

**МІНІСТЕРСТВО ВНУТРІШНІХ СПРАВ УКРАЇНИ**

**Харківський національний університет внутрішніх справ**

**Факультет № 4**

**Кафедра інформаційної безпеки**

**НАВЧАЛЬНО-МЕТОДИЧНІ МАТЕРІАЛИ  
ДО ПРАКТИЧНИХ ЗАНЯТЬ**

**з дисципліни «Основи алгоритмізації та програмування»**

**Галузь знань - 08 Право**

**Спеціальність - 081 Право**

**Спеціалізація – протидія кіберзлочинності**

**Ступінь вищої освіти: бакалавр**

**Форма навчання: денна**

м. Харків  
2016

## Передмова

Навчально-методичні вказівки до практичних занять з дисципліни «Основи алгоритмізації та програмування» для курсантів за спеціальністю - 081 Право на 40 арк.

### СХВАЛЕНО

Науково-методичною радою  
Харківського національного  
університету внутрішніх справ  
26.12.2016      Протокол № 10

### ЗАТВЕРДЖЕНО

Вченою радою факультету № 4  
15.11.2016      Протокол № 12  
\_\_\_\_\_ Марков В.В.

### ПОГОДЖЕНО

Секцією науково-методичної ради  
з технічних дисциплін  
Харківського національного  
університету внутрішніх справ  
12.12.2016      Протокол № 11  
\_\_\_\_\_ Сезонова І.К.

### ЗАТВЕРДЖЕНО

На засіданні кафедри інформаційної  
безпеки  
14.11.2016      Протокол № 8  
\_\_\_\_\_ Сезонова І.К.

### Рецензенти:

Нефьодов Л.І., завідувач кафедри АКТ ХНАДУ, д.т.н., професор  
Гнусов Ю.В., завідувач кафедри кібербезпеки ХНУВС, к.т.н., доцент

Розробники: Ільге І.Г., - Харків: Харківський національний університет  
внутрішніх справ, 2016р.

## 1. Загальні методичні вказівки

Заняття з дисципліни «Основи алгоритмізації та програмування» проводяться у вигляді практичних занять з використанням електронно-обчислювальної техніки (ПК). На їх проведення відводиться 14 учбових годин. Основними завданнями практичних занять є отримання курсантами необхідних знань в області алгоритмізації та програмування. Критеріями оцінювання стану вивчення навчальної дисципліни є вміння використовувати методи програмування в дослідженні правопорушень.

Номер та назва змістового модулю, номер та найменування теми	Кількість годин відведених на вивчення навчальної дисципліни					Література, сторінки	Вид контролю	
	Всього	з них:						
		лекції	Семінарські заняття	Практичні заняття	Лабораторні заняття			Самостійна робота
Семестр № 2								
Тема № 1: Алгоритм як основне поняття програмування. Лексичні основи мов високого рівня.	8	2		4		2	Конспект лекцій Література [1]: с. 14-26, [2]: с. 48-78 [4]: с. 24-36 [6]: с. 31-42	Аудиторна контрольна робота
Тема № 2: Алгоритмічна мова C++. Основні типи даних.	6	2		2		2	Конспект лекцій Література [2]: с. 51-76, [8]: с. 82-100	
Тема № 3: Програмування обчислювальних процесів. Оператори управління програмою	8	2		2		4	Конспект лекцій Література [2]: с. 51-76, [8]: с. 82-100	Самостійна контрольна робота
Тема №4: Функції	6	2		2		2	Конспект лекцій Література [2]: с. 51-76, [8]: с. 82-100	
Тема 5. Масиви. Похідні типи даних	8	2		4		2	Конспект лекцій Література [2]: с. 51-76, [8]: с. 82-100	
Всього за семестр № 2	36	10		14		12		Залік.

### 3. Методичні вказівки до практичних занять

#### ПРАКТИЧНА РОБОТА №1 ОСНОВНІ ПОНЯТТЯ АЛГОРИТМІЗАЦІЇ

**Мета роботи:** Одержати практичні навички в розробці і дослідженні алгоритмів розв'язання задач.

**Час проведення:** 4 години. **Місце проведення:** комп'ютерний клас.

**Навчальні питання:**

1. Визначення та властивості алгоритму
2. Схема алгоритму
3. Елементи базових структур алгоритмів

**Література:**

1. Основи програмування мовою C++: Навчальний посібник/ Є.П. Путятін, В.П. Степанов, В.П. Пчелінов, Т.Г. Долженкова, О.О. Матат. – Х. ТОВ «Компанія СМІТ», 2005. – 320 с.
2. Деннис М. Язык программирования С. ; Пер. с англ. / Деннис М. Ритчи, Брайан У. Керниган. - М. : Издательский дом "Вильямс", 2009. - 304 с.
3. Кнут Д. Искусство программирования для ЭВМ / Д. Кнут - т.1-3, М. : Мир, 1976 - 1977.

**Хід проведення заняття:**

I. Вступ до заняття.

Розв'язання задач за допомогою комп'ютера складається з декількох етапів:

1. Постановка задачі.
2. Побудова математичної моделі.
3. Розробка алгоритму.
4. Складання програми.
5. Компіляція програми.
6. Компонування програми.
7. Налаштування програми
8. Експлуатація програми.

Етап розробки алгоритму є одним найважливіших і визначає успішність розв'язання задачі в цілому.

II. Порядок проведення основної частини заняття.

1. Визначення та властивості алгоритму

Алгоритм – це наперед заданий чіткий опис скінченої послідовності вказівок, виконання яких дозволяє одержати правильний розв'язок задачі.

Алгоритм повинен відповідати певним вимогам і мати такі властивості:

- детермінованість;
- масовість;
- дискретність;
- результативність;
- формальність;


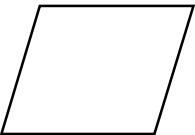

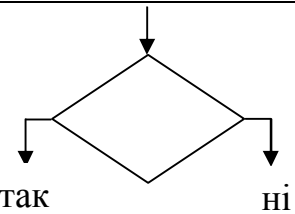
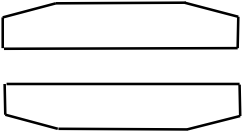

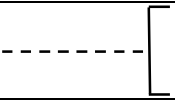

– скінченність.

Запис алгоритмів можливо здійснювати природною мовою, у вигляді графічних схем і мовою програмування.

## 2. Схема алгоритму

Схема алгоритму – зображення алгоритму у вигляді графічних схем, де кожному типу дій відповідає геометрична фігура – символ дії (ДОСТ 19.002 – 90, 19.003 – 90). Символи з'єднуються лініями, які вказують на послідовність дій.

Таблиця 1.1 – Опис основних символів схем алгоритмів

	Блок, що зображує початок чи кінець алгоритму.
	Блок введення-виведення даних.. Текст усередині блока конкретизує операцію, що виконується, і містить слово «введення» або «виведення», а також імена змінних, які необхідно ввести чи вивести.
	Арифметичний оперативний блок. Використовується для позначення дій, що задають або змінюють значення величин. Найчастіше всередині блока записують вираз використання математичних символів.
	Логічний оперативний блок. Використовується для вибору з двох можливих напрямків виконання алгоритму, в залежності від дотримання умов. Всередині блока записується умова вибору, перехід по стрілці з позначкою «так» відбувається, коли умова виконується, а перехід по стрілці з позначкою «ні» – у протилежному випадку.
	Блоки початку та кінця циклу. Всередині блока початку циклу записують діапазон та крок зміни параметра циклу, а у блоці кінця циклу – вказується ім'я параметру циклу.
	Блок виклику підпрограми (функції. Модуля), тобто допоміжних алгоритмів, які визначені автономно. Всередині блока записується, наприклад, ім'я функції, яка викликається, та список фактичних параметрів.
	Символ, що використовується для зв'язку елемента схеми з коментарем.
	Символ-з'єднувач застосовується для обриву лінії схеми та продовження її у другому місці.

Уся різноманітність задач і алгоритмів, їх розв'язання можуть бути представлені у вигляді сукупності базових структур.

До основних базових структур алгоритмів відносяться: лінійні, розгалужені; циклічні; складні.

Лінійною базовою структурою називається така структура, в якій команди виконуються послідовно одна за одною.

Розгалуженою базовою структурою називається така структура, що містить хоча б одну умову, в результаті перевірки якої здійснюється перехід до одного з можливих кроків. Процес розгалуження організується за допомогою логічного операторного блока.

Циклічна базова структура містить повторення кілька разів з новими вихідними даними певної послідовності команд, що утворює тіло циклу. У циклі повинна існувати змінна (параметр циклу), яка при кожному наступному виконанні тіла циклу змінює своє значення і визначає кількість повторень.

У складних структурах застосовується різноманітні обчислювальні процеси.

### 3. Елементи базових структур алгоритмів

Базові структури алгоритмів можуть бути представлені за допомогою незначної кількості структурних елементів. Їх використання є основою структурного підходу до розробки алгоритму.

Структурний підхід до розробки алгоритмів і програм припускає використання тільки декількох основних структурних блоків, комбінація яких надає всю різноманітність алгоритмів і програм.

Розглянемо основні структурні блоки.

Проходження – це послідовне розміщення блоків або груп блоків (послідовне виконання операторів).

Розгалуження – проходження по одній з двох альтернативних гілок алгоритму в залежності від деякої умови. Структурний блок розгалуження вміщує в собі перевірку умови та дві гілки.

Цикл – це багатократне виконання послідовності дій в залежності від деякої умови. Для організації повторювання дій використовується змінна, яка носить назву параметр циклу. В умові порівнюється поточне значення параметру циклу з умовою виходу з циклу. При кожному повторюванні циклу значення параметру циклу змінюється. Це забезпечує вихід із циклу. Послідовність дій, яка повторюється декілька разів, називається тілом циклу.

Структура цикл містить в собі перевірку умови виходу з циклу та тіло циклу. Умова може перевірятися до чи після виконання тіла циклу. В залежності від цього розподіляють цикли з передумовою та цикли з постумовою.

Умова повторювання дій у тілі циклу вказується всередині символу “рішення” і перевіряється при кожному входженні в цикл. Цикл виконується доти, доки додержується умова. У протилежному випадку (умова не додержується) – вихід із циклу. Початкові значення параметру циклу (наприклад,  $i=1$ ) присвоюються до першого входження в цикл. Поточне значення параметру циклу ( $i=i+1$ ) обчислюється безпосередньо у тілі циклу.

Ця конструкція використовується для організації циклу, коли кількість повторювань тіла циклу заздалегідь невідома.

Приклад 1.1 Обчислити значення функції  $y = ax^2 - \cos x$ , якщо  $x \in [-2; 1]$ ;  $h_x = 0.5$ ;  $a = 6.5$

У цьому прикладі проста змінна  $x$  є аргументом функції і змінюється від -2 до 1 с кроком 0,5.

Схема алгоритму розв’язання приведеної задачі включає:

1. Блок введення значень коефіцієнтів  $a$  та кроку зміни аргументу  $h_x$ .
2. Блок присвоювання початкових значень змінній  $x$ .
3. Блок обчислення значення  $y$ .
4. Блок виведення значень  $x$  і  $y$ .
5. Блок збільшення значення  $x$  на  $h_x$ .
6. Блок перевірки нерівності  $x \leq 1$  і, якщо  $x$  перевищить значення 1, забезпечує

вихід з циклу.

