

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Запорізький національний технічний університет

МЕТОДИЧНІ ВКАЗІВКИ
до виконання лабораторних та контрольних робіт
із дисципліни "Програмування" (І частина)
для студентів спеціальності 123 "Комп'ютерна інженерія"
заочної форми навчання

2018

Методичні вказівки до виконання лабораторних робіт із дисципліни Методичні вказівки до виконання лабораторних та контрольних робіт із дисципліни "Програмування" (І частина) для студентів спеціальності 123 "Комп'ютерна інженерія" заочної форми навчання / Укл.: Н.В. Луценко. – Запоріжжя: ЗНТУ, 2018. – 38 с.

Укладачі:

Н.В. Луценко, ст. викладач

Рецензент:

В.В. Шкарупило, к.т.н., доцент

Відповідальний за випуск

Н.В. Луценко, ст. викладач

Затверджено
на засіданні кафедри КСМ
Протокол № 8 від 25.04.2018

Рекомендовано до видання
на засіданні НМК факультету КНТ
Протокол № 9 від 27.04.2018

ЗМІСТ

Порядок виконання лабораторних та контрольних робіт	4
Загальні відомості	5
Лабораторні роботи	16
Лабораторна робота № 1. Базові типи змінних. Введення та виведення даних.....	16
Лабораторна робота № 2. Оператори вибору. Оператори циклу .	18
Лабораторна робота № 3. Масиви і покажчики	21
Варіанти завдань до контрольної роботи.....	24
Завдання № 1. Базові типи змінних. Введення та виведення даних	24
Завдання № 2. Оператори вибору. Оператори циклу	26
Завдання № 3. Масиви і покажчики.....	31
Контрольні питання до контрольної роботи	37
Рекомендована література.....	38

ПОРЯДОК ВИКОНАННЯ ЛАБОРАТОРНИХ ТА КОНТРОЛЬНИХ РОБІТ

Лабораторні роботи складаються з декількох програм, завдань до них. Студент повинен розробити, налагодити і зберегти у робочому каталозі програми в точній відповідності з приведеними в лабораторних роботах текстами програм (вихідні програми) та проаналізувати результати їхньої роботи. Розробити для кожної вихідної програми самостійне завдання. Продемонструвати викладачу роботу програм.

Контрольна робота складається з декількох завдань з варіантами. Студент повинен створити, налагодити і зберегти програми згідно заданого варіанту, проробити і проаналізувати результати їхньої роботи та дати відповіді на контрольні питання.

Зміст контрольної роботи

1. Титульний лист з номером варіанту.
2. Тексти завдань згідно варіанту.
3. Тексти програм.
4. Відповіді на контрольні питання.
5. Носій з відлагодженими програмами.

Поради та рекомендації

Запускаємо середовище програмування **Microsoft Visual C++ 2010 Express Edition**. При цьому відкривається початкова сторінка, в якій потрібно створити пустий проект. Для цього обирається вкладка **File (Файл)** та пункт **New->Project (Создать->Проект)**. В вікні, що з'явилося, обирається **Visual C++**, а справа обирається значок **Empty project (Пустой проект)**. Нижче задається назва проекту та вказується папка, в якій буде зберігатися цей проект.

Для додавання в проект файлу з кодом програми використовується меню **Project (Проект)** та пункт **Add New Item (Добавить новый элемент)**. В вікні, що з'явилося, обирається значок **C++File.cpp (Файл.cpp)**, а нижче задається ім'я нового файлу. Далі натискається кнопка **Add (Добавить)**. Відкривається сторінка редактора, в якому пишеться код програми.

Для запуску програми обирається значок ► (**Start Debugging**).

Для відкриття написаної програми обирається вкладка **File (Файл)** та пункт **Open->Project (Открыть->Проект)**.

ЗАГАЛЬНІ ВІДОМОСТІ

Програма мовою Сі – це послідовність препроцесорних директив, описів та визначень глобальних змінних та функцій.

У програмі повинна бути визначена одна головна функція з ім'ям **main()**. Крім цієї функції в програмі можна визначити будь-яку кількість інших функцій. Визначення функції складається з заголовка і тіла. Всі змінні та інші об'єкти, які визначені поза функцій, є зовнішніми, або глобальними.

Компілятор перетворює текст програми з мови Сі в об'єктний (машинний) код, виконує синтаксичну перевірку тексту програми і видає діагностичні повідомлення про помилки. При цьому формується файл програми з розширенням **obj**.

Компоновщик підключає необхідні бібліотечні функції, оформляє код програми відповідно з операційною системою комп'ютера і формує виконуваний файл програми з розширенням **.exe**. При цьому, якщо якусь функцію не вдалося підключити до програми, компоновщик видає діагностичні повідомлення.

Препроцесорні директиви – це записані в тексті програми інструкції, які починаються з нового рядка зі знаку **#**. Вони виконуються препроцесором до компіляції, доповнюючи вхідний текст програми.

Директива #include включає в текст програми зміст зазначеного файлу. Ця директива має дві форми:

```
#include "ім'я_файлу"
```

```
#include <ім'я_файлу>
```

Директива #define виконує заміну ідентифікаторів константами або будь-якою послідовністю операторів (виразів), що найчастіше використовуються. Ця директива може розміщуватися в будь-якому місці тексту програми, а дія її починається від точки розміщення до кінця тексту. Синтаксис директиви **#define** такий:

```
#define ідентифікатор рядок_заміщення
```

Заміни в тексті можна відмінити директивою **#undef**:

```
#undef ідентифікатор
```

Наприклад, перша директива замінить у тексті програми кожний символ **W** на число **80**, а друга директива відмінить цю заміну.

```
#define W 80
```

```
#undef W
```

Базові типи даних. Для визначення змінних цілого типу даних використовують ключові слова **char**, **int**, **short**, **long**. Слова **signed**, **unsigned** позначають знакове чи беззнакове ціле число, за умовчанням всі числа є знакові. Для визначення змінних дійсного типу даних використовують ключові слова **float**, **double**.

Базові цілі типи даних

Тип	Розмір	Діапазон
char	1 байт (8 біт)	$-2^7 \div 2^7 - 1$
unsigned char	1 байт (8 біт)	$0 \div 2^8 - 1$
short	2 байт (16 біт)	$-2^{15} \div 2^{15} - 1$
unsigned short	2 байт (16 біт)	$0 \div 2^{16} - 1$
int	4 байт (32 біт)	$-2^{31} \div 2^{31} - 1$
unsigned int	4 байт (32 біт)	$0 \div 2^{32} - 1$
long	4 байт (32 біт)	$-2^{31} \div 2^{31} - 1$
unsigned long	4 байт (32 біт)	$0 \div 2^{32} - 1$

Базові дійсні типи даних

Тип	Розмір	Діапазон
float	4 байт (32 біт)	$\pm 3.4 \cdot 10^{-38} \div \pm 3.4 \cdot 10^{+38}$
double	8 байт (64 біт)	$\pm 1.7 \cdot 10^{-308} \div \pm 1.7 \cdot 10^{+308}$

Константи цілого типу за умовчанням мають тип **int**. Для завдання константи типу **long** дописується символ **L|l**, типу **unsigned** – символ **U|u**): **234L**, **456u**, **23456Lu**. Для завдання константи в 8-й системі числення (СЧ) на початку константи дописується **0**, в 16-й СЧ – **0x**: **020**, **0x4a**.

Константи дійсного типу за умовчанням мають тип **double**. Для завдання константи типу **float** дописується символ **F|f**: **23.45E-6f**.

Елементарні введення та виведення даних

Форматоване виведення даних на екран здійснюється функцією: **printf(рядок_формату, список_виведення)**;

Список_виведення може містити константи, змінні та будь-які вирази, розділені комами. **Рядок_формату** записується в подвійних лапках і може містити: послідовність звичайних символів, що виводяться на екран; спеціальні символи, що керують розміщенням даних на екрані (**ESC**-послідовності); специфікатори формату даних.

Специфікатори формату – це наступна послідовність:

% [довжина поля] [.точність] [l або L] специфікатор_типу

де **[довжина поля]** – кількість позицій для виведення; **[.точність]** – кількість цифр дробової частини дійсних чисел, чи кількість символів для рядків, що виводяться; **l** або **L** – позначка для чисел типу **long**, **double**.

Для виведення даних використовують такі специфікатори типів:

%c – ціле число у вигляді символу;
%d – знакове ціле число у 10-ій системі числення (СЧ);
%u – беззнакове ціле число у 10-ій СЧ;
%o %x – ціле число типу у 8-ій СЧ або в 16-ій СЧ;
%ld – знакове ціле число типу **long** у 10-ій СЧ;
%lu – беззнакове ціле число типу **long** у 10-ій СЧ;
%lo %lx – ціле число типу **long** у 8-ій СЧ або в 16-ій СЧ;
%f – дійсне число типу **float**;
%e – дійсне число типу **float** в експоненціальній формі;
%lf – дійсне число типу **double**;
%le – дійсне число типу **double** в експоненціальній формі;
%s – рядок;
%p – адреса.

