Додаток 1

Міністерство освіти і науки України
Національний технічний університет України “Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи №8 з дисципліни
«Алгоритми та структури даних-1.
Основи алгоритмізації»

«Дослідження алгоритмів пошуку та сортування»

Варіант 29

Виконав студент ІП-15 Рибалка Ілля Сергійович
(шифр, прізвище, ім'я, по батькові)

Перевірив Вєчерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

Київ 2021

Лабораторна робота 8

Дослідження алгоритмів пошуку та сортування

Мета – дослідити алгоритми пошуку та сортування, набути практичних навичок використання цих алгоритмів під час складання програмних специфікацій.

Індивідуальне завдання

Варіант 29

Завдання

Розробити алгоритм та написати програму, яка складається з наступних дій:
1. Опису змінної індексованого типу (двовимірний масив) згідно з варіантом.
2. Ініціювання змінної, що описана в пункті 1 даного завдання.
3. Створення нової змінної індексованого типу (одновимірний масив) та її ініціювання значеннями, що обчислюються згідно з варіантом.

№	Розмірність	Тип даних	Обчислення значень елементів одновимірного масиву
29	5 x 8	Цілий	Із мінімальних значень елементів стовпців двовимірного масиву. Відсортувати методом Шелла за зростанням.

1. Постановка задачі

Створити одновимірний масив і матрицю. Заповнити матрицю довільними числами. Знайти мінімальний елемент кожного стовпця матриці і додати їх в одновимірний масив. Відсортувати знайдений масив методом Шелла за зростанням.

2. Побудова математичної моделі

Змінна	Тип	Ім’я	Призначення
Основна програма			
Число рядків	Ціле	row	Початкові дані
Число стовпців	Ціле	col	Початкові дані
Максимальне число рандомізації	Ціле	rmax	Початкові дані
Мінімальне число рандомізації	Ціле	rmin	Початкові дані
Матриця	Ціле	matrix	Початкові / Проміжні дані
Одновимірний масив	Ціле	res	Проміжні дані / Результат
Заповнення матриці числами	Підпрограма	crmatrix	Початкові дані

Заповнення масиву мінімальними елементами стовпців	Підпрограма	colminlst	Початкові дані
Сортування методом Шелла	Підпрограма	sort	Початкові дані
Вивід масиву	Підпрограма	out	Початкові дані
Розмір масиву	Ціле	len	Проміжні дані
Матриця	Ціле	mtrx	Проміжні дані
Одновимірний масив	Ціле	lst	Проміжні дані

Початковими даними програми є чотири константи *row*=5 і *col*=8, *rmax*=99 і *rmin*=10. Після цього ініціалізується два масиви, одновимірний *res[col]* і двовимірний масив *matrix[row][col]*. Заповнюємо матрицю довільними числами, за рахунок функції *rand* від *rmin* до *rmax*, цю дію виконує підпрограма *crmatrix*.

Заповнювати масив *res* буде підпрограма *colminlst*, що обходить кожен стовпець матриці, і знаходить мінімальне число стовпця копіюючи його в масив.

Підпрограма *sort* виконує сортування масиву *res* за рахунок методу Шелла.

Розв’язання

Програмні специфікації запишемо у псевдокоді та графічній формі у вигляді блок-схеми.
Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо дію виведення матриці за рахунок арифметичних циклів.

Крок 3. Деталізуємо дію заповнення матриці за рахунок підпрограми crmatrix.

Крок 4. Деталізуємо дію заповнення масиву за рахунок підпрограми colminlst.

Крок 5. Деталізуємо дію сортування масиву за рахунок підпрограми sort.

Крок 6. Деталізуємо дію виведення масиву за рахунок підпрограми out.

Псевдокод

Основна програма

крок 1
початок
row = 5
col = 8
rmax=99
rmin=10
crmatrix(matrix, row, col, rmax, rmin)
Вивід matrix
colminlst(matrix, res, row, col, rmax)
Вивід "Знайдений масив = "

```
out(res, col)
sort(res, col)
Вивід "Відсортований масив = "
out(res, col)
кінець
```

крок 2

```
початок
row = 5
col = 8
rmax=99
rmin=10
crmatrix(matrix, row, col, rmax, rmin)
Вивід "Згенерована матриця:"
для i від 0 до row повторити
        Вивід "|"
        для j від 0 до col повторити
                Вивід matrix[i][j]
        все повторити
        Вивід "|"
все повторити
colminlst(matrix, res, row, col, rmax)
Вивід "Знайдений масив = "
out(res, col)
sort(res, col)
Вивід "Відсортований масив = "
out(res, col)
кінець
```