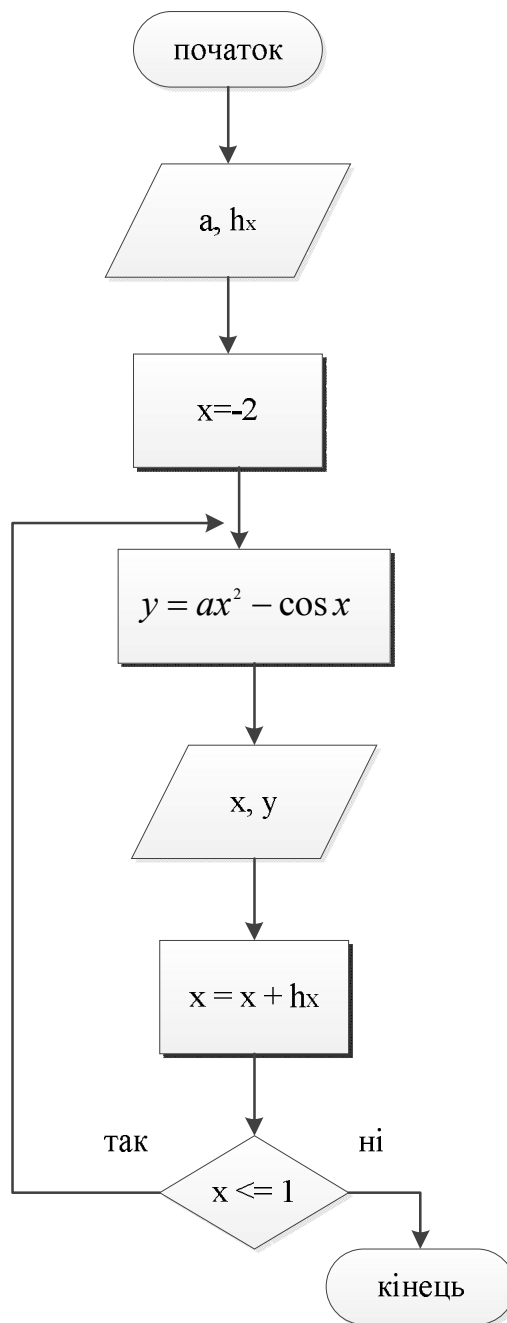


Рисунок 1.1 – Схема алгоритма обчислення значень функції

#### Контрольні запитання і завдання

1. Назвіть основні властивості алгоритму.
2. Що таке алгоритм?
3. Визначте основні етапи розробки алгоритмів.
4. Перелічить базові конструкції.
5. Перелічить складні базові конструкції.
6. Дайте визначення конструкції розгалуження.
7. Дайте визначення конструкції циклу.
8. Сформулюйте правило виконання циклу з передумовою.
9. Сформулюйте правило виконання циклу з відомою кількістю повторювань тіла циклу.

Завдання

Під час виконання роботи необхідно розробити алгоритм для розв'язання задачі, номер якої відповідає порядковому номеру студента в журналі академгрупи.

Завдання 1. розробити лінійний алгоритм для розв'язання задачі

**Варіант 1.**  $h_a$ ,  $h_b$ ,  $h_c$  – висоти трикутника за формулою  $h = \frac{2}{a} \sqrt{p(p-a)(p-b)(p-c)}$ , де  $p = (a+b+c)/2$ . Значення сторін трикутника вибрати самостійно.

**Варіант 2.**  $a = \frac{2 \cos(x - \frac{\pi}{6})b}{\frac{1}{2} + \sin^2 y}$ ;  $b = 1 + \frac{z^2}{3 + z^2/5}$ , де  $x=1,45$ ;

$y=-1,22$ ;  $z=3,5$ .

**Варіант 3.**  $s = 1 + x + \frac{x^2}{2} + \frac{x^3}{3} + \frac{x^4}{4}$ ;  $z = (\sin x^3 + \cos^2 y)$ , де  $x=1,2$ ,  $y=-0,8$ .

**Варіант 4.** Площу поверхні зрізаного конуса і його об'єм за формулами:  $S = \pi(R+r)l + \pi R^2 + \pi r^2$ ;  $l = \sqrt{h^2 + (R-r)^2}$ ;

$V = \frac{\pi}{3}(R^2 + r^2 + Rr)h$ . Значення  $R$ ,  $r$ ,  $h$  вибрати самостійно.

**Варіант 5.**  $y = e^{-bt} \sin(at+b) - \sqrt{|bt+a|}$ ;  $s = b \sin(at^2 \cos 2t) - 1$ , де  $a = -0,5$ ;  $b = 1,7$ ;  $t = 0,44$ .

**Варіант 6.**  $P = \sqrt{x^2 + b} - b^2 \sin^3(x+a)/xy$ , де  $b=15,5$ ;  $x=-2,9$ ;  
 $y = \cos^2 x^3 - \frac{x}{\sqrt{a^2 + b^2}}$ ;  $a=1,5$ .

**Варіант 7.**  $S = \frac{x^3 t g^2(x+b)^2 + a}{\sqrt{x+b}} c$ ;  $c = \frac{bx^2 - c}{e^{ax} - 1}$ , де  $a=16,5$ ;  $b=3,4$ ;  $x=0,61$ .

**Варіант 8.**  $\gamma = |x^{y/x} - \sqrt[3]{y/x}|$ ;  $\varphi = (x-y) \frac{y-z/(y-x)}{1+(y-x)^2}$ ; де  $x=1,82$ ,  $y=18,5$ ;  $z=-$

3,4.

**Варіант 9.**  $R = x^2(x+1)/b - \sin^2(x-a)$ ;  $S = \sqrt{xb/a} + |\cos(x+b)^3|$ , де  $a=0,7$ ;  $b=0,05$ ;  $x=0,43$ ;

**Варіант 10.**  $a = \frac{\sqrt{|x-1|} - \sqrt{y}}{1 + \frac{x^2}{2} + \frac{y^2}{4}}$ ;  $b = ctge^{(x+3)}$ . Значення  $x$ ,  $y$ ,  $z$  вибрати

самостійно.

**Варіант 11.**  $p = \frac{a}{b}$ , де  $a = (1 - y) \frac{(x + y)^2 / (x + 4)^3}{e^{-(x-2)} + (x^3 + 4)}$ ;

$b = \frac{1 + \cos(y - 2)}{x^{4/2} + \sin^2(y - 2)}$ ;  $x=1,25$ ;  $y=0,93$ .

**Варіант 12.**  $P = \frac{\sin^3(ax^3 + by^2 - ab)}{\sqrt[3]{(ax^3 + by^2 - a)^2 + \pi}} + \operatorname{tg}(ax^3 + by^2 - ab)$ , де  $x=0,25$ ;

$y=1,31$ ;  $a=3,5$ ;  $b=0,9$ .

**Варіант 13.**  $R = x^3(x + 1) / b^2 - \sin^2(x(x + a))$ ;

$S = \sqrt{xb/a} + \cos^2(x + b)^3$ , де  $a=0,7$ ;  $b=0,05$ ;  $x=0,5$ .

**Варіант 14.**  $y = \sin^3(a^2) - \sqrt[3]{x/b}$ ;  $z = x^2 / a + \cos^2(x + b)^3$ , де  $a=1,1$ ;  $b=0,004$ ;  $x=0,2$ .

**Варіант 15.**  $\varphi = \sqrt{mtgx + |c \sin x|}$ ;  $\xi = m \cos(bx + \sin x) + c^3$ , де  $m=2$ ;  $c=1$ ;  $x=1,2$ ;  $b=0,7$ .

**Варіант 16.**  $P = \frac{1 + \sin^2(x + 1)}{2 + |x - 2x^3 / (1 + x^2 y^3)|} + x^4$ ;  $Q = \cos^2(\operatorname{arctg} \frac{1}{z})$ , де

$x=0,25$ ;  $y=0,79$ ;  $z=0,81$ .

**Варіант 17.**  $Y = b^3 \operatorname{tg}^2 x - \frac{a}{\sin^2(x/a)}$ ;  $Z = ae^{-\sqrt{3}} \cos(bx/a)$ , де  $a=3,2$ ;

$b=17,5$ ;  $x=-4,8$ .

**Варіант 18.**  $K = \ln(a + x^3) + \sin^2(x/b)$ ;  $M = e^{-cx} \frac{x + \sqrt[3]{x+a}}{x - \sqrt{|x-b|}}$ , де  $a=10,2$ ;

$b=9,3$ ;  $x=2,4$ ;  $c=0,5$ .

**Варіант 19.**  $Y = \frac{a^{2x} + b^{-x} \cos(a + b)x}{|x + 1|}$ ;  $R = \sqrt{x^2 + b} - b^2 \sin^3(x + a) / x$ ,

де  $a=0,3$ ;  $b=0,9$ ;  $x=0,53$ .

**Варіант 20.**  $\alpha = \frac{a^x + b^{-x} \sin(a - b)}{\sqrt{|a - b|}}$ ;  $\beta = a \cdot e^{-\sqrt{a}} \cos(bx/a)$ ;  $a=0,5$ ;  $b=2,9$ ;

$x=0,3$ .

**Варіант 21.**  $\alpha = \sqrt{|ax^2 \sin 2x + e^{-2x}(x + b)|}$ ;

$\omega = 1 / \cos^2 x^3 - x / \sqrt[3]{a^2 + b^2}$ , де  $a=0,5$ ;  $b=3,1$ ;  $x=1,4$ .

**Варіант 22.**  $U = \frac{a^3 + e^{-x} \cos bx}{bx - e^{-x} \sin bx + 1}$ ;  $F = e^{2x} \ln(a + x) - b^{3x} \ln|x - b|$ , де

$a=0,5$ ;  $b=2,9$ ;  $x=0,3$ .

$$\text{Варіант 23. } Z = \frac{\sin x}{\sqrt{1 + m^3 \sin^2 x}} - mx \ln mx; \quad S = e^{-ax} \sqrt{x+1} + e^{bx} \sqrt{|x-3|},$$

де  $m=0,7$ ;  $x=1,77$ ;  $a=0,5$ ;  $b=1,08$ .

$$\text{Варіант 24. } P = \sqrt[3]{m \cdot \operatorname{tg} t + |c \cdot \sin t|}; \quad K = \cos^2 t - m / \sqrt[3]{|m^2 - t^2|}, \quad \text{де } m=2,1;$$

$t=1,02$ ;  $c=-1$ .

$$\text{Варіант 25. } a = y + \frac{x}{y^3 + \left| \frac{x^2}{y + \sqrt[3]{x^2}} \right|}; \quad b = 1 + \operatorname{tg}^2(x/2), \quad \text{де } x=1,23; y=0,79.$$

Завдання 2. Розробити алгоритм, що розгалужується для розв'язання задачі номер якої відповідає порядковому номеру студента в журналі академгрупи

$$\text{Варіант 1. } y = \begin{cases} ax + b \cos x, & x < 0,5, \\ bx^2 + c \sin 2x, & 0,5 \leq x < 1; \end{cases}$$

де  $x \in [0,2]$ ;  $h_x = 0,1$ ;  $a = 0,75$ ;  $b = 1,19$ ;  $c = -2,5$ .

$$\text{Варіант 2. } z = \begin{cases} a^x + b^y, & x + y \leq 1, \\ ax^2 + \ln(bxy), & x + y > 1; \end{cases}$$

де  $x \in [1,2]$ ;  $h_x = 0,1$ ;  $a = 1$ ;  $b = 2$ ;  $y \in [1,2]$ ;  $h_y = 0,2$ .

$$\text{Варіант 3. } y = \begin{cases} \sin|ax + b|, & |x| < a/b, \\ \cos|ax - b|, & |x| \geq a/b; \end{cases}$$

де  $a = 5$ ;  $y = 3$ ;  $x \in [1,2]$ ;  $h_x = 0,1$ .

$$\text{Варіант 4. } y = \frac{4ax^2 + 37x + b}{a - 0,5};$$

де  $h_a = 0,5$ ;  $h_x = 0,2$ ;  $a \in [1,2]$ ;  $x \in [0,1]$ ;  $b = 1,2$ .

$$\text{Варіант 5. } y = \begin{cases} \frac{\sin(x + 0,4)^2}{4x^2}, & |x| \leq 3, \\ x^4 + 2x^2 \cos x, & |x| > 3; \end{cases}$$

де  $x \in [-5,5]$ ;  $h_x = 1$ .

$$\text{Варіант 6. } y = \begin{cases} \frac{x}{15x^2}, & x^2 + y^2 \leq a^2, \\ x^2 + e^x, & x^2 + y^2 > a^2; \end{cases}$$

де  $x \in [-3,3]$ ;  $h_x = 0,5$ ;  $a = 5,3$ ;  $y = 1,1$ .

$$\text{Варіант 7. } y = e^{x/2} b^x (\cos z + \sin z); \quad z = \frac{\sqrt{4b + x^2}}{2};$$

де  $x \in [1,2]$ ;  $h_x = 0,2$ ;  $b = 2;4;6;8$ .