Форматоване введення з клавіатури даних виконується функцією:

scanf (рядок_формату, список_введення) ;

Список_введення може містити тільки адреси змінних, раніш визначених у програмі. **Рядок_формату** записується в подвійних лапках і може містити тільки специфікатори типів даних.

Функція **scanf()** здійснює введення чисел, символів та рядків з клавіатури до натискання клавіші **<Enter>** або до виявлення першого неприпустимого символу у вхідному потоці.

```
int a; float b; char c; char st[20];
scanf("%d %f %c %s",&a,&b,&c,st);
printf("a=%d, b=%f, c=%c st=%s \n",a,b,c,st);
```

Неформатоване введення символів з клавіатури здійснюється функціями **getchar()**, **getch()**, **getche()**, а неформатоване виведення символів на екран – **putchar()**, **putch()**.

```
char c1,c2,c3;
c1=getchar(); // введення символу з натисканням клавіші <Enter>
c2=getch();   // введення без клав. <Enter> і без виведення на екран
c3=getche();  // введення без клав.<Enter> і з виведенням на екран
putchar(c1);  // виведення символу на екран
putch(c2);    // виведення символу на екран
```

Неформатоване введення рядків виконується функцією `gets()`, а неформатоване виведення рядків – `puts()`.

```
char st1[20];
gets(st1); //введення рядка до натискання клавіші <Enter>
puts(st1); //виведення рядка на екран з переведенням на новий рядок
```

В мові C++ введення даних може здійснюватися за допомогою функції `cin>>`, а виведення даних – за допомогою функції `cout<<`.

```
int a; cout<<"Input a"<<endl; //виведення на екран повідомлення
cin>>a; // введення з клавіатури значення змінної a
cout<<a; // виведення на екран введеного значення
```

Маніпулятори: `dec`, `oct`, `hex` – завдання системи числення; `endl` – переведення на новий рядок; `left`, `right` – вирівнювання вліво чи вправо; `setw(4)` – завдання ширини поля; `showpoint`, `scientific` – завдання точності, ширини.

```
char ch1,ch2; double d1,d2;
cin>>ch1>>d1>>d2>>ch2; // введення чисел
cout<<dec<<ch1<<oct<<a<<hex<<a<<endl; // виведення чисел
cout<<setw(5)<<ch2; // виведення числа з шириною 5 позицій
```

ESC-послідовності (керуючі послідовності) – це послідовності символів, які використовують для запису неграфічних та деяких інших символів в рядках та символьних константах. ESC-послідовність складається з нахиленої риски вліво (зворотній слеш), після якого записується буква або знак. Наприклад: `\n` – новий рядок; `\t` – горизонтальна табуляція; `\'` – одиночні лапки; `\"` – подвійні лапки; `\\` – нахилена вліво риска; `\?` – знак питання.

Умовний оператор `if...else` обирає один з двох варіантів виразів обчислень та має такий синтаксис:

```
if(умова) вираз_1 else вираз_2
```

Якщо *умова* – істина, то виконується *вираз_1*, інакше – *вираз_2*. Будь-яке значення *умови*, крім 0, вважається істиною. Наприклад:

```
if(x>y) {max=x; min=y;}
else {max=y; min=x;}
```

Допустима скорочена форма умовного оператора, в якій відсутня конструкція **`else вираз_2`**. В цьому разі при хибності перевіряємої умови ніяких дій не виконується. Наприклад:

```
if(x>max) x=max;
res=x; // виконується завжди
```

Оператор-перемикач `switch` дозволяє обрати один з декількох варіантів обчислень та повідомлень. Синтаксис цього оператора такий:

```
switch(перемикаючий_вираз)
{ case константний_вираз_1: оператори_1; break;
  case константний_вираз_2: оператори_2; break;
  . . .
  default: оператори_n+1;
}
```

Оператор **`switch`** порівнює значення **перемикаючого_виразу**, який може приймати тільки цілі значення, з **константними_виразами** та передає керування тому оператору, **константний_вираз** якого співпадає зі значенням **перемикаючого_виразу**. Якщо такі оператори відсутні, то виконуються оператори, що позначені міткою **`default`**. Оператор **`break`** здійснює примусовий вихід з оператора **`switch`**. Якщо оператор **`break`** відсутній, то виконуються не тільки оператори відповідної гілки **`case`**, а й усі інші гілки до кінця оператора **`switch`**. Оператори **`break`** та **`default`** необов'язкові в операторі **`switch`**. Наприклад:

```
unsigned int a;
puts("Input value:");
scanf("%u",&a); // введення з клавіатури значення змінної a
switch(a)
{ case 0: puts("це нуль"); break;
  case 1: puts("це одиниця"); break;
  case 2: puts("це двійка"); break;
  case 3: puts("це трійка"); break;
  default: puts("це більше трьох");
}
```

Оператори циклу. В мові C++ є три оператори, за допомогою яких можна організувати цикли: **`for`**, **`while`**, **`do...while`**.

Цикл з оператором **`for`** частіше використовується у випадку, коли відома точна кількість повторів. Цей оператор має синтаксис:

```
for(ініціалізація_циклу; умова; список_виразів) тіло_циклу
```

Ініціалізація_циклу – це послідовність визначень та виразів, розділених комами, що виконуються один раз перед початком виконання циклу. Потім перевіряється **умова**. Якщо значення **умови** не дорівнює 0 (тобто є істиною), то виконується **тіло_циклу**, а потім **список_виразів** та знову перевіряється **умова**. При хибному значенні **умови** цикл завершується. Наприклад:

```
for(int i=0,sum=0,pr=1; i<3; i++) // i=1,2,3
{sum+=5; // sum=5,10,15
 pr*=(i+1); // pr=1,2,6
}
```

Цикл з оператором while використовується в тому випадку, коли невідоме точне число повторів. Синтаксис оператора наступний:

while(умова) тіло_циклу

Спочатку перевіряється **умова**, якщо **умова** є істиною, то виконується **тіло_циклу** і знову перевіряється **умова**. При хибному значенні **умови** цикл завершується. Наприклад:

```
int a=0, sum=0;
while(sum<20)
{a+=2; // a=2,4,6,8
 sum+=a; // sum=2,6,12,20
}
```

Цикл з оператором do...while використовується в тих випадках, коли **тіло_циклу** повинно бути виконане, принаймні, один раз. Синтаксис оператора наступний:

do тіло_циклу while(умова);

Спочатку виконується **тіло_циклу**, потім перевіряється **умова**. Якщо **умова** є істиною, то знову виконується **тіло_циклу**. При хибному значенні **умови** цикл завершується.

Наприклад:

```
int m=0,n=0;
do{m+=5; n++; // m=5,10,15 n=1,2,3
 } while(n<3);
```

Оператори циклу можуть мати оператори **continue**, **break**, **return**. Оператор **continue** пропустить залишок тіла циклу та почне виконання наступної ітерації (повторення). Оператор **break** завершить виконання усього циклу. Оператор **return** завершить виконання функції або всієї програми.

```
int n1=0,n2=0,a,sum=0;
while((n1+n2)<20) //перевірка кількості введених значень змінної a
{scanf("%d",&a); // введення 20 значень змінної a
 if((a%2)==0)
 {n2++; continue;} // перехід до наступної ітерації, якщо a парне,
 if(a==99) break; // закінчення циклу, якщо a=99
 if(a==100) return; // закінчення програми, якщо a=100
 n1++; sum+=a; } // обчислення суми непарних змінних a
```

Оператор безумовного переходу `goto` передає керування на будь-який позначений міткою оператор в тілі функції. При використанні оператора `goto` не можна входити усередину умовного оператора, перемикача, циклу. Наприклад:

```
if (n==n_max) goto M1; // перехід на M1 при n==n_max
n++;
M1: printf("n=%d", n);
```

Одновимірні масиви. Масив – це сукупність змінних одного типу, що об'єднані під одним ім'ям. В пам'яті елементи розташовуються послідовно.

Масиви бувають різного класу пам'яті. Масиви, визначені поза функціями, у тому числі поза функцією `main()`, є глобальними (зовнішніми). Масиви, які визначені усередині функції, можуть бути статичними (зі словом `static`) або автоматичними.

При визначенні масиву задають його розмір, значення розміру повинно бути константним цілочисловим виразом.

Три способи завдання розміру масиву :

1) цілою числовою константою

```
int a1[10]; //масив елементів типу int, що складається з 10 елементів
```

2) пойменованою константою - із службовим словом `const`;

```
const int M=4;
```

```
float f1[M]; //складається з 4 елементів типу float
```

3) за допомогою препроцесорної директиви `#define`.

```
#define N 20
```

```
char c1[N]; //масив, що містить 20 символів.
```

Доступ до елементів масиву може здійснюватися через індекси, причому перший елемент масиву має індекс **0**. Наприклад:

```
int a[3];
```

```
a[0]=2;
```

```
a[1]=6;
```

```
a[2]=a[0]+a[1]*5; // a[2]=32
```

Для роботи з масивами часто застосовуються оператори циклу, найбільш часто використовується оператор циклу `for`.