Підпрограми

крок 3

```
crmatrix(mtrx, row, col, rmax, rmin)
для i від 0 до row повторити
        для j від 0 до col повторити
                mtrx[i][j]=rand від rmin до rmax
        все повторити
все повторити
кінець crmatrix
```

крок 4

```
colminlst(mtrx, lst, row, col, rmax)
для i від 0 до row повторити
        min = rmax
        для j від 0 до col повторити
                якщо mtrx[j][i] < min
                        min = mtrx[j][i]
                все якщо
        все повторити
        lst[i] = min
все повторити
кінець colminlst
```

крок 5

```
sort(lst, len)
для i від len/2 до 0 з кроком i/2 повторити
        для j від i до len повторити
                для k від j - i поки k>=0 з кроком k-1 повторити
                        якщо lst[k] > lst[k + i]
                                lst[k + i] += lst[k]
                                lst[k] = lst[k + i] - lst[k]
                                lst[k + i] -= lst[k]
                        все якщо
                все повторити
        все повторити
все повторити
кінець sort
```

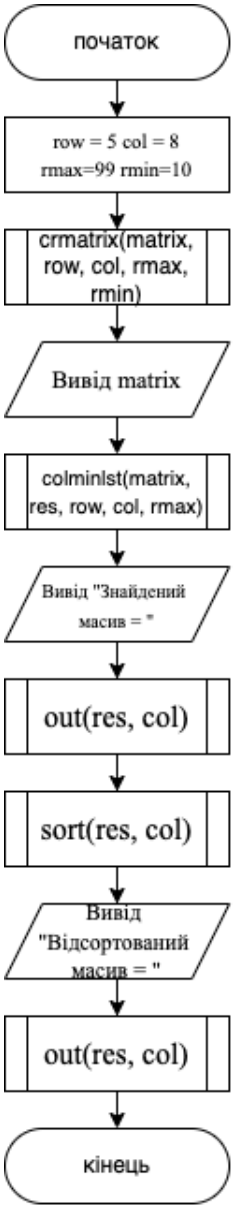
крок 6

```
out(lst, len)
Вивід "["
для i від 0 до len повторити
        Вивід lst[i]
        якщо i<len-1 то
                Вивід ","
        все якщо
все повторити
Вивід "]"
кінець out
```