**Варіант 8.**  $y = z \frac{\sin(2+x)^2}{2+x}$ ;  $z = |x| + x^3$ ;

де  $x \in [1,4]$ ;  $h_x = 0,5$ .

**Варіант 9.**  $y = \begin{cases} \sin|ax + b^a|, & x < b, \\ \cos|ax - b^a|, & x \geq b; \end{cases}$

де  $x \in [0,5;3,1]$ ;  $h_x = 0,21$ ;  $a \in [-0,1;-1,4]$ ;  $h_a = 0,1$ ;  $b = 1,5$ .

**Варіант 10.**  $y = \begin{cases} a^{b-x} + c, & x < 2, \\ b^{c-x} + a, & 2 \leq x \leq 4, \\ c^{a-x} + b, & x > 4; \end{cases}$

де  $x \in [0,5]$ ;  $h_x = 0,31$ ;  $a = 2$ ;  $b = 3$ ;  $c = 7$ .

**Варіант 11.**  $y = \begin{cases} \sin|ax + b^a|, & x < b, \\ \cos|ax - b^a|, & x \geq b; \end{cases}$

де  $x \in [1,3]$ ;  $h_x = 0,1$ ;  $a \in [1,5]$ ;  $b = 2$ .

**Варіант 12.**  $z = \begin{cases} \frac{x^2}{(x-5)^3}, & x > y, \\ \frac{(x-2)^3}{y(x-5)^4}, & x \leq y; \end{cases}$  де  $x \in [1;10]$ ;  $h_x = 2$ ;  $y \in [-4,3]$ ;

$h_y = 1$ .

**Варіант 13.**  $z = \begin{cases} \ln \sqrt{ax^2 + by^3}, & xy > 0, \\ a^x + b^y, & xy \leq 0; \end{cases}$  де  $x \in [-1,1]$ ;  $h_x = 0,9$ ;

$a = 1,7$ ;  $b = 2,3$ ;  $y \in [-2,2]$ ;  $h_y = 1$ .

**Варіант 14.**  $z = \frac{a^x \ln|x+2^3|}{x+1}$ ; де  $x \in [-3,3]$ ;  $h_x = 0,5$ ;  $a = 2,4$ .

**Варіант 15.**  $z = \frac{\sqrt{x-1,5} + x^a}{(x-2)^{1/3}}$ ;  $x \in [1,4]$ ;  $h_x = 0,5$ ;  $a > -0,5$ ;  $h_a = 0,2$ , де  $x$

і  $a$  змінюються одночасно.

**Варіант 16.** Обчислити функцію  $y = \begin{cases} \cos(ax+2), & x > 2, \\ \operatorname{tg}|x-2a|, & x \leq 2; \end{cases}$  де  $x \in [0,5;3,1]$ ;

$h_x = 0,2$ ,  $a$  – має початкове значення  $0,1$  і змінюється одночасно зі змінною  $x$  з кроком  $h_a = 0,3$ .

**Варіант 17.** Обчислити функцію двох змінних  $y = \begin{cases} a^2 + x, & a > x, \\ a^2, & a = x, \\ a^2 - x, & a < x; \end{cases}$  де

$$x \in [1,8]; h_x = 0,5; a \in [2;10]; h_a = 1.$$

**Варіант 18.** Обчислити функцію  $y = z \frac{\sin(2+x)^2}{2+x}$ ; де  $z = |x| + x^2$ ;

$$x \in [-2,5;2]; h_x = 0,5.$$

**Варіант 19.** Підрахувати, скільки разів функція  $y = \cos x^2 \cdot e^{-x}$  приймає негативне значення, якщо  $x \in [0,3;5]; h_x = 0,1$ .

**Варіант 20.** Увести координати десяти точок на площині і визначити, в якій чверті більше всього точок.

**Варіант 21.** Увести п'ять наборів сторін трикутника  $a, b, c$  і визначити, для яких сторін висота  $h_a$  буде найбільшою. Підказка:  $h_a = \frac{2}{a} \sqrt{p(p-a)(p-c)}$ , де  $p = (a+b+c)/2$ .

**Варіант 22.** Ввести координати п'яти точок і визначити, яка з них потрапить в окружність з радіусом  $R$  і координатами центра  $(a, b)$ . Підказка: рівняння окружності має вигляд:  $(x-a)^2 + (y-b)^2 = R^2$ .

**Варіант 23.**  $z = \frac{\cos(x)-1}{x+1}$ , де  $x \in [-3,3]; h_x = 0,5; a = 2,4$ .

**Варіант 24.** Підрахувати, скільки разів функція  $y = \cos x^2 \cdot e^{-x}$  приймає негативне значення, якщо  $x \in [0,3;5]; h_x = 0,1$ .

**Варіант 25.** Обчислити функцію у вигляді формули  $y = \sum_{i=1}^{10} \frac{x^i}{i}$  для кожного  $i$ .

III Заключна частини заняття: відповіді на контрольні запитання і виконання контрольних завдань.

## ПРАКТИЧНА РОБОТА №2

### РОЗРОБКА ТА ВИКОНАННЯ ПРОСТИХ ПРОГРАМ У СЕРЕДОВИЩІ VISUAL C++

**Мета роботи:** Освоїти прийоми роботи з текстовим редактором і основними компонентами інтегрованого середовища програмування C++; придбати практичні навички введення - виведення даних, складання і налагодження найпростіших програм мовою C++; освоїти правила опису типів даних.

**Час проведення:** 2 годин. **Місце проведення:** комп'ютерний клас.

#### Навчальні питання:

1. Розробка консольного додатка у середовищі VISUAL C++
2. Структура програми і типи даних

3. Введення – виведення даних
4. Використання математичних функцій

Література:

4. Бюдбельский В.В. Язык С++. 5-е издание. – М: Финансы и статистика, 2004. – 560 с.
5. Основи програмування мовою С++: Навчальний посібник/ Є.П. Путятін, В.П. Степанов, В.П. Пчелінов, Т.Г. Долженкова, О.О. Матат. – Х. ТОВ «Компанія СМІТ», 2005. – 320 с. (с.25-56).
6. Павловская Т. А. С/С++. Программирование на языке высокого уровня. / Т. А. Павловская - СПб. : Питер, 2006. - 461 с. : ил.

### Хід проведення заняття:

I. Вступ до заняття.

VISUAL C++ – це інструмент професійного програмування в середовищі Windows. Основна сила пакета – використання бібліотеки класів. Існують спеціальні програми: майстри пишуть за користувача значну частину проекту – програми. Інтегроване середовище VISUAL C++ дає можливість писати та виконувати складні програми. Програма в цьому середовищі – це проект, що складається з набору файлів, розміщених в робочих областях. У середовищі можна також реалізовувати звичайні (без використання стандартних класів) програми мовою С++. Для цього застосовується консольний режим.

II. Порядок проведення основної частини заняття.

1. Розробка консольного додатка у середовищі VISUAL C++

Для реалізації програми необхідно виконати таку послідовність дій:

1. Запустити за допомогою меню VISUAL C++ .
  2. Використати команду *File→New*, з'являється вікно *New*.
  3. Перейти на вкладку *Projects* і вибрати зі списку рядок *Win32 Console Application*.
  4. В текстовому полі *Project* ввести ім'я створюемого проекту **Primer**, а в полі *Location* вказати каталог для проекту за допомогою ієрархічного списку в правому кутку.
  5. Натиснути кнопку *Ok* у вікні *New*. Проект **Primer** створено як файли **Primer.dsw** і **Primer.dsp**.
  6. Якщо необхідно продовжити роботу над програмою, з якою працювали раніше, слід використати команду *File→Open Workspaces* і відкрити файл з розширенням **\*.dsw**.
  7. Для створення файла **Primer.cpp** слід знову використати команду *File→New*, але з вкладкою *Files*.
  8. Вибрати зі списку рядок *C++ Source File*, ввести ім'я файла **Primer.cpp** у полі *Filename* та слідкувати, щоб прапорець *Add to project* було виставлено! Натиснути *Ok*. В результаті файл **Primer.cpp** буде створено і відкрито у середовищі VISUAL C++.
  9. Якщо необхідний файл **\*.cpp** знаходиться на дискеті, слід спочатку створити проект і в ньому всередині замінити його файл з розширенням **\*.cpp** на файл **\*.cpp**, що знаходиться на дискеті. Для цього найкраще використовувати програму роботи з файлами *Far* .
  10. Набрати текст програми на C++ та відкомпілювати його командою *Build→Build Primer.exe* або натисненням іконічної кнопки чи клавіші *F7*.
  11. Запустити програму командою *Build→Execute Primer.exe* або клавішами *Ctrl+W*.
  12. Для пошуку та виправлення помилок компіляції у нижньому вікні слід знайти помилковий оператор та два рази натиснути на ньому лівою клавішею мишки. Виходимо на текст програми , виправляємо помилку та знову виконуємо компіляцію. При цьому треба не забувати зберігати на дискові новий варіант програми.
- Відладчик вмикається та вимикається в полі меню *Build→start Debug→Configuration*.

## 2. Структура програми і типи даних

Як відомо, кожна програма є послідовністю інструкцій в машинних кодах, що керують поведінкою певного обчислювального засобу. Кожна програма мовою C++ включає до себе підключення заголовних файлів за допомогою директиви препроцесора **#include**<ім'я заголовного файлу.h>, головну функцію **main()** і, якщо це необхідно, декілька інших функцій, створених користувачем. Загальна структура простішої програми мовою C++ має вигляд:

```
#include<ім'я заголовного файлу.h>
void main ()
{тіло головної функції;}
```

Специфікатор типу *void* головної функції показує, що вона не повертає ніякого результату. Фігурні скобки означають початок і кінець блока обчислень і можуть бути скільки потрібно вкладеними.