Наприклад:

```
int a[10], sum1=0, sum2=0, k=0, i;
```

```
puts("Введіть масив");
```

```
for (i=0; i<10; i++)
```

```
scanf ("%d", &a[i]); // введення масиву з клавіатури
```

```

for (i=0; i<10; i++)
{
    sum1+=a[i]; // обчислення суми усіх елементів масиву
    if (a[i]%2==0)
    {
        k++; // обчислення кількості парних елементів масиву
        sum2+=a[i]; // обчислення суми парних елементів масиву
    }
}
printf("sum1=%d sum2=%d\n", sum1, sum2);
printf("sr=%f\n", float(sum2)/k); // середнє ариф. парних ел-ів

```

Покажчики – це цілі беззнакові змінні, що містять адреси пам'яті якого-небудь об'єкту програми. Покажчик може містити адресу змінної, масиву, структури, іншого покажчика і т.п. Розмір покажчика – 4 байти: 2 байта – сегмент і 2 байта – зсув в сегменті. Тип покажчика – це тип змінної, адреса якої може знаходитися в покажчику.

Покажчик можна ініціалізувати або йому можна привласнювати:

- 1) адресу раніш визначеної змінної,
- 2) значення раніш визначеного іншого покажчика,
- 3) явно задану адресу пам'яті (з перетворенням типу).

```

int a=1;
int *ptrb; // покажчики ptrb, ptrb
ptrb=&a; // адреса змінної a
ptrb=ptrb; // значення покажчика ptrb
int *ptrc=(int*)0xB8000000; // явно адреса пам'яті

```

Значення покажчика збільшується (при додаванні цілого числа **N**) або зменшується (при відніманні цілого числа **N**) на відповідну кількість змінних того типу, адресу яких вмістить покажчик. Тобто значення покажчика завжди змінюється на кількість байтів, що дорівнює **N*sizeof(тип_покажчика)**.

Різниця покажчиків показує, скільки змінних відповідного типу може розміститися між цими покажчиками.

Для звернення (доступу) до об'єкта, адресу якого містить покажчик, використовується операція розіменування *****.

```

float c;
float *ptrc=&c;
*ptrc=1.2; // зміна c=1.2

```

Основні операції, які можна виконувати з покажчиками:

- 1) розіменування, 2) привласнення адреси, 3) додавання і віднімання цілих чисел, 4) інкремент і декремент, 5) віднімання покажчиків, 6) операції порівняння, 7) визначення адреси покажчика.

Ім'я масиву – це покажчик-константа, значення якого є адреса нульового елемента масиву. Це значення не можна змінити. Наприклад, якщо визначити `int arr[8]`, то при цьому виділиться пам'ять для 8 елементів масиву, а ім'я `arr` стане покажчиком-константою на цей масив. При визначенні розміру покажчика `arr` визначається розмір усього масиву. Ім'я масиву можна привласнювати іншим покажчикам. З елементами масивів, зв'язаними з покажчиками, можна працювати за допомогою індексів і операції розіменування. Наприклад:

```
int arr[8]; int *ptr=arr;
arr[0]=10; *(arr+1)=20; *(2+arr)=30; 3[arr]=40;
ptr[4]=50; *(ptr+5)=60; *(6+ptr)=70; 7[ptr]=80;
```

Рядок – це послідовність елементів типу `char`, останнім елементом якої є `'\0'`. Ініціалізується рядок послідовністю символів у парних подвійних лапках. Адреса першого символу може використовуватися для ініціалізації масиву типу `char`, у цьому випадку ім'я масиву й адреса першого символу стають синонімами. Наприклад:

```
char str1[5]="name";
char str2[5]={'n','a','m','e','\0'};
```

Адреса першого елемента рядка може використовуватися для ініціалізації покажчика типу `char`, наприклад:

```
char *stptr="name" ;
```

У цьому випадку змінна-покажчик `stptr` (але не покажчик-константа) одержує значення адреси символу `'n'`, а сам рядок розташовується в сегменті даних завантажувального модуля програми.

Масив покажчиків – це масив, елементами якого є покажчики. Приклад визначення масиву з трьох покажчиків на об'єкти типу `int`:

```
int a,b,c;
int*mas[3]={&a,&b,&c};
```

Ім'я масиву – `mas`, він містить три покажчика типу `int*`. Вираз `(mas+1)` відповідає зсуву у пам'яті на `sizeof(int*)` байтів від початку масиву (тобто на 4 байти).

Масиви покажчиків на рядки частіше використовуються для раціонального розміщення в пам'яті й сортування рядків. Наприклад:

```
char *name[3]={"Іванов","Петро","Єгорович"};
```

При такому визначенні в пам'яті виділяється `3*sizeof(char*)=12` байт для покажчиків `name[0]`, `name[1]`, `name[2]`, 7 байт – для рядка "Іванов", 6 – для "Петро" і 9 – для "Єгорович".

Багатовимірний масив – це масив масивів, тобто такий масив, елементами якого є масиви.

Ім'я багатовимірного масиву є показчик-константа на масив показчиків-констант, елементами якого є показчики-константи з адресами початку кожного з рядків масиву елементів. Елементи багатовимірного масиву розташовуються в пам'яті за рядками, тобто в порядку зростання найправішого індексу. Доступ до елементів двовимірного масиву може здійснюватися за індексом або за допомогою показчиків. Наприклад, звернення до `arr[i][j]` еквівалентно `*(arr[i]+j)`, `*(*(arr+i)+j)` або `*(arr+i)[j]`.

Ініціалізація масивів – це привласнення початкових значень елементам масивів при їх визначенні. Ініціалізація масивів, може здійснюватися за умовчанням та явно. За умовчанням глобальні і статичні масиви ініціалізуються нулями, автоматичні – довільними даними. Явна ініціалізація буває повна (з завданням кількості елементів і початкових значень усіх елементів масиву) та неповна (з завданням кількості елементів, але початкові значення задаються не для всіх елементів масиву):

```
int a[5]={2,5,7,9,1}; // явна повна ініціалізація
int b[7]={5,7,9};      // явна неповна ініціалізація
```

У останньому рядку інші чотири елементи масиву `b` будуть мати або нульові значення (для зовнішнього і статичного масивів), або їхні значення будуть невизначені (для автоматичного масиву).

При повній явній ініціалізації одновимірного масиву можливо не задавати розмір масиву. Компілятор визначає кількість елементів масиву за списком ініціалізації:

```
int c[]={5,7,9,8}; // Масив із чотирьох елементів
```

Нижче наведені приклади ініціалізації двовимірних масивів. При явній ініціалізації багатовимірних масивів можливо вводити додаткові скобки.

```
int m[3][2]={2,4,6,8,1,3}; // явна повна ініціалізація
int n[2][3]={ {1,9,3} {2,8,4} }; // явна повна ініціалізація
int d[4][3]={ {1}, {2}, {3}, {4} }; // явна неповна ініціалізація
```

Багатовимірні масиви можна ініціалізувати без вказівки найлівої розмірності:

```
int p[][3]={ {3,2,1}, {3,8,9}, {2,7,4} };
float f[][2]={ {1.2}, {2.2,3.4}, {2.5,6.4}, {2.0} };
```

В цьому прикладі визначені масиви `int p[3][3]` і `float f[4][2]`.

Масиви динамічної пам'яті. Для даних такого масиву виділяється динамічна пам'ять під час виконання програми в так званій "купі", розмір і положення якої залежать від моделі пам'яті. При цьому розмір масиву можна визначити як змінну цілого типу. Це зручно у випадку, коли невідомо, скільки об'єктів буде створено при роботі програми.

Для формування масивів динамічної пам'яті використовуються покажчики заданого типу та спеціальні функції і операції.

В мові Сі використовується функція **malloc()** для запиту і виділення потрібної кількості пам'яті в байтах. Ця функція повертає в програму адресу початку вільної ділянки пам'яті, тобто покажчик типу **void***, який потрібно перетворити до типу даних створюваного масиву. Для звільнення пам'яті служить функція **free()**. Прототипи цих функцій описані в заголовному файлі **<stdlib.h>**.

Наприклад:

```
int n;
scanf("%d",&n); //введення розміру масиву (розмір – змінна)
int *ptri=(int*)malloc(n*sizeof(int)); //запит пам'яті
if(ptri==NULL) //перевірка виділення пам'яті
{puts("Error"); return -1;}
...
free(ptri); // звільнення пам'яті
```

У мові С++ для цієї мети існують операції **new** і **delete**. Операція **new** автоматично визначає необхідний обсяг пам'яті в байтах та повертає покажчик заданого типу. Для звільнення пам'яті служить операція **delete[]**.

```
double *ptrd =new double[n]; //запит пам'яті
if(ptrd==NULL) //перевірка виділення пам'яті
{puts("Error"); return -1;}
...
delete[] ptrd; // звільнення пам'яті
```

Звернення до елементів динамічного масиву може здійснюватися через індекси або за допомогою операції розіменування.

