**Завдання 1. Доступ до елементів масиву через покажчики**

Задача 1. Написати програму, яка визначає кількість ненульових елементів одновимірного масиву.

#include <iostream>

#include <conio.h>

#include <ctime>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

srand(time(NULL));

const int n = 35;

int kol = 0;

int \*Arr = new int [n]; // виділення динамічної пам'яті

for(int i = 0; i < n; i++) // заповнення і виведення масиву

{

\*Arr = rand() % 20;

cout<<\*Arr<<" ";

if(\*(Arr++) != 0) // підрахунок ненульових елементів

kol++;

}

cout<<"\nКоличество ненулевыых елементов: "<<kol<<"";

delete Arr; // вивільнення динамічної пам'яті

\_getch();

}

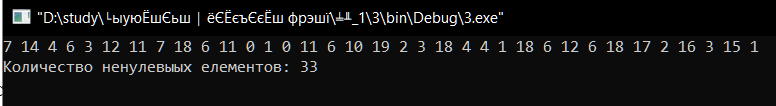


Рис.1 – робота програмного коду(Задача 1)

Задача 2. Написати програму, яка заміняє нулями непарні елементи двовимірного масиву цілих чисел.

#include <iostream>

#include <conio.h>

#include <ctime>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

srand(time(NULL));

int n;

int i, j;

cout<<"Введите количество строк: ";

cin>>n;

// виділення динамічної пам'яті

int \*\*A = new int\*[n];

for(i = 0; i < n; i++)

A[i] = new int[n];

// заповнення і виведення масиву

cout<<"\n Старый массив:\n";

for(i = 0; i < n; i++)

{

for(j = 0; j < n; j++)

{

A[i][j] = rand() % 20;

cout<<A[i][j]<<"\t";

}

cout<<"\n";

}

for(i = 0; i < n; i++)

{

for(j = 0; j < n; j++)

{

if(A[i][j] % 2 != 0) // якщо елемент непарний, то змінюється на нуль

A[i][j] = 0;

}

}

cout<<"\n";

cout<<"\n Новый массив:\n";

for(i = 0; i < n; i++) //виводимо новий масив

{

for(j = 0; j < n; j++)

{

cout<<A[i][j]<<"\t";

}

cout<<"\n";

}

// вивільнення динамічної пам'яті

for(i = 0; i < n; i++)

delete[]A[i];

delete[]A;

\_getch();

}

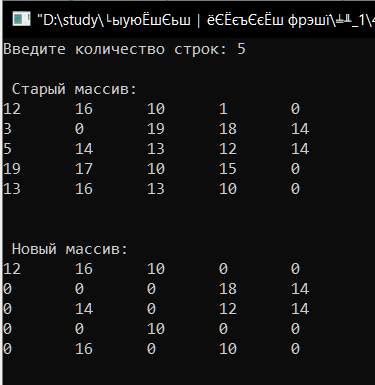


Рис. 2 – робота програмного коду (Задача 2)

Задача 4. Написати програму, яка визначає максимальний елемент двовимірного масиву цілих чисел та його положення у масиві.

#include <iostream>

#include <conio.h>

#include <ctime>

using namespace std;

int main()

{

srand(time(NULL));

setlocale(LC\_ALL,"Russian");

int n,m,i,ki,kj,max;

cout<<"Введите количество строк: ";

cin>>n;

cout<<"Введите количество столбцов: ";

cin>>m;

// виділення динамічної пам'яті

int \*\*Arr = new int\*[n];

for(i = 0; i < n; i++)

Arr[i] = new int[m];

// заповнення і виведення масиву

for(int i = 0; i < n; i++)

{

for(int j = 0; j < m; j++)

{

Arr[i][j] = rand()%35;

cout<<Arr[i][j]<<" ";

}

cout<<"\n";

}

max = Arr[0][0]; // вважатимемо максимальним перший елемент

// визначаємо максимальний елемент та його індекси

for (int i = 0; i < n; i++)

{

for(int j = 0; j < m; j++){

if (Arr[i][j] > max)

{

max = Arr[i][j];

ki = i;

kj = j;

}

}

}

//виведення результату

cout<<"\nMAX = "<<max<<"["<<ki<<"]["<<kj<<"]";

// вивільнення динамічної пам'яті

for(i = 0; i < n; i++)

delete[]Arr[i];

delete[]Arr;

\_getch();

}

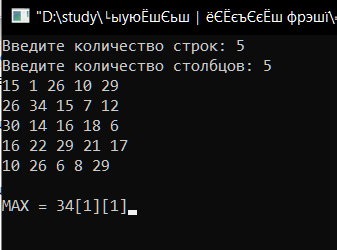


Рис.3 – робота програмного коду (Задача 4)

Задача 5. Написати програму, яка підраховує суму елементів масиву цілих чисел по стовпчикам. Результати збережіть у іншому масиві.