Коментарі. У процесі створення програми використовуються коментарі, текст яких завжди ігнорується компілятором, але дозволяє програмісту описувати призначення тої чи іншої частини програми. В мові C++ використовується два різновиди коментарів:

```
/* текст */      – багатострочний коментар, який не може бути вкладеним і
// текст         – однострочний коментар до кінця строки.
```

Змінні та типи даних. Для того, щоб програма могла на протязі свого виконання зберігати визначені дані, використовуються змінні та константи. Одним із базових понять властивостей програми є ідентифікатор – ім'я змінної, ім'я функції або позначка. У програмі ідентифікатор може містити прописні та строчні латинські букви, цифри та символ підкреслення та обов'язково починатися з букви або символу підкреслення. Ідентифікатор немає співпадати з ключовими словами з урахуванням регістра. Ключові слова – це зарезервовані мовою ідентифікатори, які мають спеціальне призначення.

Для використання змінної в програмі вона має бути оголошена. Створення ідентифікаторів і здійснюється при такому оголошенні.

Змінна – це об'єкт програми, що займає в загальному випадку декілька комірок пам'яті, та призначений зберігати дані. При оголошенні змінної для неї резервується якась область пам'яті, розмір якої залежить від конкретного типу змінної. Будь-яка константа, змінна, значення чи функції вираження в C++ характеризуються своїм типом. Тип цих об'єктів визначає множину припустимих значень, форму внутрішнього уявлення, а так само множину припустимих операцій. Тому всі дані, що використовуються в програмі, до їхньої появи обов'язково мають бути оголошені (описані).

Нижче наведено перелік базових типів змінних і їхній розмір у байтах (табл. 1.1).

Наприклад,

```
bool dd=true;      – логічна змінна dd=1;
bool dd1=1;        – логічна змінна dd1=true;
int a=1,b=0;        – цілі змінні a=1, b=0;
char sim='A';       – символічна змінна sim='A';
float Age=18.5;     – десяткова змінна Age=18.5 з плаваючою точкою;
void MyFunction();  – функція не повертає ніякого значення тощо.
```

Константи відрізняються від змінних тим, що значення, яке присвоєне константі первісне, не може бути змінено на протязі, виконання всієї програми. Оголошення констант виконується за такою схемою:

```
const [тип] ідентифікатор = значення;
```

Наприклад:

**const double** Pi=3.1415; // константа Pi приймає значення 3.1415;

**const** a=1; // за замовчанням тип **int**.

Таблиця 1.1 – Базові типи даних для ПК на базі платформи Intel

Тип	Розмір, байт	Значення
<b>Bool</b>	1	true (1) або false (0)
<b>unsigned short int</b>	2	від 0 до 65535
<b>Short int</b>	2	від -32768 до 32767
<b>unsigned long int</b>	4	від 0 до 4 294 967 295
<b>long int</b>	4	від – 2 147 483 648 до 2 147 483 367
<b>int</b> (16 розрядів)	2	від -32768 до 32767
<b>int</b> (32 розрядів)	4	від – 2 147 483 648 до 2 147 483 367
<b>unsigned int</b> (16розрядів)	2	від 0 до 65535
<b>unsigned int</b> (32 розрядів)	4	від 0 до 4 294 967 295
<b>Char</b>	1	від 0 до 256
<b>Float</b>	4	від 1.2e-38 до 3.4e38
<b>double</b>	8	від 2.2e-308 до 1.8e308
<b>long double</b>	10	від 3.4e-4932 до 3.4e4932
<b>Void</b>	2 або 4	

### 3. Введення – виведення даних

Під час введення – виведення потоку всі дані розглядаються як потік окремих байтів. Функції введення – виведення для потоку дозволяють обробляти дані різних розмірів і форматів: від одиночного символу до великих структур даних. Програміст може використовувати функції чи бібліотеки, розробляти власні і включати їх у бібліотеку. Для доступу до бібліотеки цих класів треба включити в програму відповідні файли заголовків, наприклад:

```
#include <iostream.h>.
```

За умовчанням стандартні введення, виведення і виведення повідомлень щодо помилки відносяться до консолі користувача (клавіатура й екран). Це означає, що кожен раз, коли програма очікує введення зі стандартного потоку, дані мають надходити з клавіатури, а якщо програма виводить, то на екран.

Найчастіше в C++ використовується поточне введення даних, операції якого включені до складу класів **istream** чи **iostream**. Воно може здійснюватися з визначеним у цих класах вхідним потоком **сін** чи вхідним потоком, що визначений користувачем. Для читання символів з цього потоку вказується операція витягу з потоку, що позначається за допомогою символів **">>"**. Це перевантажена операція, визначена для всіх простих типів і покажчика на **char**.

Формат запису оператора **сін** має вид:

```
сін [>>values];
```

Наприклад, для введення значень змінних **x** і **y** можна написати:

```
сін>>x>>y;
```

Виведення даних може бути неформатованим і форматованим. Найчастіше для виведення застосовується визначена операція <<, що записується разом з ім'ям вихідного потоку cout. Таким чином, запис:

```
cout<<x;
```

означає виведення значення змінної x (чи запис у потік). Цей оператор вибирає необхідну функцію перетворення даних у потік байтів.

Формат запису **cout** відрізняється від форми запису команд C++ :

```
cout<< data [<< data<<"\n"];
```

де data – це змінні, константи чи вирази, "\n" – керуючий символ переведення каретки.

Для додаткового керування вихідними даними використовуються маніпулятори: **setw(n)** і **setprecision(k)**. Маніпулятор **setw(n)** служить для вказівки довжини поля, що приділяється для виведення наступних даних (тут n – кількість позицій у рядку), **setprecision(k)** призначений для вказівки кількості позицій у дробовій частині дійсних чисел. Для використання цих маніпуляторів треба підключати заголовний файл **<iomanip.h>**.

Для форматованого введення даних використовується функція **scanf** заголовного файлу **<stdio.h>**. Функція **scanf** змінне число параметрів, але як фактичні параметри вона використовує адреси змінних, а не їхні значення. При цьому перед відповідним параметром ставиться знак **&** – символ взяття адреси змінної. Наприклад, **&x** означає "адреса змінної x", а не значення, що ця змінна має в даний момент. Рядок форматів функції **scanf** указує, які дані очікуються на вході. Якщо функція зустрічає у форматному рядку знак %, за яким впливає знак перетворення, то вона буде пропускати на вході символи, поки не зустріне який-небудь непорожній символ. Таким чином функція **scanf** змінює значення змінних у викликаючій програмі.

Форма запису цієї функції має вигляд:

```
scanf ("рядок форматних кодів", список імен змінних );
```

Для форматованого виведення даних використовується функція з того ж заголовного файлу **<stdio.h>**. Для звертання до функції використовуються параметри, які розташовують у круглі дужки. Найчастіше функція **printf** використовується для виведення значень змінних. Першим аргументом у звертанні до функції ставиться рядок форматів (розміщується в подвійні лапки), а наступними, якщо вони є, виведені об'єкти.

Рядок форматів може включати звичайні символи, що копіюються при виведенні і специфікації перетворення, що починаються зі знака %: за специфікаціями впливає символ перетворення. Кожна специфікація перетворення відповідає одному з аргументів, що впливають за форматним рядком, і між ними встановлюється взаємно однозначна відповідність, наприклад:

```
printf ("Значення a, b, c дорівнюють: %d %d %d \n", a, b, c);
```

Тут буква **d** у специфікації перетворення вказує, що значення аргументу має бути надруковане як десяткове ціле число.

Під час виведення використовуються ті ж специфікації, що і під час введення:

%c – для виведення окремого символу;

%s – для печатки символьного рядку;

%x – для виведення шістнадцяткової літери ;

%0 – для виведення вісімкових чисел;

%f – для виведення чисел з плаваючою точкою.

#### 4. Використання математичних функцій

Використання математичних функцій у мові C++ потребує підключення заголовкового файлу **#include<math.h>** (табл. 1.2).

Таблиця 1.2– Математичні функції заголовкового файлу <math.h>

Прототип функції	Ім'я	Зміст
double acos (double x);	acos(x)	Арккосинус
double asin (double x);	asin (x)	Арксинус
double atan (double x);	atan (x)	Арктангенс
double atan2 (double y, double x);	atan2 (y, x)	Арктангенс від y/x
double ceil (double x);	ceil (x)	Округлення в більший бік
double cos (double x);	cos (x)	Косинус x в радіанах
double cosh (double x);	cosh (x)	Косинус, гіперболічний
double exp (double x);	exp (x)	$e^x$ , e в ступені x
double fabs (double x);	fabs (x)	Абсолютне значення $ x $ типу double
Double floor (double x);	floor (x)	Повертає найближче ціле, не більше x
Double fmod(double x, double y);	fmod(x,y)	Залишок від ділення x на y
Double log (double x);	log (x)	Натуральний логарифм
Double log10 (double x);	log10 (x)	Десятковий логарифм
Double pow (double x, double y);	pow (x,y)	$x^y$ , x в ступені y
Double sin (double x);	sin (x)	Синус x, в радіанах
Double sinh (double x);	sinh (x)	Синус x, гіперболічний
Double sqrt (double x);	sqrt (x)	Корінь з x, $x > 0$
Double tan (double x);	tan (x)	Тангенс x, в радіанах
double tanh (double x);	tanh (x)	Тангенс x, гіперболічний
double abs (int x);	abs (x)	Модуль x типу int
double atof (const char *s);	atof (s)	Перетворює рядок символів у число з плаваючою комою
double hypot (double x, double y);	hypot (x,y)	Корінь з $(x^2+y^2)$
long labs (long x);	labs (x)	Абсолютна величина типу long $ x $
double pow10 (int p);	pow10 (p)	Повертає $10^p$

**Приклад.** Обчислити значення площі круга за значенням радіуса, який вводиться з клавіатури.

```
#include <iostream.h>
void main()
{
    const double Pi=3.1415;
    float Rad;
    double Square;
    cout<<"Input radius =";
    cin>>Rad;
    Square=Pi*Rad*Rad; // Square=Pi*pow(Rad,2);
    cout<<"Square="<<Square<<endl;
}
```

Контрольні запитання і завдання:

1. Розкажіть про структуру простішої програми мовою C++.
2. Які типи даних вам відомі?
3. Чим відрізняються змінні від констант?
4. Як можна ввести інформацію?
5. Як можна вивести інформацію?

Завдання:

Під час виконання роботи необхідно створити та відлагодити програму для розв'язання задачі, номер якої відповідає порядковому номеру студента в журналі академгрупи.

Для наведених нижче варіантів необхідно скласти програму обчислення значення функцій при різних значеннях аргументів.

**Варіант 1.** Завдання 1. Написати програму обчислення об'єму паралелепіпеда. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, які вводяться користувачем, виділені напівжирним шрифтом).

Обчислення; об'єму паралелепіпеда.

Введіть початкові дані:

Довжина (см) > **9**

Ширина(см)> **7.5**

Висота(см)> **5**

Об'єм: 337.50 куб.см.

Завдання 2. Написати програму перерахунку ваги з фунтів в кілограми (один фунт - це 405,9 грами).

**Варіант 2.** Завдання 1. Написати програму обчислення площі поверхні паралелепіпеда. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, які вводяться користувачем, виділені напівжирним шрифтом).

Обчислення площі поверхні паралелепіпеда.

Введіть початкові дані:

Довжина (см) > **9**

Ширина(см)> **7.5**

Висота(см)> **5**

Площа поверхні: 90.00 кв.см.

Завдання 2. Написати програму перерахунку відстані з кілометрів у версти (одна верста – це 1066,8 м).

**Варіант 3.** Завдання 1. Написати програму обчислення об'єму куба. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, які вводяться користувачем, виділені напівжирним шрифтом).