Наприклад:

```
for(i=0;i<n;i++)
    ptri[i]=10*i;
for(i=0;i<n;i++)
    printf("%lf ",*(ptrd+i));
```

ЛАБОРАТОРНІ РОБОТИ

Лабораторна робота № 1

Базові типи змінних. Введення і виведення даних

Мета роботи – ознайомитися з основними типами даних, засвоїти основні оператори мови Сі.

Програма 1_1. Програма виводить символ і змінні та визначає їх розміру байтах.

```
#include<stdio.h>
#include<conio.h>
#include <stdlib.h>
int main()
{puts("My first program"); //виведення повідомлення
  char s='a';
  printf("Symbol %c =( %d) 10S =( %o) 8S ",s,s,s);
  printf("(= %x) 16S size char= %d bait\n",s,sizeof(s));
  int a=78;
  printf("a= %d size int= %d bait\n",a,sizeof(a));
  float b=56.23;
  printf("b= %f size float= %d bait\n",b,sizeof(b));
  system("pause"); return 0;}
```

Завдання до програми 1_1. Доповніть програму: виведіть на екран своє прізвище, виведіть замість букви 'a' першу букву Вашого імені з кодами; визначте нові змінні типу **short**, **long**, **double** з будь-якими значеннями та виведіть на екран їх значення та розмір в байтах.

Програма 1_2. Програма виконує прості арифметичні дії.

```
#include<stdio.h>
#include <stdlib.h>
int main()
{int a1=9, a2=2, b1, b2, b3;
  a2+=a1; // a2=2+9=11
  b1=a2/4+a2%4; // b1=11/4+11%4=2+3=7
  b2=b1>b2&&b1>a2; // b3= (7>3) та (7>11)? ні 0
  printf("b2= %d\n",b2);
  b3=-b2; // b2=b2-1=0-1=-1 b3=b2=-1
  printf("a2= %d b1= %d b2= %d b3= %d\n",a2,b1,b2,b3);
  float f1=3.6, f2=0.5, r1, r2;
  r1=f1/f2; // r1=3.6/0.5=7.2
  r2=(f1<f2)?f1+2:--f2; // r2=(3.6<0.5)? ні (0.5-1)= -0.5
  printf("r1= %5.2f r2= %5.2f\n",r1,r2);
  system("pause"); return 0;}
```

Завдання до програми 1_2. Розробіть програму, що обчислює задані вирази. Результати поясніть за допомогою коментарів.

```
a1=3; a2=8; a2-=a1; b1=a1+2*a22; b2=!a1; b3=b2++;  
f1=1.5; f2=1.5; r1=f12+2*f2; r2=(f1!=f2)?f1-0.3:f2+0.4;
```

Програма 1_3. Програма за введеними з клавіатури значеннями радіусу та висоти обчислює та виводить на екран об'єм циліндра.

```
#include <stdio.h>  
#include<stdlib.h>  
int main()  
{float h,r,V;  
  puts("Input r, h");  
  scanf("%f %f", &r,&h);  
  V=3.141592f*r*r*h;  
  printf("r=%6.2f, h=%6.2f V=%8.3f\n",r,h,V);  
  system("pause");  
  return 0;}
```

Завдання до програми 1_3. Розробіть програму, в якій вводяться з клавіатури змінні **x,y** (типу **float**), обчислюється вираз $res = (6x^2 - y) / 2.6$ та результат виводиться на екран.

Програма 1_4. Програма за введенням з клавіатури значенням кількості дюймів виводить на екран відповідне значення сантиметрів. Значення цілої змінної **d** вводиться за допомогою функції **cin>>**, а результат виводиться на екран за допомогою функції **cout<<**.

```
#include<stdlib.h>  
#include <iostream>  
using namespace std;  
int main()  
{float d, D;  
  cout<<"Input the number of inches\n";//введіть кіл-ть дюймів  
  cin>>d;  
  D=d*2.54f; cout<<d<<" inches = "<<D<<" cm\n\n";  
  system("pause");  
  return 0;}
```

Завдання до програми 1_4. Розробіть програму, в якій обчислюється об'єм куба $V=a^3$. Значення змінної **a** введіть за допомогою функції **cin>>**, а результат виведіть за допомогою функції **cout<<**.

Лабораторна робота № 2

Оператори вибору. Оператори циклу

Мета роботи – ознайомитися з операторами вибору: з умовними операторами (**if...else**) та операторами-перемикачами (**switch**), ознайомитися з організацією циклічного обчислювального процесу з застосуванням операторів **while()**, **do...while()**, **for()**, навчитися програмувати найпростіші алгоритми, які розгалужуються.

Програма 2_1. Програма визначає, чи є введене число парним.

```
#include<stdio.h>
#include<stdlib.h>
#include <iostream>
int main()
{int n;
  puts("Input number:");
  scanf("%d",&n);
  if(n%2==0) printf("number %d even \n", n);
  else printf("number %d odd\n\n", n);
  system("pause"); return 0;}
```

Завдання до програми 2_1. Розробіть програму, в якій з клавіатури вводиться ціле число та перевіряється, воно більше 100. На екран виводяться відповідні повідомлення.

Програма 2_2. Програма визначає, чи є введене число додатним та більшим чи меншим заданого числа.

```
#include<stdio.h>
#include<stdlib.h>
#include <iostream>
int main()
{float x;
  puts("Input value:");
  scanf("%f",&x);
  if(x>0) if(x>100) printf("Value %.2f >0 and > 100\n",x);
  else printf("Value %.2f >0 and <= 100\n",x);
  else if(x<-100) printf("Value %.2f <=0 and <-100\n",x);
  else printf("Value %.2f <=0 and >=-100\n",x);
  system("pause"); return 0;}
```

Завдання до програми 2_2. Розробіть програму з вкладеним умовним оператором, в якій з клавіатури вводиться ціле число. Для парного числа перевіряється воно більше 25 чи ні, для непарного числа перевіряється воно додатне чи ні. На екран виводяться чотири відповідних повідомлення.

Програма 2_3. Програма в залежності від введеного символу виводить на екран назву фрукти за допомогою оператора `switch()` із застосуванням оператору `break`.

```
#include <stdio.h>
#include<stdlib.h>
#include <iostream>
int main()
{char s;
 puts("Input sing (a, k, m, o):");
 s=getchar(); fflush(stdin);
 switch(s)
 {case 'a': puts("apple"); break;
 case 'k': puts("kiwi"); break;
 case 'm': puts("mango"); break;
 case 'o': puts("orange"); break;
 default: puts("Error! sing (a, k, m, o)");
 }
 system("pause");
 return 0;}
```

Завдання до програми 2_3. Розробіть програму з використанням оператора `switch()`, яка згідно з введеною з клавіатури першої букви виводить на екран назви 5 міст України або повідомлення про помилку, якщо введена буква не співпадає з заданими.

Програма 2_4. Програма обчислює суму ряду цілих парних чисел, доки сума не стане більше 20, використовуючи оператор `while`.

```
#include <stdio.h>
#include <iostream>
int main()
{int a=0, sum=0, n=0;
 while(sum<20)
 {n+=1;
 a+=2;
 sum+=a;
 printf("n=%d a=%d sum=%d\n",n,a,sum);
 }
 printf("Loops n=%d\n", n);
 system("pause"); return 0;}
```

Завдання до програми 2_4. Розробіть програму, що обчислює суму ряду цілих чисел, кратних 5, доки останнє число не стане більше 40. На екран виводяться результат обчислення, останнє число та кількість циклів.

Програма 2_5. Програма обчислює квадрати послідовності чисел.

```
#include <stdio.h>
#include <stdlib.h>
#include <iostream>
int main()
{int i, b, a=0,kol;
 puts("Input number of cycles ");
 scanf("%d",&kol); // Введення кількості циклів
 for(i=0; i<kol; i++)
     {if(a>500){puts("a>500"); // Завершення циклу при a>500
      break;}

      a+=1;
      b=a*a;
      printf("a=%3d b=%d\n",a,b);
    }
 system("pause");
 return 0;}
```

Завдання до програми 2_5. Розробіть програму, в якій загальна кількість чисел та числа вводяться з клавіатури. Потім обчислюється сума та кількість тільки цілих додатних непарних чисел та результат виводиться на екран (оператор **for**). При введенні числа **-11** цикл завершується (оператор **break**).

Програма 2_6. Програма обчислює та виводить на екран таблицю множення, використовуючи вкладений оператор циклу.

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <iostream>
int main()
{int i,j;
 for(i=1; i<=10; i++)
     {for(j=1; j<=10; j++)
      printf("%d * %d = %2d\n",i,j,i*j);
      printf("\n"); getch();
    }
 system("pause"); return 0;}
```

Завдання до програми 2_6. Розробіть програму з вкладеним оператором циклу, що обчислює та виводить на екран значення **a**, **b** та функції $F=a^2+b^2$. Діапазон зміни **a** від 4 до 20 з кроком 2; діапазон зміни **b** від 3 до 21 з кроком 3.

Лабораторна робота № 3

Масиви і покажчики

Мета роботи – ознайомитися з формами визначення одновимірних масивів у мові Сі, із значенням покажчиків при роботі з масивами, навчитися формувати числові і символьні масиви.

Програма 3_1. У програмі визначається та заповнюється випадковими значеннями одновимірний масив. Розглядається розміщення масиву в пам'яті та визначається його розмір. Визначається мінімальний за значенням елемент масиву.