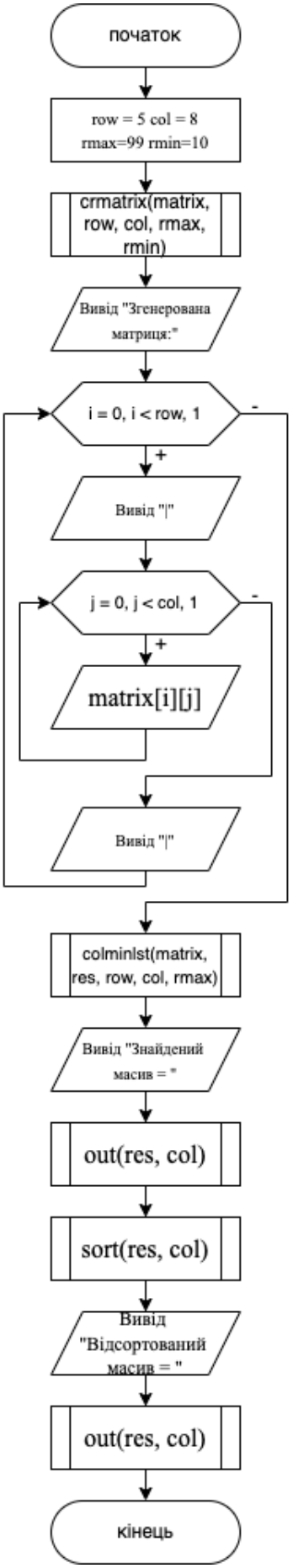
Блок-Схема

Основна програма

крок 1

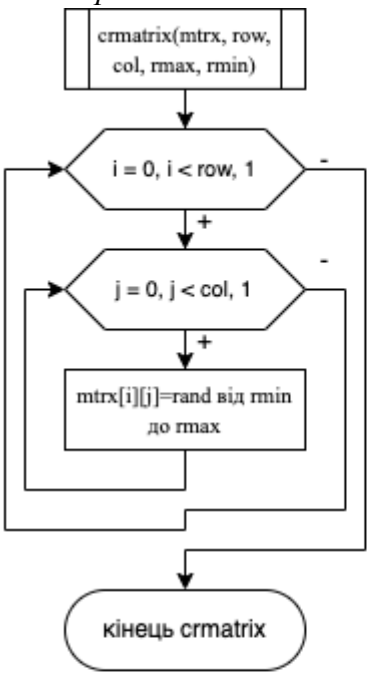


крок 2

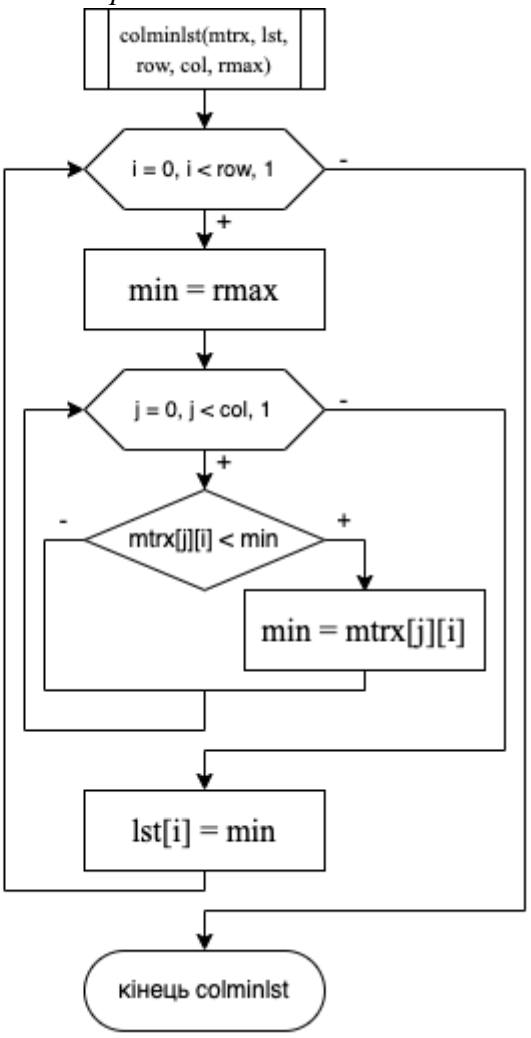


Підпрограми

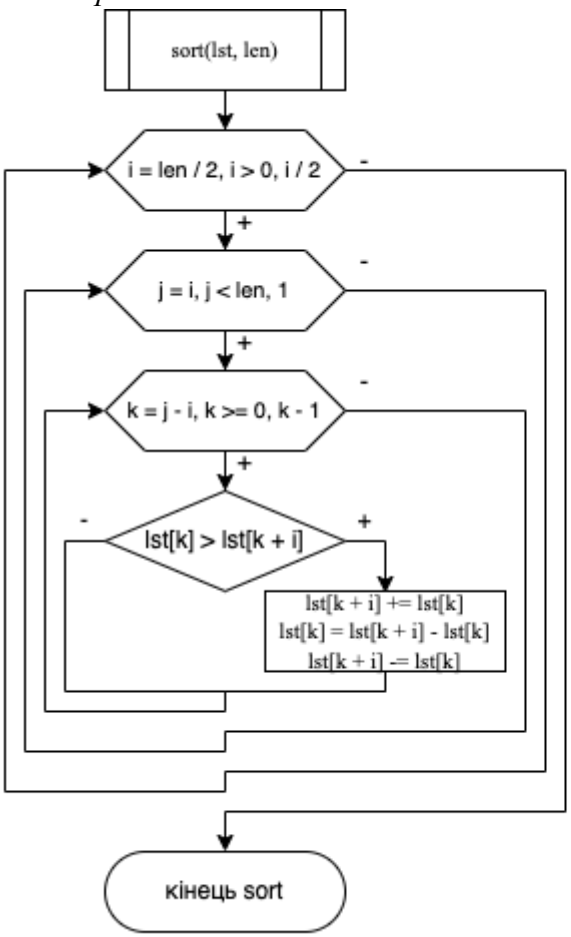
крок 3



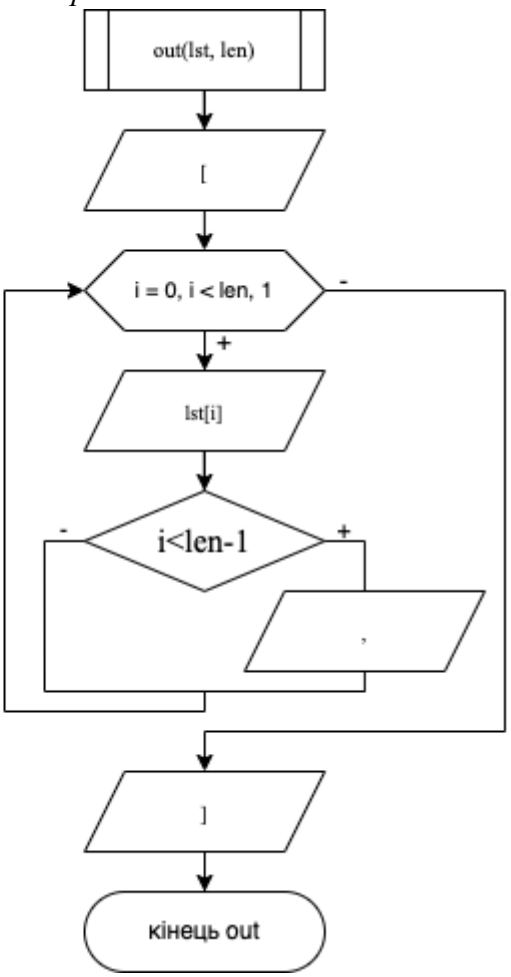
крок 4



крок 5



крок 6



Код

```
#include <iostream>
#include <iomanip>
#include <ctime>

void crmatrix(int **, int, int);
void colminlst(int **, int [], int, int);
void sort(int [], int);
void out(int [], int);

int main()
{
   const int row = 5, col = 8;
   int **matrix = new int *[row];
   int res[col];
   for (int i=0; i<row; i++)
   {
       matrix[i] = new int [col];
   }
   crmatrix(matrix, row, col);
   std::cout << "Згенерована матриця:" << std::endl;
   for (int i=0; i<row; i++)
   {
       std::cout << "|";
       for (int j=0; j<col; j++)
       {
           std::cout << std::setw(3) << mtrx[i][j];
       }
       std::cout << std::setw(2) << "|" << std::endl;
   }
   colminlst(matrix, res, row, col);
   for (int i=0; i<row; i++)
   {
       delete [] matrix[i];
   }
   delete [] matrix;
   std::cout << "Знайдений масив = ";
   out(res, col);
   sort(res, col);
   std::cout << "Відсортований масив = ";
   out(res, col);
   return 0;
}

void crmatrix(int **mtrx, int row, int col)
{
   srand(time(NULL));
```

```
    for (int i=0; i<row; i++)
    {
        for (int j=0; j<col; j++)
        {