Обчислення об'єму куба.

Введіть довжину ребра (см) і натисніть клавішу <Enter>

> **9.5**

Об'єм куба: 857.38 куб.см.

Завдання 2. Написати програму обчислення площі трапеції  $S = \frac{a+b}{2}h$ , де  $a$  і  $b$  – довжини підстав;  $h$  – висота трапеції.

**Варіант 4.** Завдання 1. Написати програму обчислення об'єму циліндра. Нижче

приведений вид екрану під час виконання програми, що рекомендується (дані, які вводяться користувачем, виділені напівжирним шрифтом).

Обчислення об'єму циліндра.

Введіть початкові дані:

радіус підстави (см) > **5**

висоту циліндра (см) > **10**

Об'єм циліндра 1570.80 см. куб.

Для завершення натисніть клавішу <Enter>.

Завдання 2. Написати програму обчислення вартості покупки складається з декількох зошитів, обкладинок до них і олівців.

**Варіант 5.** Завдання 1. Написати програму обчислення вартості покупки, що складається з набору зошитів і олівців. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, які вводяться користувачем, виділені напівжирним шрифтом).

Обчислення вартості покупки.

Введіть початкові дані:

Ціна зошита (грн.) > **2.75**

Кількість зошита (грн.) > **5**

Ціна олівців (грн.) > **0.85**

Кількість олівців > **2**

Вартість покупки: 15.45 грн.

Завдання 2. Написати програму обчислення об'єму порожнистого циліндра  $V = \pi \cdot h(r_1^2 - r_2^2)$ , де  $r_1$  – радіус циліндра;  $r_2$  – радіус отвору;  $h$  – висота циліндра.

**Варіант 6.** Завдання 1. Написати програму обчислення вартості покупки складається з декількох зошитів і такої ж кількості обкладинок до них. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, які вводяться користувачем, виділені напівжирним шрифтом).

Обчислення вартості покупки.

Введіть початкові дані:

Ціна зошита (грн.) > **2.75**

Ціна обкладинок (грн.) > **0.5**

Кількість комплектів (грн.) > **7**

Вартість покупки: 15.45 грн.

Завдання 2. Написати програму обчислення об'єму і площі поверхні кулі, радіусом  $r$ , за

формулами  $V = \frac{4}{3} \cdot \pi \cdot r^3$ ;  $S = 4 \cdot \pi \cdot r^2$ .

**Варіант 7.** Завдання 1. Написати програму обчислення площі трикутника, якщо відома довжина підстави і висоти. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, які вводяться користувачем, виділені напівжирним шрифтом).

Обчислення площі трикутника.

Введіть початкові дані:

Підстава (см) > **8.5**

Висота(см)> **10**

Площа трикутника 42.50 кв.см.

Завдання 2. Напишіть програму, яка виконує ці розрахунки: Верстат –автомат виготовляє в 1 годину КОЛ шайб. Скільки шайб він виготовить за 8-ї годинну зміну, за

робочий день при двозмінній роботі і за 30-ти денний місяць, якщо в місяці 4 вихідних дня і в кожній зміні верстат знаходиться 1 година на профілактиці?

**Варіант 8.** Завдання 1. Написати програму обчислення площі трикутника, якщо відомі довжини двох його сторін і величина кута між цими сторонами. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, які вводяться користувачем, виділені напівжирним шрифтом).

Обчислення площі трикутника.

Введіть (через пропуск) довжини двох сторін (см) трикутника > 25 17

Введіть величину кута між сторонами трикутника > **30**

Площа трикутника: 106.25 кв.см.

Завдання 2. Напишіть програму визначення числа болтів в ящику з розмірами ( $H*B*S$ ) мЗ, якщо один болт в середньому займає об'єм  $2\text{ см}^3$ . Розміри ящика  $H$ ,  $B$  і  $S$  повинні вводитися з клавіатури.

**Варіант 9.** Завдання 1. Написати програму обчислення опору електричного ланцюга, що складається з двох паралельно сполучених опорів. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, які вводяться користувачем, виділені напівжирним шрифтом).

Обчислення опору електричного ланцюга

при паралельному з'єднанні елементів.

Введіть початкові дані:

Величина першого опору (Ом) > **15**

Величина другого опору (Ом) > **20**

Опір ланцюга: 8.57 Ом

Завдання 2. Написати програму обчислення об'єму конуса 
$$S = \frac{1}{3} \cdot \pi \cdot r^2 \cdot h.$$

**Варіант 10.** Завдання 1. Написати програму обчислення опору електричного ланцюга, що складається з двох послідовно з'єднаних опорів. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, які вводяться користувачем, виділені напівжирним шрифтом).

Обчислення опору електричного ланцюга при паралельному з'єднанні елементів.

Введіть початкові дані:

Величина першого опору (Ом) > **15**

Величина другого опору (Ом) > **27.3**

Опір ланцюга: 42.30 Ом

Завдання 2. Напишіть програму, яка виконує ці розрахунки: В російсько-англійському словнику 948 сторінок; на кожній сторінці в середньому 30 російських слів. Кожне слово в середньому містить 5 букв. Кожна буква займає завширшки 2 мм.

Скільки російських слів в словнику? Відрізок якої довгі займуть всі російські слова, якщо написати їх підряд?

**Варіант 11.** Завдання 1. Написати програму обчислення сили струму в електричній ланцюзі. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, які вводяться користувачем, виділені напівжирним шрифтом).

Обчислення сили струму в електричному ланцюзі.

Введіть початкові дані:

Напруга (вольт) > **36**

Опір (Ом) > **1500**  
Сила струму: 0.024 Ампер.

Завдання 2. Напишіть програму обчислення різниці творів ( $a*b$ ) і ( $c*d$ ) чотирьох довільних дійсних чисел  $a$ ,  $b$ ,  $c$  і  $d$ .

Введіть з клавіатури значення:  $a=2,1265123$ ;  $b=3,1116743$ ;  $c=1,5224222$ ;  $d=4,346372$ .

**Варіант 12.** Завдання 1. Написати програму обчислення вартості поїздки на автомобілі на дачу (туди і назад). Початковими даними є: відстань до дачі (км); кількість бензину, яка споживає автомобіль на 100 км пробігу; ціна одного літра бензину. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, які вводяться користувачем, виділені напівжирним шрифтом).

Обчислення вартості поїздки на дачу і назад.

Введіть початкові дані:

Відстань до дачі (км) > **67**

Витрата бензину (літрів на 100 км пробігу) > **8.5**

Ціна літра бензину (грн.) > **6.5**

Поїздка на дачу і назад обійдеться в 74.04 грн.

Завдання 2. Написати програму обчислення струму, по відомих значеннях напруги і опору електричного ланцюга.

**Варіант 13.** Завдання 1. Написати програму, що обчислює швидкість, з якою бігун пробіг дистанцію. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, які вводяться користувачем, виділені напівжирним шрифтом).

Обчислення швидкості бігу.

Введіть довжину дистанції (метрів) > **1000**

Ведіть час (хвилин.секунд) > **3.25**

Дистанція: 1000 м

Часи 3 мін 25 сік = 205 сік

Швидкості бігу = 17.56 км/година

Завдання 2. Написати програму обчислення опору електричної цінуї по відомих значеннях напруги сили струму.

**Варіант 14.** Завдання 1. Написати програму обчислення об'єму циліндра. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, які вводяться користувачем, виділені напівжирним шрифтом).

Обчислення об'єму циліндра.

Введіть початкові дані:

Радіус підстави (см) > **5.5**

Висот циліндра (см) > **7**

Об'єм 665.23 см.куб.

Завдання 2. Написати програму обчислення опору електричний ланцюгу складається з трьох послідовно сполучених резисторів.

**Варіант 15.** Завдання 1. Написати програму обчислення площі поверхні циліндра. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, які вводяться користувачем, виділені напівжирним шрифтом).

Обчислення площі поверхні циліндра.

Введіть початкові дані:

Радіус підстави (см) > **5.5**

Висота циліндра (см) > 7

Площа поверхні циліндра: 431.97 кв.см.

Завдання 2. Написати програму обчислення опору електричного ланцюга, що складається з двох паралель сполучених резисторів.

**Варіант 16.** Завдання 1. Написати програму обчислення об'єму паралелепіпеда. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, які вводяться користувачем, виділені напівжирним шрифтом).

Обчислення об'єму паралелепіпеда.

Введіть в одному рядку довжину, ширину і висоту паралелепіпед (в сантиметрах).

Числа розділяйте пропусками.

Після введення останнього числа натискуйте <Enter> **7.5 2.5 3**

Об'єм паралелепіпеда 56,25 см.куб.

Завдання 2. Написати програму обчислення вартості покупки складається з помідорів, огірків і декількох пучків кропу.

**Варіант 17.** Написати програму обчислення величини доходу по внеску. Процентна ставка (% річних) і час зберігання (днів) задаються під час роботи програми. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, введені користувачем, виділені напівжирним шрифтом).

Обчислення величини доходу по внеску.

Введіть початкові дані:

Величина внеску (грн.) > **2500**

**Термін внеску (днів) > 30**

Процентна ставка (річних в %) > **20**

Дохід: 41.10 грн.

Сума, після закінчення терміну внеску: 2541.10 грн.

Завдання 2. Написати програму обчислення об'єму порожнистого циліндра  $V = \pi \cdot h \cdot (r_1^2 - r_2^2)$ , де  $r_1$  – радіус циліндра,  $r_2$  – радіус отвору,  $h$  – висота циліндра.

**Варіант 18.** Завдання 1. Написати програму перерахунку величини тимчасового інтервалу, заданого в хвилинах, у величину, виражену в годинах і хвилинах. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, які вводяться користувачем, виділені напівжирним шрифтом).

Введіть часовий інтервал (в хвилинах) > **150**

150 хвилин - це 2 ч. 30 мін.

Завдання 2. Напишіть програму, яка обчислює площу трикутника, якщо відомі координати його кутів.

**Варіант 19.** Завдання 1. Написати програму, яка перетворить введене з клавіатури дробове число в грошовий формат. Наприклад, число 12,5 повинне бути перетворено до вигляду 12 грн. 50 коп.

Перетворення числа в грошовий формат.

Введіть дробове число > **23.6**

23.6 грн. - це 23 грн. 60 коп.

Завдання 2. Напишіть програму, яка виконує ці розрахунки: Верстат–автомат виготовляє за 1 годину КОЛ шайб. Скільки шайб він виготовить за 8-и годинну зміну, за

робочий день при двозмінній роботі і за 30-ти денний місяць, якщо в місяці 4 вихідних дня і в кожній зміні верстат знаходиться 1 годину на профілактиці?

**Варіант 20.** Завдання 1. Напишіть програму, яка обчислює площу трикутника, якщо відомі координати його кутів. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, введені користувачем, які вводяться напівжирним шрифтом).

Обчислення площі трикутника.

Введіть координати кутів

(числа розділяйте пропуском):

x1,y1 > **-2 5**

x2,y2 > **1 7**

x3,y3 > **5 -3**

Площа трикутника: 23.56 кв.см.

Завдання 2. Написати програму обчислення вартості покупки, що складається з набору зошитів і олівців.

**Варіант 21.** Завдання 1. Написати програму, що обчислює швидкість, з якою бігун пробіг дистанцію. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, які вводяться користувачем, виділені напівжирним шрифтом).

Обчислення швидкості бігу.