```
#include<iostream>
#include<time.h> //для запуску генератора випадкових чисел
#include<stdlib.h> //для підключення генератора випадкових чисел
#include <stdio.h>
int main()
{int arr[10], min, i;
  srand( (unsigned) time (NULL) ); // запуск генератора випадк. чисел
  for(i=0; i<10; i++)
    {arr[i]=rand()%101-50; // значення від -50 до +50
    printf("arr[%d]=%3d  %p\n",i,arr[i],&arr[i]);    }
  printf("Size massiv arr: %d bait\n",sizeof(arr));
  min=arr[0];
  for(i=0; i<10; i++)
    if(arr[i]<min) min=arr[i];
  printf("min = %d\n", min);
  system("pause");
  return 0;}
```

Завдання до програми 3_1. Розробіть програму, в якій з клавіатури вводиться масив з 7-и цілих чисел та обчислюється середнє арифметичне усіх елементів. На екран виводяться значення елементів масиву і результати обчислень.

Програма 3_2 В програмі визначається і ініціалізується масив з 10-и цілих чисел та покажчик на цей масив. Потім визначається значення максимального елемента масиву, використовуючи метод доступу через розіменування покажчика.

```
#include<stdio.h>
#include<stdlib.h>
#include<iostream>
int main()
{int arr[10]={5,3,2,-4,6,7,110,-17,0,13}, max, i=0;
```

```

int *px= arr; // визначення покажчика типу int на масив
max=*px; //привласнення max значення нульового елемента масиву
while(i<10)
    {if(*(px+i)>max) max=*(px+i);
      i++; }
printf("max=%d\n", max);
system("pause"); return 0;}

```

Завдання до програми 3_2. Розробіть програму в якій визначається масив з 8-и цілих чисел і покажчик на цей масив. Перший елемент масиву дорівнює 500, всі інші елементи вводяться із клавіатури. Потім, використовуючи метод доступу до елементів масиву через розіменування покажчика, обчислюється мінімальний елемент масиву, що кратний 5. На екран виводяться значення елементів масиву і результати обчислень.

Програма 3_3. В програмі визначається і ініціалізується двовимірний масив та обчислюються суми значень елементів двовимірного масиву в рядках та стовпцях.

```

#include <stdio.h>
#include<stdlib.h>
#include<iostream>
int arr[4][3]={ {2,3,4},{7},{1,9},{8,4,10}},s1[4],s2[3];
int main()
{int i,j;
 puts("Array arr[4][3] ");
 for(i=0;i<4;i++)
    {for(j=0;j<3;j++)
        printf("%5d",arr[i][j]);
      printf("\n");}
 printf("\n Sum elements strings\n");
 for(i=0;i<4;i++)
    {for(j=0;j<3;j++)
        s1[i]+=arr[i][j];
      printf("s1[%d]=%d\n",i,s1[i]); }
 printf("\n Sum elements columns\n");
 for(j=0;j<3;j++)
    {for(i=0;i<4;i++)
        s2[j]+=arr[i][j];
      printf("s2[%d]=%d\n",j,s2[j]); }
 system("pause");
 return 0;}

```

Завдання до програми 3_3. Розробіть програму, в якій визначається двовимірний масив цілих чисел розміром масиву 5 на 4. Масив заповнюється випадковими значеннями у діапазоні від -100 до $+100$ і виводиться на екран. Потім обчислюється суми непарних елементів в стовбцях масиву. Результат виводиться на екран.

Програма 3_4. В програмі створюється динамічний масив для змінних типу `int`. Розмір масиву та значення елементів масиву вводяться з клавіатури. По закінченні роботи з масивом пам'ять звільняється.