#include <iostream>

#include <conio.h>

#include <ctime>

using namespace std;

int main()

{

srand(time(NULL));

setlocale(LC\_ALL,"Russian");

int n,m,i,j;

int sum[m];

cout<<"Введите количество строк: ";

cin>>n;

cout<<"Введите количество столбцов: ";

cin>>m;

cout<<"\n";

// виділення динамічної пам'яті

int \*\*Arr = new int\*[n];

for(i = 0; i < n; i++)

Arr[i] = new int[m];

// заповнення і виведення масиву

for(int i = 0; i < n; i++)

{

for(int j = 0; j < m; j++)

{

Arr[i][j] = rand()%35;

cout<<Arr[i][j]<<"\t";

}

cout<<"\n";

}

// додаємо елементи по стовпцям

for(j = 0; j < m; j++)

{

sum[j] = 0;

for(i = 0; i < n; i++)

{

sum[j] += Arr[i][j];

}

}

cout<<"\n";

// виводимо суму для кожного стовпця

for(j = 0; j < m; j++)

{

cout<<sum[j]<<"\t";

}

// вивільнення динамічної пам'яті

for(i = 0; i < n; i++)

delete[]Arr[i];

delete[]Arr;

\_getch();

}

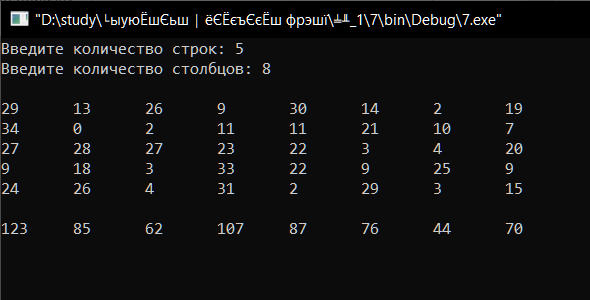


Рис.4 – робота програмного коду (Задача 5)

**Завдання 2. Функції**

Задача 1. Знайти суму елементів головної діагоналі масиву цілих чисел.

#include <iostream>

#include <conio.h>

#include <ctime>

using namespace std;

int Arr(int \*\*a, int n)

{

int sum = 0;

srand(time(NULL));

// заповненя масиву рандомними числами і його виведення на екран

for(int i = 0; i < n; i++)

{

for(int j = 0; j < n; j++)

{

a[i][j] = rand() % 10;

cout<<a[i][j]<<" ";

}

cout<<"\n";

}

for(int i = 0; i < n; i++)

sum += a[i][i]; // додавання чисел головної діагоналі

return sum;

}

int main()

{

srand(time(NULL));

setlocale(LC\_ALL,"Russian");

int n,i,result;

cout<<"Введите количество строк: ";

cin>>n;

// виділення динамічної пам'яті

int \*\*a = new int\*[n];

for(i = 0; i < n; i++)

a[i] = new int[n];

result=Arr(a,n);

cout<<"Сумма = "<<result;

// вивільнення динамічної пам'яті

for(i = 0; i < n; i++)

delete[]a[i];

delete[]a;

\_getch();

}

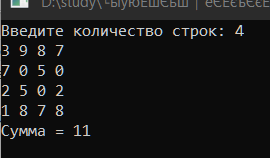


Рис.6 – робота програмного коду (Задача 1)

Задача 2. Написати програму, яка визначає, скільки раз у масиві зустрічається введене число з клавіаутри.

#include <iostream>

#include <conio.h>

#include <ctime>

using namespace std;

int in(int \*Arr, int n, int currentNum)

{

srand(time(NULL));

int kol = 0;

for(int i = 0; i < n; i++) // введення і виведення масиву

{

\*Arr = rand() % 20;

cout<<\*Arr<<" ";

if(\*(Arr++) == currentNum) // підраховуємо потрібне число

kol++;

}

return kol;

}

void out(int c)

{

cout<<"\nЧисло встречается в масиве "<<c<<" раз"; // виводимо результат

return;

}

int main()

{

setlocale(LC\_ALL,"Russian");

int n,result,currentNum;

cout<<"Введите кол-во елементов массива: ";

cin>>n;

cout<<"\nВведите число: ";

cin>>currentNum;

// виділення динамічної пам'яті

int \*Arr = new int[n];

result = in(Arr,n,currentNum);

out(result);

// вивільнення динамічної пам'яті

delete Arr;

\_getch();

}

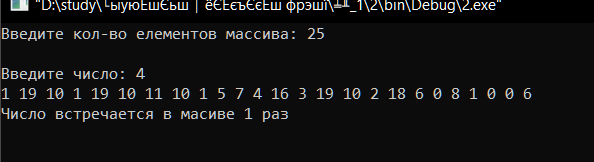


Рис.7 – робота програмного коду (Задача 2)

**Висновки**

На лабораторній роботі №1 ми навчилися працювати з елементами масивів через покажчики та функціями. Створили декілька програм для закріплення знань.