Введіть довжину дистанції (метрів) > **1000**

Ведіть час (хвилин.секунд) > **3.25**

Дистанція: 1000 м

Часи 3 мін 25 сік = 205 сік

Швидкості бігу = 17.56 км/година

Завдання 2. Написати програму обчислення об'єму  $V = \frac{3}{4} \cdot \pi \cdot r^3$  і площі поверхні кулі

$$S = 4 \cdot \pi \cdot r^2.$$

**Варіант 22.** Завдання 1. Написати програму обчислення опору електричного ланцюга, що складається з двох послідовно з'єднаних опорів. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, які вводяться користувачем, виділені напівжирним шрифтом).

Обчислення опору електричного ланцюга

при паралельному з'єднанні елементів.

Введіть початкові дані:

Величина першого опору (Ом) > **15**

Величина другого опору (Ом) > **27.3**

Опір ланцюга: 42.30 Ом

Завдання 2. Напишіть програму визначення суми Вашого внеску в банку «Альфа-Омега», якщо в кінці кожного року Ваш внесок збільшується не 3% від суми, що знаходиться на внеску (не від первинної суми, а від суми, що знаходиться на вкладі). Програма повинна визначити суму Вашого внеску після 2-х перерахунків. Первинний внесок – довільний і його значення повинне вводиться з клавіатури. Результат був виведений на екран монітора.

**Варіант 23.** Завдання 1. Написати програму обчислення вартості покупки складається з декількох зошитів і такої ж кількості обкладинок до них. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, які вводяться користувачем, виділені напівжирним шрифтом).

Обчислення вартості покупки.  
Введіть початкові дані:  
Ціна зошита (грн.) > **2.75**  
Ціна обкладинок (грн.) > **0.5**  
Кількість комплектів (грн.) > **7**  
Вартість покупки: 15.45 грн.

Завдання 2. Напишіть програму, яка виконує ці розрахунки: Верстат –автомат виготовляє в 1 годину КОЛ шайб. Скільки шайб він виготовить за 8-і годинну зміну, за робочий день при двозмінній роботі і за 30-ти денний місяць, якщо в місяці 4 вихідних дня і в кожній зміні верстат знаходиться 1 годину на профілактиці?

**Варіант 24.** Завдання 1. Написати програму обчислення об'єму куба. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, які вводяться користувачем, виділені напівжирним шрифтом).

Обчислення об'єму куба.  
Введіть довжину ребра (см) і натисніть клавішу <Enter> **9.5**  
Об'єм куба: 857.38 куб.см.

Завдання 2. Написати програму обчислення опору електричного ланцюга, що складається з двох паралельно сполучених резисторів.

**Варіант 25.** Завдання 1. Написати програму обчислення об'єму циліндра. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, які вводяться користувачем, виділені напівжирним шрифтом).

Обчислення об'єму циліндра.  
Введіть початкові дані:  
радіус підстави (см) > **5**  
висоту циліндра (см) > **10**  
Об'єм циліндра 1570.80 см. куб.  
Для завершення натисніть клавішу <Enter>.

Завдання 2. Написати програму обчислення опору електричний ланцюгу складається з трьох послідовно сполучених резисторів.

III Заключна частини заняття: відповіді на контрольні запитання і виконання контрольних завдань і збереження файлу проекту лабораторної роботи.

### **ПРАКТИЧНА РОБОТА №3 ОПЕРАТОРИ ЦИКЛУ**

**Мета роботи:** Освоїти прийоми роботи з текстовими редакторами й основними компонентами інтегрованого середовища програмування C++. Придбати практичні навички складання і налагодження програм із циклами

**Час проведення:** 2 години. **Місце проведення:** комп'ютерний клас.

**Навчальні питання:**

1. Цикл **for**
2. Оператор циклу з передумовою
3. Оператор циклу з післяумовою

## Література:

1. Подбельский В.В. Язык C++. 5-е издание. – М: Финансы и статистика, 2003. – 560 с.
2. Основи програмування мовою C++: Навчальний посібник/ Є.П. Путятін, В.П. Степанов, В.П. Пчелінов, Т.Г. Долженкова, О.О. Матат. – Х. ТОВ «Компанія СМІТ», 2005. – 320 с. (с.25-56).
3. Павловская Т. А. C/C++. Программирование на языке высокого уровня. / Т. А. Павловская - СПб. : Питер, 2006. - 461 с. : ил.

### Хід проведення заняття:

#### I. Вступ до заняття.

Під час підготовки до роботи повторити використання й організації циклів, а також використання інтегрованого середовища програмування на C++.

В мові C++ існують три оператори циклу **while**, **do**, **for**.

#### II. Порядок проведення основної частини заняття.

##### 1 Цикл **for**

- інструкція **for** використовується для організації циклів з фіксованим, відомим під час розробки програми, числом повторень;
- кількість повторень циклу визначається початковим значенням змінної-лічильника і умовою завершення циклу;
- змінна-лічильник повинна бути цілого (int) типу і може бути оголошена безпосередньо в інструкції циклу.

**Приклад.** Підсумувати всі парні числа від парного числа а до парного числа b ( $a < b$ ).

```
#include<iostream.h>
void main()
{
    int sum=0,a,b,i;
    cout<<"Vvedi a, a=";
    cin>>a;
    cout<<"Vvedi b, b=";
    cin>>b;
    for (i=a;i<b;i+=2)
        sum+=i;
    cout<<"Summa ravna sum="<<sum;
    cout<<"\n";
}
```

##### 2. Оператор циклу з передумовою

Оператор циклу з передумовою **while** виконується, якщо умова перевіряється до початку циклу, і має вигляд

**while**(вираз-умова) оператор; ,

де **оператор** – тіло циклу, що може бути представлено простим або складеним оператором, який взагалі не обчислюється, якщо **вираз-умова** має нульове значення (неправда), а керування передається наступному за циклом **while** оператору. Якщо **вираз-умова** відмінний від нуля (істинно), тоді обчислюється **оператор**, і керування передається до

початку циклу. У результаті тіло циклу **while оператор** виконується доки вираз прийме значення нуль (неправда). Коли вираз прийме значення нуль, керування буде передане наступному за циклом оператору. Таким чином, число повторень **while** визначається ходом виконання програми.

Наприклад:

```
// Обчислення 10!  
int i=1, p=1;  
while(i<=10)  
    {p*=i;  
      i++;  
    }
```

Для завершення циклу **while** в тілі циклу обов'язково повинні бути інструкції, виконання яких впливає, умова завершення циклу.

### 3. Оператор циклу з післяумовою

Оператор циклу з післяумовою **do while** застосовуються у випадках, коли тіло циклу виконується хоча б один раз і має таку форму запису:

**do оператор**  
**while(вираз-умова); .**

У процесі виконання оператора **do while** спочатку здійснюється вхід до тіла циклу і виконується **оператор**, який може простим або складеним, далі перевіряється вираз і, якщо він правдивий (істинно) – цикл повторюється, а коли вираз помилковий (неправда) – здійснюється вихід з циклу.

Наприклад:

```
int n=1,p1,p2;  
do {  
    p1=1./n;  
    n++;  
    p2=p1+1./n;  
}  
while(1./n<0.001);
```

Оператори циклу **do while**, як правило, використовуються для організації наближених обчислень, в задачах пошуку і обробки даних, що вводяться з клавіатури або з файлу.

Контрольні запитання і завдання

1. Які оператори циклу використовуються у C++?
2. Як працює оператор циклу **for**?
3. Поясніть на прикладах використання циклу з передумовою і циклу з післяумовою?

Завдання

Під час виконання роботи необхідно створити та відлагодити програму для розв'язання задачі, номер якої відповідає порядковому номеру студента в журналі академгрупи.

**Варіант 1.** Знайти суму цифр введеного з клавіатури числа.

**Варіант 2.** Напишіть програму, яка виводить на екран таблицю значень функції

$y = 2x^2 - 5x - 8$  в діапазоні від -4 до 4. Крок зміни аргументу 0,5.

**Варіант 3.** Написати програму, яка виводить таблицю квадратів перших десяти цілих позитивних чисел.

**Варіант 4.** Написати програму, яка виводить таблицю квадратів перших п'яти цілих позитивних непарних чисел.

**Варіант 5.** Написати програму, яка обчислює суму перших  $n$  цілих позитивних чисел. Кількість підсумованих чисел повинна вводитися під час роботи програми.

**Варіант 6.** Написати програму, яка обчислює суму перших  $n$  цілих позитивних парних чисел. Кількість підсумованих чисел повинна вводитися під час роботи програми.

**Варіант 7.** Написати програму, яка обчислює суму перших  $n$  членів ряду: 1, 3, 5, 7 ... Кількість підсумованих членів ряду задається під час роботи програми.

**Варіант 8.** Написати програму, яка обчислює суму перших  $n$  членів ряду:  
$$1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots$$

Кількість підсумованих членів ряду задається під час роботи програми.

**Варіант 9.** Написати програму, яка виводить таблицю ступенів двійки від нульової до десятої.

**Варіант 10.** Написати програму, яка обчислює факторіал введенного з клавіатури числа.

**Варіант 11.** Написати програму, яка виводить таблицю значень функції  $y = -2,4x + 5x - 3$  в діапазоні від -2 до 2, з кроком 0,5. Вид екрану під час виконання програми, що рекомендується, приведений нижче.

**Варіант 12.** Написати програму, яка вводять з клавіатури 5 дробових чисел і обчислює їх середнє арифметичне.

**Варіант 13.** Скласти програму, яка обчислює середнє арифметичне послідовності дробових чисел, яка вводиться з клавіатури. Кількість чисел повинна задаватися під час роботи програми.

**Варіант 14.** Скласти програму, яка вводять з клавіатури послідовність з п'яти дробових чисел і після введення кожного числа виводить середнє арифметичне введеній частині послідовності.

**Варіант 15.** Скласти програму, яка обчислює середнє арифметичне послідовності дробових чисел, що вводяться з клавіатури. Після введення останнього числа програма повинна вивести мінімальне і максимальне число послідовності. Кількість чисел послідовності повинна задаватися під час роботи програми.

**Варіант 16.** Скласти програму, яка генерує послідовності з 10 випадкових чисел в діапазоні від 1 до 10, виводить ці числа на екран і обчислює їх середнє арифметичне.

**Варіант 17.** Скласти програму, яка генерує три послідовності з десяти випадкових чисел в діапазоні від 1 до 10, виводить кожну послідовність на екран і обчислює середнє арифметичне кожній послідовності.

**Варіант 18.** Скласти програму, яка виводить на екран таблицю вартості, наприклад, яблук в діапазоні від 100 г до 1 кг з кроком 100 гр.

**Варіант 19.** Скласти програму, яка виводить таблицю значень функції  $y = |x|$ . Діапазон зміни аргументу -4 до 4, крок приросту аргументу 0,5.

**Варіант 20.** Скласти програму, яка виведе таблицю значень функцій  $y = |x - 2| + |x + 1|$ . Діапазон зміни аргументу від -4 до: 4, крок приросту аргументу 0,5.

**Варіант 21.** Скласти програму, яка виводить на екран таблицю множення, наприклад, на 7.

**Варіант 22.** Скласти програму, яка обчислює часткову суму ряду:  $1 - 1/3 + 1/5 - 1/7 + 1/9 - \dots$  і порівнює набуте значення з  $\pi/4$  (при підсумовуванні достатньо великої кількості членів цього ряду, величина часткової суми наближається до  $\pi/4$ ).

**Варіант 23.** Скласти програму наближеного обчислення інтеграла функції

$f(x) = 5x^2 - x + 2$  методом прямокутників.