```
#include <stdio.h>
#include<stdlib.h>
#include<iostream>
int main()
{int n,i;
  int *ptr; //визначення покажчика на масив чисел типу int
  puts("Input size array");
  scanf("%d",&n);
  //запит пам'яті
  ptr=new int[n]; //запит пам'яті
  // ptr=(int*)malloc(n*sizeof(int)); //запит пам'яті
  if(ptr==NULL) //перевірка виділення пам'яті
  {puts("Error"); return -1;}
  puts("\nInput array");
  for(i=0;i<n;i++)
    scanf("%d",&ptr[i]);
  puts("\nArray");
  for(i=0;i<n;i++)
    printf("%d ",*(ptr+i));
  delete [] ptr; //звільнення пам'яті
  // free(ptr); //звільнення пам'яті
  system("pause");
  return 0;}
```

Завдання до програми 3_4. Розробіть програму, в якій створюється динамічний масив для змінних типу `float` з використанням функції `malloc`. Розмір масиву вводиться з клавіатури. Масив заповнюється випадковими числами у діапазоні від -150.0 до $+150.0$ та виведіть його на екран. У кінці звільніть пам'ять.

ВАРІАНТИ ЗАВДАНЬ ДО КОНТРОЛЬНОЇ РОБОТИ

Завдання № 1

Базові типи змінних. Введення та виведення даних

Варіант 1_1

1. Розробіть програму, в якій з клавіатури вводяться змінні **x, z** (типу **float**), обчислює вираз **res=(4*x-z)²/5.1**. Значення змінних введіть за допомогою функції **scanf()**, а результат виведіть за допомогою функції **printf()**.
2. Розробіть програму, в якій обчислюється об'єм конуса **v=πR²h/3**. Значення змінних **R, h** введіть за допомогою функції **cin>>**, а результат виведіть за допомогою функції **cout<<**.

Варіант 1_2

1. Розробіть програму, в якій з клавіатури вводяться змінні **a, b, c** (типу **double**), обчислюється вираз **res=a+2.3*b-c²**. Значення змінних введіть за допомогою функції **scanf()**, а результат виведіть за допомогою функції **printf()**.
2. Розробіть програму, в якій обчислюється площа квадрата **S=a²**. Значення змінної **a** введіть за допомогою функції **cin>>**, а результат виведіть за допомогою функції **cout<<**.

Варіант 1_3

1. Розробіть програму, в якій з клавіатури вводяться змінні **k, g** (типу **double**), обчислюється вираз **res=k²+(g-k)/3.6**. Значення змінних введіть за допомогою функції **scanf()**, а результат виведіть за допомогою функції **printf()**.
2. Розробіть програму, в якій обчислюється довжина кола **L=2πR**. Значення змінної **R** введіть за допомогою функції **cin>>**, а результат виведіть за допомогою функції **cout<<**.

Варіант 1_4

1. Розробіть програму, в якій з клавіатури вводяться змінні **v, s, f** (типу **float**), обчислюється вираз **res=s³/v-f*v**. Значення змінних введіть за допомогою функції **scanf()**, а результат виведіть за допомогою функції **printf()**.
2. Розробіть програму, в якій обчислюється площу прямокутника **S=a*b**. Значення змінних **a, b** введіть за допомогою функції **cin>>**, а результат виведіть за допомогою функції **cout<<**.

Варіант 1_5

1. Розробіть програму, в якій з клавіатури вводяться змінні **x, z, h** (типу **double**), обчислюється вираз **res=x³+z/h²**. Значення змінних введіть за допомогою функції **scanf()**, а результат виведіть за допомогою функції **printf()**
2. Розробіть програму, в якій обчислюється площу круга **S=πR²**. Значення змінної **R** введіть за допомогою функції **cin>>**, а результат виведіть за допомогою функції **cout<<**.

Варіант 1_6

1. Розробіть програму, в якій з клавіатури вводяться змінні **x, y, z** (типу **double**), обчислюється вираз **res=xcos(y)+6z**. Значення змінних введіть за допомогою функції **scanf()**, а результат виведіть за допомогою функції **printf()**
2. Розробіть програму, в якій обчислюється периметр квадрата **L=4a**. Значення змінної **a** введіть за допомогою функції **cin>>**, а результат виведіть за допомогою функції **cout<<**.

Варіант 1_7

1. Розробіть програму, в якій з клавіатури вводяться змінні **x, y** (типу **double**), обчислюється вираз **res=(x-5)²+sin(y)**. Значення змінних введіть за допомогою функції **scanf()**, а результат виведіть за допомогою функції **printf()**
2. Розробіть програму, в якій обчислюється периметр прямокутника **L=2(a+b)**. Значення змінних **a, b** введіть за допомогою функції **cin>>**, а результат виведіть за допомогою функції **cout<<**.

Варіант 1_8

1. Розробіть програму, в якій з клавіатури вводяться змінні **v, s, f** (типу **float**), обчислюється вираз **res=s³/v-f*v**. Значення змінних введіть за допомогою функції **scanf()**, а результат виведіть за допомогою функції **printf()**
2. Розробіть програму, в якій обчислюється площу трикутника **S=ah/2**. Значення змінних **a, b** введіть за допомогою функції **cin>>**, а результат виведіть за допомогою функції **cout<<**.

Варіант 1_9

1. Розробіть програму, в якій з клавіатури вводяться змінні **x, y** (типу **double**), обчислюється вираз **res=(3x)²+cos(y)**. Значення змінних

введіть за допомогою функції **scanf()**, а результат виведіть за допомогою функції **printf()**

2. Розробіть програму, в якій обчислюється периметр трикутника $L=(a+b+c)$. Значення змінних **a**, **b**, **c** введіть за допомогою функції **cin>>**, а результат виведіть за допомогою функції **cout<<**.

Варіант 1_10

1. Розробіть програму, в якій з клавіатури вводяться змінні **k, m, n** (типу **float**), обчислюється вираз $res=k^2/m+n$. Значення змінних введіть за допомогою функції **scanf()**, а результат виведіть за допомогою функції **printf()**

2. Розробіть програму в якій обчислюється об'єм кулі $V=4/3*\pi R^3$. Значення змінної **R** введіть за допомогою функції **cin>>**, а результат виведіть за допомогою функції **cout<<**.

Завдання № 2

Оператори вибору. Оператори циклу

Варіант 2_1

1. Розробіть програму, в якій з клавіатури вводиться ціле число, перевіряється, воно більше **100** чи ні, та на екран виводяться відповідні повідомлення.

2. Розробіть програму з вкладеним умовним оператором, в якій з клавіатури вводяться два цілих числа. Для більшого з них числа перевіряється, чи знаходиться воно в діапазоні від **5** до **30**. На екран виводяться чотири відповідних повідомлення.

3. Розробіть програму з використанням оператора **switch()**, в якій згідно з введеною з клавіатури першої букви виводиться на екран назви **5** річок Європи або повідомлення про помилку, якщо введена буква не співпадає з заданими.

4. Розробіть програму, в якій обчислюється сума ряду цілих чисел, кратних **4**, доки останнє число не стане більше **40**. На екран виводяться результат обчислення, останнє число та кількість циклів.

5. Розробіть програму з вкладеним оператором циклу, в якій обчислюються та виводяться на екран значення **a**, **b** і функції $F=a^2+b^2$. Діапазон зміни **a** від **4** до **20** з кроком **2**; діапазон зміни **b** від **3** до **21** з кроком **3**.

Варіант 2_2

1. Розробіть програму, в якій з клавіатури вводиться ціле число, перевіряється, воно менше 20 чи ні, та на екран виводяться відповідні повідомлення.
2. Розробіть програму з вкладеним умовним оператором, в якій з клавіатури вводиться ціле число. Для парного числа перевіряється воно додатне, чи ні, для непарного числа – більше 25 чи ні. На екран виводяться чотири відповідних повідомлення.
3. Розробіть програму з використанням оператора **switch()**, в якій згідно з введеною з клавіатури першої букви виводиться на екран назви 7 країн Європи або повідомлення про помилку, якщо введена буква не співпадає з заданими.
4. Розробіть програму, в якій обчислюється сума ряду цілих чисел, що вводяться з клавіатури, доки сума не стане більше 100. На екран виводяться результат обчислення, останнє число та кількість циклів.
5. Розробіть програму з вкладеним оператором циклу, в якій обчислюються і виводяться на екран значення змінних **a**, **b** та функції $F=a^2-4b$. Діапазон зміни **a** від 2 до 10 з кроком 1; діапазон зміни **b** від 5 до 20 з кроком 5.

Варіант 2_3

1. Розробіть програму, в якій з клавіатури вводиться ціле число, перевіряється, воно закінчується на 3 чи ні. На екран виводяться відповідні повідомлення.
2. Розробіть програму з вкладеним умовним оператором, в якій з клавіатури вводиться ціле число. Якщо воно більше 50, то перевіряється, чи кратне воно 4. Інакше перевіряється, чи кратне воно 3. На екран виводяться чотири відповідних повідомлення.
3. Розробіть програму з використанням оператора **switch()**, в якій згідно з введеною з клавіатури першої букви виводиться на екран назви 6 квітів або повідомлення про помилку, якщо введена буква не співпадає з заданими.
4. Розробіть програму, в якій обчислюється сума ряду цілих чисел, кратних 5, доки останнє число не стане більше 50. На екран виводяться результат обчислення, останнє число та кількість циклів.
5. Розробіть програму з вкладеним оператором циклу, в якій обчислюються і виводяться на екран значення змінних **a**, **b** та функції $F=a^2-b^2$. Діапазон зміни **a** від 0 до 40 з кроком 4; діапазон зміни **b** від 6 до 20 з кроком 2.

Варіант 2_4

1. Розробіть програму, в якій з клавіатури вводиться ціле число, перевіряється, воно закінчується на 0 чи ні, та на екран виводяться відповідні повідомлення.
2. Розробіть програму з вкладеним умовним оператором, в якій з клавіатури вводяться два цілих числа. Для меншого з них числа перевіряється, чи є воно непарним. На екран виводяться чотири відповідних повідомлення.
3. Розробіть програму з використанням оператора **switch()**, в якій згідно з введеною з клавіатури першої букви виводиться на екран назви 6 дерев або повідомлення про помилку, якщо введена буква не співпадає з заданими.
4. Розробіть програму, в якій обчислюється сума ряду цілих чисел, що вводяться з клавіатури, доки сума не стане більше 50. На екран виводяться результат обчислення, останнє число та кількість циклів.
5. Розробіть програму з вкладеним оператором циклу, в якій обчислюються та виводяться на екран значення змінних **a**, **b** і функції $F=2a^2+b$. Діапазон зміни **a** від 3 до 30 з кроком 3; діапазон зміни **b** від 8 до 20 з кроком 4.

Варіант 2_5

1. Розробіть програму, в якій з клавіатури вводиться ціле число, перевіряється, воно ділиться на 5 чи ні, та на екран виводяться відповідні повідомлення.
2. Розробіть програму з вкладеним умовним оператором, в якій з клавіатури вводиться ціле число. Для додатного числа перевіряється, чи кратне 5. Інакше перевіряється, чи менше воно за -100. На екран виводяться чотири відповідних повідомлення.
