування мовою C. – Львів: Оріяна-Нова, 2006. – 432 с.

Методична література

9. Красовська Г.В. Основи програмування та алгоритмічні мови. Мова програмування Паскаль. Основні поняття та управляючі конструкції: Конспект лекцій, ч.1 / Красовська Г.В. - К.:КНУБА, 2003 - 64 с.
10. Красовська Г.В. Основи програмування та алгоритмічні мови. Мова програмування Сі [Електронний ресурс]: Конспект лекцій, ч.2 / Г.В. Красовська - К.:КНУБА, 2008 - 54 с. – Режим доступу: <http://org.knuba.edu.ua/mod/resource/view.php?id=2151> (дата звернення 26.11.2014)
11. Красовська Г.В. Основи програмування та алгоритмічні мови [Електронне видання]: Методичні вказівки до лабораторних робіт/ Красовська Г.В. – К.:КНУБА, 2012. – 80 с. – Режим доступу: <http://org.knuba.edu.ua/mod/resource/view.php?id=2711> (дата звернення 26.11.2014)
12. Красовська Г.В. Основи програмування та алгоритмічні мови: методичні вказівки до вивчення дисципліни / Красовська Г.В. – К.:КНУБА, 2007. – 52 с.
13. Красовська Г.В. Основи програмування та алгоритмічні мови: методичні вказівки до практичних занять / Красовська Г.В. – К.:КНУБА, 2007. – 40 с.
14. Красовська Г.В. Основи програмування та алгоритмічні мови: методичні вказівки до виконання індивідуального завдання та курсової роботи / Красовська Г.В. – К.:КНУБА, 2007. – 60 с.

Для нотаток

Навчально-методичне видання

АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМУВАННЯ

Методичні вказівки
до виконання лабораторних робіт
для студентів, які навчаються за напрямом підготовки
6.050101 «Комп'ютерні науки»

Укладач **КРАСОВСЬКА** Ганна Валеріївна

Комп'ютерне верстання *А.П. Морозюк*

Підписано до друку Формат 60 × 84 ^{1/16}
Ум. друк. арк. 3,72 Обл.-вид. арк. 4,0
Електронний документ. Вид. № 1/III-15
Видавець і виготовлювач
Київський національний університет будівництва і архітектури

Повітрофлотський проспект, 31, Київ, Україна, 03680
Е-mail: red-isdat@ukr.net, тел. (044)241-54-22, 241-54-87

Свідоцтво про внесення до Державного реєстру суб'єктів

Видавничої справи ДК № 808 від 13.02.2002 р.