**Варіант 24.** Скласти програму наближеного обчислення інтеграла методом трапецій.

**Варіант 25.** Скласти програму, що обчислює суму і середнє арифметичне послідовності позитивних чисел, які вводяться з клавіатури.

III Заключна частини заняття: відповіді на контрольні запитання і виконання контрольних завдань і збереження файлу проекту лабораторної роботи.

## **ПРАКТИЧНА РОБОТА №4 ВИКОРИСТАННЯ ФУНКЦІЙ ПІД ЧАС РОЗРОБКИ ПРОГРАМ**

**Мета роботи:** Освоїти розв'язання практичних завдань із застосуванням апарату функцій у мові C++.

**Час проведення:** 2 години. **Місце проведення:** комп'ютерний клас.

### **Навчальні питання:**

- 1 Визначення функції
- 2 Оголошення функції
- 3 Виклик функції

### **Література:**

1. Бюдбельский В.В. Язык C++. 5-е издание. – М: Финансы и статистика, 2004. – 560 с.
2. Основи програмування мовою C++: Навчальний посібник/ Є.П. Путятін, В.П. Степанов, В.П. Пчелінов, Т.Г. Долженкова, О.О. Матат. – Х. ТОВ «Компанія СМІТ», 2005. – 320 с. (с.25-56).
3. Павловская Т. А. C/C++. Программирование на языке высокого уровня. / Т. А. Павловская - СПб. : Питер, 2006. - 461 с. : ил.

### **Хід проведення заняття:**

#### **I. Вступ до заняття.**

Елементарною одиницею програми мовою C++ є функція. Функція призначена для вирішення визначеної задачі різного ступеня складності. Функції, що призначені для вирішення складних задач, можуть у свою чергу містити звертання до довільного числа функцій, призначених для вирішення менш складних задач. Таким чином, вирішення задачі мовою C++ припускає її функціональну декомпозицію, коли функції більш високого рівня забезпечують даними і сприймають результат функцій більш низького рівня. [1-7, конспект лекцій].

З використанням функцій в мові C++ пов'язані три поняття - визначення функції (опис дій, які виконуються функцією), оголошення функції (опис форми звертання до функції) і виклик функції.

#### **II. Порядок проведення основної частини заняття.**

##### **1 Визначення функції**

Визначення (опис) містить тіло функції - код, який виконується при звертанні до функції. Визначення функції задає тип значення, що повертається, ім'я функції, типи і кількість формальних параметрів, а також оголошення змінних і операторів, які називаються тілом функції, і визначають дію функції. У визначенні функції також може бути заданий клас

пам'яті.

Відповідно до синтаксису мови C визначення функції має наступну форму:

```
[специфікатор-класу-пам'яті] [специфікатор-типу] ім'я-функції (  
[список-формальних-параметрів]  
)  
{  
тіло-функції  
}
```

Необов'язковий специфікатор-класу-пам'яті задає клас пам'яті функції, який може бути static або extern.

Специфікатор-типу функції задає тип значення, що повертається і може задавати будь-який тип. Якщо специфікатор-типу не заданий, то передбачається, що функція повертає значення типу int.

Наприклад:

```
int rus (unsigned char r)  
{  
if (r>='A' && c<=' ')  
return 1;  
else  
return 0;  
}
```

У мові Cі немає вимоги, щодо визначення функції обов'язково до її виклику. Визначення використовуваних функцій можуть слідувати за визначенням функції main, перед ним, або знаходитися в іншому файлі.

Однак для того, щоб компілятор міг здійснити перевірку відповідності типів переданих фактичних параметрів типам формальних параметрів до виклику функції потрібно помістити оголошення (прототип) функції.

Функція повертає значення якщо її виконання закінчується оператором return, що містить деякий вираз. Для функцій, що не використовують значення, що повертається, повинен бути використаний тип void, який вказує на відсутність значення, що повертається.

Список-формальних-параметрів - це послідовність оголошень формальних параметрів, розділена комами. Формальні параметри - це змінні, використовувані усередині тіла функції і отримують значення при виконанні функції шляхом копіювання в них значень відповідних фактичних параметрів.

Якщо функція не використовує параметрів, то наявність круглих дужок обов'язкова, а замість списку параметрів рекомендується вказати слово void.

Порядок і типи формальних параметрів повинні бути однаковими у визначенні функції і у всіх її оголошеннях. Типи фактичних параметрів при виклику функції повинні бути сумісні з типами відповідних формальних параметрів.

Тіло функції - це складений оператор, що містить оператори, що визначають дію функції.

Всі змінні, оголошені в тілі функції без вказівки класу пам'яті є локальними. При виконанні функції локальним змінним відводиться пам'ять в стеку і проводиться їх ініціалізація. Управління передається першому оператору тіла функції і починається виконання функції, яке триває до тих пір, поки не зустрінеться оператор return або останній оператор тіла функції. Управління при цьому повертається в точку, наступну за точкою виклику, а локальні змінні стають недоступними. При новому виклику функції для локальних змінних пам'ять виділяється знову, і тому старі значення локальних змінних втрачаються.

## 2. Оголошення функції

Оголошення функції вказує її назву, тип параметра що повертається та атрибути функції, описаної в іншому місці програми. Оголошення функції повинно передувати зверненню до неї в тексті програми. Оголошення, яке також містить інформацію про кількість і тип переданих аргументів, називається також прототипом функції. Компілятор використовує прототип для звірки типів аргументів і необхідних перетворень типів в наступних викликах функції.

Для функції з попереднього прикладу прототип буде мати вигляд:

```
int rus (unsigned char r);
```

або

```
rus (unsigned char);
```

У програмах на мові C широко використовуються, так звані, бібліотечні функції, тобто функції попередньо розроблені і записані в бібліотеки. Прототипи бібліотечних функцій знаходяться в спеціальних заголовних файлах, що поставляються разом з бібліотеками в складі систем програмування, і включаються в програму за допомогою директиви `#include`.

Якщо оголошення функції не задано, то за замовчуванням будується прототип функції на основі аналізу першого посилання на функцію. Однак такий прототип не завжди узгоджується з наступним визначенням чи викликом функції. Рекомендується завжди задавати прототип функції. Це дозволить компілятору або видавати діагностичні повідомлення, при неправильному використанні функції, або коректним чином регулювати невідповідність аргументів при виконанні програми.

### 3 Виклик функції

Виклик функції передає управління з функції, що викликає, до функції, що викликається. Виконання виклику функції відбувається наступним чином:

1. Обчислюються значення в списку виразів і піддаються звичайним арифметичним перетворенням. Потім, якщо відомий прототип функції, тип отриманого фактичного аргументу порівнюється з типом відповідного формального параметра. Якщо вони не співпадають, то або виконується перетворення типів, або формується повідомлення про помилку. Число виразів в списку виразів має збігатися з числом формальних параметрів. Якщо в прототипі функції зазначено, що їй не потрібні параметри, а при виклику вони вказані, формується повідомлення про помилку.

2. Відбувається присвоювання значень фактичних параметрів відповідним формальним параметрам.

3. Управління передається першому оператор функції.

4. Виконання оператора `return` в тілі функції повертає управління і можливо, значення в функцію що визиває. При відсутності оператора `return` управління повертається після виконання останнього оператора тіла функції, а значення що повертається не визначено.

Контрольні запитання та завдання:

1. Для чого потрібен прототип функцій?
2. Чим прототип функції відрізняється від опису функції?
3. Як запрограмувати функції зі змінною кількістю параметрів?
4. Як правильно працювати з даними всередині функцій?
5. Організація роботи з масивами у функціях.
6. Назвіть засоби передачі параметрів у функціях.

Завдання

Під час виконання роботи необхідно створити та відлагодити програму для розв'язання задачі, номер якої відповідає порядковому номеру студента в журналі

академгрупи.

Варіант 1. Створити функцію знаходження мінімального елемента квадратної матриці і сортування її головної діагоналі.

Варіант 2. Розробити функцію, яка знаходить однакові елементи у двох масивах та використати її для обробки рядків матриці.

Варіант 3. Розробити функцію, що знаходить оптимуми математичної функції, заданої на деякому інтервалі.

Варіант 4. Для заданих  $m$  та  $n$  знайти число сполучень та перестановок, скориставшись функцією обчислення факторіала числа.

Варіант 5. Розробити функцію для обчислення середнього арифметичного та середнього геометричного масиву чисел.

Варіант 6. Створити функцію, яка міняє місцями перший і останній рядки квадратної і знаходить добуток елементів головної діагоналі.

Варіант 7. Створити функцію, яка вставляє задане число у відсортований масив будь-яку кількість разів.

Варіант 8. Розробити функцію, яка для довільної матриці знаходить мінімуми в рядках, а серед них - максимум.

Варіант 9. Перевірити, чи є задана матриця ортогональною, тобто скалярні добутки кожного рядка на інший дорівнюють 0.

Варіант 10. Написати функцію, яка в залежності від заданої ознаки проводить сортування масиву за зростанням або за зменшенням.

Варіант 11. Створити функцію сортування другого рядка матриці і знаходження її максимального елемента.

Варіант 12. Написати функцію, що розраховує відсотки за вкладом в залежності від виду та розміру вкладу,

Варіант 13. Створити функцію знаходження суми додатних елементів квадратної матриці і сортування головної діагоналі.

Варіант 14. Знайти шість найменших елементів масиву, користуючись функцією сортування елементів.

Варіант 15. Створити функцію знаходження суми діагоналей квадратної матриці і сортування її першого стовпця.

Варіант 16. Створити функцію знаходження суми елементів, що розташовані над головною діагоналлю квадратної матриці, і в тому випадку, коли ця сума парна, обнулити головну діагональ.

Варіант 17. Створити функцію, що визначає номери елементів, які зустрічаються у масиві тільки один раз,

Варіант 18. Створити функцію, яка у масиві обнуляє елементи, що знаходяться між максимумом та мінімумом.

Варіант 19. Створити функцію, яка змінює місцями перший і останній стовпці матриці квадратної матриці і знаходить суму елементів побічної діагоналі.

Варіант 20. Провести злиття відсортованих за зростанням масивів, розробивши відповідну функцію.

Варіант 21. Створити функцію, яка відсортовує елементи матриці за зростанням для кожного стовпця і знаходить мінімальний елемент.

Варіант 22. Створити функцію знаходження суми елементів квадратної матриці і сортування її головної діагоналі.

Варіант 23. Скориставшись функцією сортування масиву, відсортувати рядки матриці.

Варіант 24. Створити функцію знаходження добутку трикутної матриці, що знаходиться нижче головної діагоналі у квадратній матриці, та сортування її другого стовпця.

Варіант 25. Створити функцію, яка відсортовує елементи масиву до максимального елемента.

III Заключна частини заняття: відповіді на контрольні запитання і виконання контрольних завдань і збереження файлу проекту лабораторної роботи.

## ПРАКТИЧНА РОБОТА №5 РОЗВ'ЯЗАННЯ ЗАДАЧ З ВИКОРИСТАННЯМ МАСИВІВ

**Мета роботи:** Освоїти складання та реалізацію у середовищі Visual C++ програми обробки масивів.

**Час проведення:** 4 години. **Місце проведення:** комп'ютерний клас.

### Навчальні питання:

1. Оголошення одновимірного масиву
2. Використання одновимірного масиву

### Література:

1. Подбельский В.В. Язык C++. 5-е издание. – М: Финансы и статистика, 2003. – 560 с.
2. Основи програмування мовою C++: Навчальний посібник/ Є.П. Путятін, В.П. Степанов, В.П. Пчелінов, Т.Г. Долженкова, О.О. Матат. – Х. ТОВ «Компанія СМІТ», 2005. – 320 с. (с.25-56).
3. Павловская Т. А. C/C++. Программирование на языке высокого уровня. / Т. А. Павловская - СПб. : Питер, 2006. - 461 с. : ил.