3. Розробіть програму з використанням оператора **switch()**, в якій згідно з введеною з клавіатури першої букви виводиться на екран назви 5 столиць Європи або повідомлення про помилку, якщо введена буква не співпадає з заданими.
4. Розробіть програму, в якій обчислюється сума ряду цілих чисел, суму ряду цілих чисел, кратних 5. Кількість циклів вводиться з клавіатури. На екран виводяться результат обчислення, останнє число та кількість циклів.

5. Розробіть програму з вкладеним оператором циклу, в якій обчислюються та виводяться на екран значення змінних **a**, **b** і функції $F=5a+b^2$. Діапазон зміни **a** від 5 до 50 з кроком 5; діапазон зміни **b** від 4 до 12 з кроком 2.

Варіант 2_6

1. Розробіть програму, в якій з клавіатури вводиться ціле число, перевіряється, воно більше 400 чи ні, та на екран виводяться відповідні повідомлення.
2. Розробіть програму з вкладеним умовним оператором, в якій з клавіатури вводяться два цілих числа. Для меншого з них числа перевіряється, чи є воно від'ємним. На екран виводяться чотири відповідних повідомлення.
3. Розробіть програму з використанням оператору **switch()**, в якій згідно з введеною з клавіатури першої букви виводиться на екран назви 6 країн Америки або повідомлення про помилку, якщо введена буква не співпадає з заданими.
4. Розробіть програму, в якій обчислюється сума ряду цілих чисел, що вводяться з клавіатури, доки сума не стане більше 80. На екран виводяться результат обчислення, останнє число та кількість циклів.
5. Розробіть програму з вкладеним оператором циклу, в якій обчислюються та виводяться на екран значення змінних **a**, **b** і функції $F=a^3-b$. Діапазон зміни **a** від 2 до 20 з кроком 4; діапазон зміни **b** від 5 до 40 з кроком 5.

Варіант 2_7

1. Розробіть програму, в якій з клавіатури вводиться ціле число, перевіряється, воно менше 50 чи ні, та на екран виводяться відповідні повідомлення.
2. Розробіть програму з вкладеним умовним оператором, в якій з клавіатури вводиться ціле число. Для парного числа перевіряється, чи воно більше 2, для непарного числа перевіряється чи менше воно за 10. На екран виводяться чотири відповідних повідомлення.
3. Розробіть програму з використанням оператору **switch()**, в якій згідно з введеною з клавіатури першої букви виводиться на екран назви 5 птахів або повідомлення про помилку, якщо введена буква не співпадає з заданими.

4. Розробіть програму, в якій обчислюється сума ряду цілих чисел, кратних 6, доки останнє число не стане більше 60. На екран виводяться результат обчислення, останнє число та кількість циклів.
5. Розробіть програму з вкладеним оператором циклу, в якій обчислюються та виводяться на екран значення змінних **a**, **b** і функції $F=3a-b^3$. Діапазон зміни **a** від 0 до 30 з кроком 6; діапазон зміни **b** від 4 до 12 з кроком 2.

Варіант 2_8

1. Розробіть програму, в якій з клавіатури вводиться ціле число, перевіряється, воно закінчується на 5 чи ні, та на екран виводяться відповідні повідомлення.
2. Розробіть програму з вкладеним умовним оператором, в якій з клавіатури вводяться два цілих числа. Для меншого з них числа перевіряється, чи знаходиться воно в діапазоні від -5 до 3. На екран виводяться чотири відповідних повідомлення.
3. Розробіть програму з використанням оператора **switch()**, в якій згідно з введеною з клавіатури першої букви виводиться на екран назви 5 країн Азії або повідомлення про помилку, якщо введена буква не співпадає з заданими.
4. Розробіть програму, в якій обчислюється сума ряду цілих чисел, кратних 3. Кількість циклів вводиться з клавіатури. На екран виводяться результат обчислення, останнє число та кількість циклів.
5. Розробіть програму з вкладеним оператором циклу, в якій обчислюються та виводяться на екран значення змінних **a**, **b** і функції $F=a^2+2b$. Діапазон зміни **a** від 4 до 40 з кроком 4; діапазон зміни **b** від 6 до 36 з кроком 6.

Варіант 2_9

1. Розробіть програму, в якій з клавіатури вводиться ціле число, перевіряється, воно ділиться на 3 чи ні, та на екран виводяться відповідні повідомлення.
2. Розробіть програму з вкладеним умовним оператором, в якій з клавіатури вводиться ціле число. Для числа, кратного 5, перевіряється чи кратне воно 10. Інакше перевіряється, чи воно парне. На екран виводяться чотири відповідних повідомлення.
3. Розробіть програму з використанням оператора **switch()**, в якій згідно з введеною з клавіатури першої букви виводиться на екран назви 5 річок Азії або повідомлення про помилку.

4. Розробіть програму, в якій обчислюється сума ряду цілих чисел, кратних 7, доки останнє число не стане більше 70. На екран виводяться результат обчислення, останнє число та кількість циклів.
5. Розробіть програму з вкладеним оператором циклу, в якій обчислюються та виводяться на екран значення змінних **a**, **b** і функції $F=5a-b^3$. Діапазон зміни **a** від 2 до 12 з кроком 2; діапазон зміни **b** від 0 до 30 з кроком 3.

Варіант 2_10

1. Розробіть програму, в якій з клавіатури вводиться ціле число, перевіряється, воно більше 0 чи ні, та на екран виводяться відповідні повідомлення.
2. Розробіть програму з вкладеним умовним оператором, в якій з клавіатури вводиться ціле число. Для від'ємного числа перевіряється, чи менше воно -10. Для додатного числа перевіряється, чи більше 10. На екран виводяться чотири відповідних повідомлення.
3. Розробіть програму з використанням оператору **switch()**, в якій згідно з введеною з клавіатури першої букви виводиться на екран назви 6 овочів або повідомлення про помилку, якщо введена буква не співпадає з заданими.
4. Розробіть програму, в якій обчислюється сума ряду цілих чисел, що вводяться з клавіатури, доки сума не стане більше 40. На екран виводяться результат обчислення, останнє число та кількість циклів.
5. Розробіть програму з вкладеним оператором циклу, в якій обчислюються та виводяться на екран значення змінних **a**, **b** та функції $F=a^2-5b$. Діапазон зміни **a** від 5 до 50 з кроком 5; діапазон зміни **b** від 4 до 10 з кроком 2.

Завдання № 3

Масиви і покажчики

Варіант 3_1

1. Розробіть програму, в якій з клавіатури вводиться масив з 6-и цілих чисел та обчислюється середнє арифметичне тільки ненульових елементів. На екран виводяться значення елементів масиву і результати обчислень.
2. Розробіть програму в якій визначається масив з 6-и цілих чисел та покажчик на цей масив. Перший елемент масиву дорівнює 1000, всі інші елементи вводяться із клавіатури. Потім, використовуючи метод

доступу до елементів масиву через розіменування покажчика, обчислюється номер мінімального додатного елемента. На екран виводяться значення елементів масиву і результати обчислень.

3. Розробіть програму, в якій визначається двовимірний масив цілих чисел розміром масиву 5 на 4. Масив заповнюється випадковими значеннями у діапазоні від -100 до $+100$ і виводиться на екран. Потім обчислюється суми додатних елементів в стовбцях масиву. Результат виводиться на екран.

4. Розробіть програму, в якій створюється динамічний масив для змінних типу **short** з використанням операції **new**. Розмір масиву вводиться з клавіатури. Масив заповнюється випадковими числами у діапазоні від -1000 до $+500$ та виводиться на екран. У кінці звільняється пам'ять.

Варіант 3_2

1. Розробіть програму, в якій з клавіатури вводиться масив з 8 цілих чисел та обчислюється середнє арифметичне елементів без обліку мінімального елемента масиву. На екран виводяться значення елементів масиву і результати обчислень.

2. Розробіть програму в якій визначається масив з 7-и цілих чисел і покажчик на цей масив. Перший елемент масиву дорівнює -2000 , всі інші елементи вводяться із клавіатури. Потім, використовуючи метод доступу до елементів масиву через розіменування покажчика, обчислюється номер максимального від'ємного елемента масиву. На екран виводяться значення елементів масиву і результати обчислень.

3. Розробіть програму, в якій визначається двовимірний масив цілих чисел розміром масиву 3 на 6. Масив заповнюється випадковими значеннями у діапазоні від -50 до $+50$ і виводиться на екран. Потім обчислюється кількість додатних парних елементів в рядках масиву..

4. Розробіть програму, в якій створюється для змінних типу **double** з використанням операції **new**. Розмір масиву вводиться з клавіатури. Масив заповнюється випадковими числами у діапазоні від 0.00 до 32.00 та виводиться на екран. У кінці звільняється пам'ять.

Варіант 3_3

1. Розробіть програму, в якій з клавіатури вводиться масив з 6-и цілих чисел та обчислюється середнє арифметичне тільки додатних елементів. На екран виводяться значення елементів масиву і результати обчислень.

2. Розробіть програму в якій визначається масив з 8-и цілих чисел і покажчик на цей масив. Перший елемент масиву дорівнює **-100**, всі інші елементи вводяться із клавіатури. Потім, використовуючи метод доступу до елементів масиву через розіменування покажчика, обчислюється максимальний парний елемент масиву. На екран виводяться значення елементів масиву і результати обчислень.
3. Розробіть програму, в якій визначається двовимірний масив цілих чисел розміром масиву **5** на **3**. Масив заповнюється випадковими значеннями у діапазоні від **-20** до **+20** і виводиться на екран. Потім обчислюється кількість непарних елементів в стовбцях масиву. Результат виводиться на екран.
4. Розробіть програму, в якій створюється динамічний масив для змінних типу **char** з використанням функції **malloc**. Розмір масиву вводиться з клавіатури. Масив заповнюється випадковими числами у діапазоні від **60** до **125** та виводиться на екран в десятковому та символічному вигляді. У кінці звільняється пам'ять.

Варіант 3_4

1. Розробіть програму, в якій з клавіатури вводиться масив з 8-и цілих чисел та обчислюється середнє арифметичне тільки непарних елементів. На екран виводяться значення елементів масиву і результати обчислень.
2. Розробіть програму в якій визначається масив з 7-и цілих чисел і покажчик на цей масив. Перший елемент масиву дорівнює **201**, всі інші елементи вводяться із клавіатури. Потім, використовуючи метод доступу до елементів масиву через розіменування покажчика, обчислюється мінімальний непарний елемент масиву. На екран виводяться значення елементів масиву і результати обчислень.
3. Розробіть програму, в якій визначається двовимірний масив цілих чисел розміром масиву **3** на **5**. Масив заповнюється випадковими значеннями у діапазоні від **-40** до **+40** і виводиться на екран. Потім обчислюється суму від'ємних непарних елементів в рядках масиву. Результат виводиться на екран.
4. Розробіть програму, в якій створюється динамічний масив для змінних типу **double** з використанням функції **malloc**. Розмір масиву вводиться з клавіатури. Масив заповнюється випадковими числами у діапазоні від **-15.000** до **15.000** та виводиться на екран. У кінці звільняється пам'ять.

Варіант 3_5

1. Розробіть програму, в якій з клавіатури вводиться масив з 6-и цілих чисел та обчислюється сума квадратів від'ємних елементів. На екран виводяться значення елементів масиву і результати обчислень.
2. Розробіть програму в якій визначається масив з 8-и цілих чисел і покажчик на цей масив. Перший елемент масиву дорівнює **300**, всі інші елементи вводяться із клавіатури. Потім, використовуючи метод доступу до елементів масиву через розіменування покажчика, обчислюється мінімальний елемент масиву, що кратний **3**. На екран виводяться значення елементів масиву і результати обчислень.
3. Розробіть програму, в якій визначається двовимірний масив цілих чисел розміром масиву **4** на **5**. Масив заповнюється випадковими значеннями у діапазоні від **-10** до **+10** і виводиться на екран. Потім обчислюється добуток елементів в діапазоні від **1** до **5** в стовбцях.
4. Розробіть програму, в якій створюється динамічний масив для змінних типу **float** з використанням операції **new**. Розмір масиву вводиться з клавіатури. Масив заповнюється випадковими числами у діапазоні від **-50.5** до **+50.5** та виводиться на екран. У кінці звільняється пам'ять.

Варіант 3_6

1. Розробіть програму, в якій з клавіатури вводиться масив з 7-и цілих чисел та обчислюється середнє арифметичне від'ємних елементів. На екран виводяться значення елементів масиву і результати обчислень.
2. Розробіть програму в якій визначається масив з 8-и цілих чисел і покажчик на цей масив. Перший елемент масиву дорівнює **-101**, всі інші елементи вводяться із клавіатури. Потім, використовуючи метод доступу до елементів масиву через розіменування покажчика, обчислюється максимальний непарний елемент масиву. На екран виводяться значення елементів масиву і результати обчислень.
3. Розробіть програму, в якій визначається двовимірний масив цілих чисел розміром масиву **4** на **5**. Масив заповнюється випадковими значеннями у діапазоні від **-60** до **+60** і виводиться на екран. Потім обчислюється кількість парних елементів в рядках масиву.
4. Розробіть програму, в якій створюється динамічний масив для змінних типу **short** з використанням функції **malloc**. Розмір масиву вводиться з клавіатури. Масив заповнюється випадковими числами у діапазоні від **-200** до **+400** та виводиться на екран. У кінці звільняється пам'ять.

Варіант 3_7

1. Розробіть програму, в якій з клавіатури вводиться масив з 7-и цілих чисел та обчислюється сума квадратів парних елементів. На екран виводяться значення елементів масиву і результати обчислень.
2. Розробіть програму в якій визначається масив з 8-и цілих чисел і покажчик на цей масив. Перший елемент масиву дорівнює 200, всі інші елементи вводяться із клавіатури. Потім, використовуючи метод доступу до елементів масиву через розіменування покажчика, обчислюється мінімальний парний елемент масиву. На екран виводяться значення елементів масиву і результати обчислень.
3. Розробіть програму, в якій визначається двовимірний масив цілих чисел розміром масиву 6 на 3. Масив заповнюється випадковими значеннями у діапазоні від -300 до +300 і виводиться на екран. Потім обчислюється сума додатних парних елементів в стовбцях масиву.
4. Розробіть програму, в якій створюється динамічний масив для змінних типу **double** з використанням операції **new**. Розмір масиву вводиться з клавіатури. Масив заповнюється випадковими числами у діапазоні від -50.00 до 100.00 та виводиться на екран. У кінці звільняється пам'ять.

Варіант 3_8

1. Розробіть програму, в якій з клавіатури вводиться масив з 8-и цілих чисел та обчислюється сума квадратів непарних елементів. На екран виводяться значення елементів масиву і результати обчислень.
2. Розробіть програму в якій визначається масив з 8-и цілих чисел і покажчик на цей масив. Перший елемент масиву дорівнює 5, всі інші елементи вводяться із клавіатури. Потім, використовуючи метод доступу до елементів масиву через розіменування покажчика, обчислюється максимальний додатного елемент масиву. На екран виводяться значення елементів масиву і результати обчислень.
3. Розробіть програму, в якій визначається двовимірний масив цілих чисел розміром масиву 3 на 6. Масив заповнюється випадковими значеннями у діапазоні від -20 до +20 і виводиться на екран. Потім обчислюється добуток елементів в діапазоні від 1 до 6 в рядках.
4. Розробіть програму, в якій створюється динамічний масив для змінних типу **double** з використанням функції **malloc**. Розмір масиву вводиться з клавіатури. Масив заповнюється випадковими числами у діапазоні від -10.000 до 10.000 та виводиться на екран. У кінці звільняється пам'ять.

Варіант 3_9

1. Розробіть програму, в якій з клавіатури вводиться масив з 6-и цілих чисел та середнє арифметичне елементів без обліку максимального елемента масиву. На екран виводяться значення елементів масиву і результати обчислень.
2. Розробіть програму, в якій визначається масив з 8-и цілих чисел і покажчик на цей масив. Перший елемент масиву дорівнює **-1000**, всі інші елементи вводяться із клавіатури. Потім, використовуючи метод доступу до елементів масиву через розіменування покажчика, обчислюється мінімальний від'ємний елемент. На екран виводяться значення елементів масиву і результати обчислень.
3. Розробіть програму, в якій визначається двовимірний масив цілих чисел розміром масиву **5** на **4**. Масив заповнюється випадковими значеннями у діапазоні від **-150** до **+150** і виводиться на екран. Потім обчислюється кількість від'ємних елементів в стовбцях масиву. Результат виводиться на екран.
4. Розробіть програму, в якій створюється динамічний масив для змінних типу **int** з використанням функції **malloc**. Розмір масиву вводиться з клавіатури. Масив заповнюється випадковими числами у діапазоні від **-2000** до **+1000** та виводиться на екран. У кінці звільняється пам'ять.

Варіант 3_10

1. Розробіть програму, в якій з клавіатури вводиться масив з 8-и цілих чисел та обчислюється середнє арифметичне тільки парних елементів. На екран виводяться значення елементів масиву і результати обчислень.
2. Розробіть програму, в якій визначається масив з 8-и цілих чисел і покажчик на цей масив. Перший елемент масиву дорівнює **6**, всі інші елементи вводяться із клавіатури. Потім, використовуючи метод доступу до елементів масиву через розіменування покажчика, обчислюється максимальний елемент масиву, що кратний **3**. На екран виводяться значення елементів масиву і результати обчислень.
3. Розробіть програму, в якій визначається двовимірний масив цілих чисел розміром масиву **4** на **5**. Масив заповнюється випадковими значеннями у діапазоні від **-60** до **+60** і виводиться на екран. Потім обчислюється суму додатних непарних елементів в рядках масиву. Результат виводиться на екран.

4. Розробіть програму, в якій створюється динамічний масив для змінних типу **char** з використанням операції **new**. Розмір масиву вводится з клавіатури. Масив заповнюється випадковими числами у діапазоні від **80** до **120** та виводиться на екран в десятковому та символічному вигляді. У кінці звільняється пам'ять.

ПИТАННЯ ДО КОНТРОЛЬНОЇ РОБОТИ

1. Які типи даних використовуються в мові C++? Скільки байт займає кожен із типів даних у мові Cі? Специфікатори виведення.
2. Який загальний склад аргументів функції **printf()**?
3. Який загальний склад аргументів функції **scanf()**?
4. Призначення оператора **if... else**. Яке значення може прийняти вираз умови в операторі **if (умова)**? Що є істиною, а що хибністю?
5. Призначення оператора **switch()**. Які значення може приймати вираз в операторі **switch(вираз)**?
6. Які оператори циклу Ви знаєте? Чим вони відрізняються?
7. Поясніть призначення операторів **continue**, **break**, **return** та наведіть приклади їх використання в операторах циклу.
8. Що таке масив? Що таке ініціалізація масиву?
9. Що таке глобальний, автоматичний та статичний масиви? Як вони неявно ініціалізуються?
10. Як можна задавати розмір масиву?
11. Дайте визначення рядку.
12. Що таке покажчик? Який розмір покажчика? Що визначає тип покажчика? Що показує різниця покажчиків?
13. Що таке багатовимірний масив? Як розташовуються в пам'яті елементи багатовимірних масивів?
14. Що таке динамічний масив? Чим відрізняється від звичайного?

РЕКОМЕНДОВАНА ЛИТЕРАТУРА

1. Подбельский В.В. Язык Си++: Учеб. пособие. – М.: Финансы и статистика, 2007. – 560 с.
2. Подбельский В.В. Практикум по программированию на языке Си: Учебн. пособие. – М.: Финансы и статистика, 2004. – 576 с.
3. Подбельский В.В., Фомин С.С. Программирование на языке Си: Учебн. пособие. – 2-е изд., – М.: Финансы и статистика, 2009. – 600с.
4. Павловская Т.А. С/С++. Программирование на языке высокого уровня. СПб: Питер, 2009.– 464 с.
5. Фридман А.Л. Основы объектно-ориентированного программирования на языке Си++. –М: Горячая линия. – Телеком. Радио и связь, 1999. –208 с.
6. Уильям Топп, Уильям Форд. Структуры данных в С++. / Пер. с англ.. – М.: ЗАО “Издательство БИНОМ”, 1999. – 816с.: ил.
7. Крупник А. Изучаем Си.- СПб.: Питер.-2001. – 256с.
8. С/С++. Структурное программирование. Практикум/ Т.А. Павловская, Ю.А.Шупак. – СПб.: Питер, 2002. – 240с.
9. Культин Н.Б. С /С++ в задачах и примерах.- СПб.: БХВ: Петербург, 2001. – 288с.
10. Х.М. Дейтел, П.Дж. Дейтел. Как программировать на С++. – 5-е изд. / Пер. с англ. – М.: ООО "Бином-Пресс", 2008.– 1456 с.
11. Шилдт, Герберт. Полный справочник по С++. – 4-е изд. / Пер. с англ. – М.: Вильямс, 2008. – 800 с.