### Хід проведення заняття:

#### I. Вступ до заняття.

Під час роботи з масивами треба звернути увагу на введення та виведення даних. На практиці часто виникає необхідність в обробці даних у вигляді довільного набору значень, тобто масивів.

Масив являє собою кінцеву іменовану послідовність величин одного типу, які розрізняються за порядковим номером. Опис масивів у програмі відрізняється від опису простої змінної наявністю після імені квадратних дужок «[]», в яких задається кількість елементів масиву (розмірність). У мові C++ нумерація елементів масиву починається з 0.

#### II. Порядок проведення основної частини заняття.

##### 1. Оголошення одновимірного масиву

Оголошення одновимірних масивів допускає одну з таких форм запису:

```
<тип> <ім'я>[n];  
<тип> <ім'я>[n]={значення};  
<тип> <ім'я>[ ]={значення};
```

При оголошенні одновимірного масиву, коли масив відразу ініціюється, можна не вказувати його розмір. Якщо ж ініціювання не здійснюється під час оголошення масиву, то кількість індексів слід задати обов'язково константним виразом. Надалі кількість елементів змінити неможливо.

##### 2. Використання одновимірного масиву

Наприклад. Обчислити функцію  $y = ax_i^2 - \sin x_i$ , аргументи якої  $x_i$  – елементи одновимірного масиву, що мають значення:

$x_0 = -0,81; x_1 = -0,58; x_2 = -0,11; x_3 = 0,2; x_4 = 0,91; x_5 = 1,83.$

```
#include<iostream.h>
#include<math.h>
#include<conio.h>
main()
{
    const int n=6;
    float x[n];
    float y,a=10.5;
    int i;
    //введення масиву
    cout<<"Введення масиву"<<endl;
    for(i=0; i<n; i++)
    {
        cout<<"Введіть"<<i<<"елемент";
        cin>>x[i];
    }
    //Обчислення функції
    for(i=0; i<n; i++)
    {
        y=a*pow(x[i],2)-sin(x[i]);
        cout<<"При значенні ="<<x[i]<<"y="<<y<<endl;
    }
    getch(); //затримка екрану
}
```

Алгоритм передбачає введення значень одновимірного масиву  $x_i (i = 0, \dots, n - 1)$ ,  $n = 6$  та подальше застосування для обчислення функції.

У програмі спочатку описується масив дійсних значень: **float** x[n]. Введення елементів масиву здійснюється у циклі, що реалізований оператором **for**, який містить операцію потокового введення елементів масиву **cin>>x[i]**; Зазначена операція виконується **n** разів. Програма використовує два цикли: один – для введення масиву, інший – для обчислення функції та виведення результатів. Операції введення елементів масиву та обчислення значень функції можна здійснити в одному циклі.

Типовою помилкою при використуванні масивів є звернення до неіснуючого елемента, тобто вихід індексу за допустиме значення.

Контрольні запитання і завдання

- 1.Що таке масив та які існують різновиди масивів?
2. Як здійснюється звернення до елементів масивів?
3. Як у C++ реалізується введення-виведення елементів масиву?

Завдання

Під час виконання роботи необхідно створити та відлагодити програму для розв'язання задачі, номер якої відповідає порядковому номеру студента в журналі академгрупи.

**Варіант 1.** Написати програму, яка вводить з клавіатури одновимірний масив з 5 цілих чисел, після чого виводить кількість ненульових елементів. Перед введенням кожного елемента повинна виводитися підказка з номером елемента.

**Варіант 2.** Написати програму, яка виводить мінімальний елемент введеного з

клавіатури масиву цілих чисел.

**Варіант 3.** Написати програму, яка обчислює середнє арифметичне ненульових елементів введеного з клавіатури масиву цілих чисел.

**Варіант 4.** Написати програму, яка обчислює середнє арифметичне елементів масиву без урахування мінімального і максимального елементів масиву.

**Варіант 5.** Написати програму, яка обчислює середню (за тиждень) температуру повітря. Початкові дані повинні вводитися під час роботи програми.

**Варіант 6.** Написати програму, яка перевіряє, чи знаходиться введене з клавіатури число в масиві. Масив повинен вводитися під час роботи програми.

**Варіант 7.** Написати програму, яка перевіряє, чи представляють елементи введеного з клавіатури масиву зростаючу послідовність.

**Варіант 8.** Написати програму, яка обчислює, скільки разів введене з клавіатури число зустрічається в масиві.

**Варіант 9.** Написати програму, яка перевіряє, чи є у введеному з клавіатури масиві елементи з однаковим значенням.

**Варіант 10.** Написати програму, яка методом прямого вибору сортує по убутанню введений з клавіатури одновимірний масив.

**Варіант 11.** Обчислити суму елементів масиву з непарними номерами.

**Варіант 12.** Написати програму, яка об'єднує два впорядкованих за збільшенням масиву в один, також упорядкує масив.

**Варіант 13.** Стиснути масив, видаливши з нього елементи, модуль яких не перевищує 1. Елементи, що звільнилися заповнити нулями.

**Варіант 14.** Написати програму, яка визначає кількість учнів у групі, чий зріст перевищує середній.

**Варіант 15.** Визначити номери мінімального і максимального елементів масиву.

**Варіант 16.** Написати програму, яка обчислює середнє арифметичне модулів елементів масиву, розташованих після першого негативного елементу.

**Варіант 17.** Написати програму, яка обробляє результати іспиту. Для кожної оцінки програма повинна обчислювати відсоток від загальної кількості оцінок.

**Варіант 18.** Написати програму, яка перетворює масив таким чином, щоб спочатку розташовувалися всі елементи, абсолютна величина цілої частини яких не перевищує 1. а потім всі інші.

**Варіант 19.** Написати програму, яка перетворює масив таким чином, щоб спочатку розташовувалися всі елементи, що відрізняються від максимального не більш ніж на 20%, а потім всі інші.

**Варіант 20.** Написати програму, яка визначає кількість елементів, що знаходяться у діапазоні від  $a$  до  $b$ .

**Варіант 21.** Написати програму, яка визначає номер максимального за модулем елементу вектора.

**Варіант 22.** Написати програму, яка визначає кількість учнів в групі, чия вага перевищує значення.

**Варіант 23.** Написати програму, яка визначає кількість позитивних елементів масиву.

**Варіант 24.** Написати програму, яка визначає кількість негативних елементів масиву.

**Варіант 25.** Написати програму, яка визначає середнє арифметичне елементів масиву, менших за  $C$ .

III Заключна частини заняття: відповіді на контрольні запитання і виконання контрольних завдань і збереження файлу проекту лабораторної роботи.

## 4. Список рекомендованої літератури

### Базова

1. Глушаков С.В., Коваль А.В., Смирнов С.В. Язык программирования C++. Учебный курс. – Харьков.:Фолио, 2003,
2. Подбельский В.В. Язык C++. 5-е издание. – М: Финансы и статистика, 2003. – 560 с.
3. Аммерааль Л. STL для программистов на C++. : Пер. с англ. / Л. Ам-мерааль - М. : ДМК, 1999. - 240 с. : ил.
4. Вирт Н. Алгоритмы и структуры данных ; пер. с англ. / Н. Винт М. : Мир, 1989. - 360 с., : ил.
5. Деннис М. Язык программирования С. ; Пер. с англ. / Деннис М. Ритчи, Брайан У. Керниган. - М. : Издательский дом "Вильямс", 2009. - 304 с.
6. Ишкова Э. А. C++. Начала программирования. 3-е Изд., перераб. и доп. / Э. А. Ишкова. - М. : ООО "БИНOM-Пресс", 2004. - 368 с. : ил.
7. Кнут Д. Искусство программирования для ЭВМ / Д. Кнут - т.1-3, М. : Мир, 1976 - 1977.
8. Пирогов В. Ю. Программирование на Visual C++.NET. / В. Ю. Пирогов.СПб. : БХВ-Петербург, 2003. - 800 с.
9. Румянцев П. В. Азбука программирования в Win32 API. / П. В. Румянцев - М. : Горячая линия - Телеком. 2001. - 312 с.
10. Харви Дейтел Как программировать на C++ ; пер. с англ. / Харви Дейтел, Пол Дейтел. - М. : ЗАО "Издательство БИНОМ", 2008. - 1455 с.
11. Харт, Джонсон М. Системное программирование в среде Windows, 3-е изд. ; пер. с англ. / Харт, Джонсон М. - М. : Издательский дом "Вильямс", 2005. - 592 с. : ил.
12. Щупак Ю. А. Эффективная разработка приложений. / Ю. А. Щупак СПб. : Питер, 2007. - 572 с. : ил.
13. Хортон А. Visual C++ 2005: базовый курс ; пер. с англ. / А. Хортон. М. : Издательский дом "Вильямс", 2007. - 1152 с. : ил.

### Допоміжна

14. Верма Р. Д. Справочник по функциям Win32 API. / Р. Д. Верма -М. : Горячая линия - Телеком, 2002. - 488 с.
15. Марченко А. Л. C++. Бархатный путь. / А. Л. Марченко - М. : Горячая линия Телеком, 1999. - 400 с.
16. Павловская Т. А. С/C++. Программирование на языке высокого уровня. / Т. А. Павловская - СПб. : Питер, 2006. - 461 с. : ил.
17. Подбельский В. В. Язык C++ : Учебн. пособие. / В. В. Подбельский - 4-е изд. - М. : Финансы и статистика, 1999. - 560 с.
18. Подбельский В. В. Программирование на языке С : - 2-е доп. Изд. / В. В. Подбельский, С. С. Фомин - М.: Финансы и статистика, 2004. - 600 с.
19. Саймон Р. Windows 2000 API. Энциклопедия программиста; пер. с англ. / Р. Саймон - К. : ООО "ДиасофтЮП", 2002. - 1088 с.
20. Страуструп Б. Язык программирования C++, 3-е изд. Специальное издание; пер. с англ. / Б. Страуструп. - СПб.; М. : "Невский диалект" -Издательство "БИНOM", 2004. -1104с. : ил.
21. Страуструп Б. Дизайн и эволюция языка C++: пер. с англ. / Б. Страуструп - М. : ДМК, 2000. - 444 с.

22. Шеферд Д. Программирование на Microsoft Visual C++.NET. : пер. с англ. / Д. Шеферд - М. : Издательско-торговый дом "Русская редакция", 2003. - 928с. : ил.

## **10.2. Інформаційні ресурси**

23. Главная страница MSDN, MSDN по-русски. - Режим доступа : <http://msdn.microsoft.com/ru-ru/default.aspx>.

24. Интернет Университет Информационных Технологий - дистанционное образование. - Режим доступа : <http://www.intuit.ru/>

25. CodeNet - Все для программиста. - Режим доступа : <http://www.codenet.ru/>.

26. C++ для начинающих. - Режим доступа : <http://valera.asf.ru/cpp/book/>.

27. MSDN-WindowsAPI. - Режим доступа : [http://www.vsokovikov.narod.ru/New\\_MSDN\\_API/index\\_msdn.htm](http://www.vsokovikov.narod.ru/New_MSDN_API/index_msdn.htm).