            mtrx[i][j]= rand()%89 + 10;
        }
    }
}

void colminlst(int **mtrx, int lst[], int row, int col)
{
    for (int i=0; i<col; i++)
    {
        int min = 100;
        for (int j=0; j<row; j++)
        {
            if (mtrx[j][i] < min)
            {
                min = mtrx[j][i];
            }
        }
        lst[i] = min;
    }
}

void sort(int lst[], int len)
{
    for (int i = len / 2; i > 0; i /= 2)
    {
        for (int j = i; j < len; j++)
        {
            for (int k = j - i; k >= 0; k -= i)
            {
                if (lst[k] > lst[k + i])
                {
                    lst[k + i] += lst[k];
                    lst[k] = lst[k + i] - lst[k];
                    lst[k + i] -= lst[k];
                }
            }
        }
    }
}

void out(int lst[], int len)
{
    std::cout << "[";
    for (int i=0; i<len; i++)
    {
        std::cout << std::setw(3) << lst[i];
        if (i<len-1) std::cout << ",";
    }
    std::cout << std::setw(2) << "]" << std::endl;
}
```

Тестування

```
Згенерована матриця:
| 80 87 57 49 98 19 88 87 |
| 54 64 30 68 90 73 51 47 |
| 37 22 42 88 23 18 16 95 |
| 22 56 98 49 35 45 29 69 |
| 32 94 83 33 60 39 35 11 |
Знайдений масив = [ 22, 22, 30, 33, 23, 18, 16, 11 ]
Відсортований масив = [ 11, 16, 18, 22, 22, 23, 30, 33 ]
```

Висновок

Я дослідив алгоритми пошуку та сортування, набув практичних навичок використання цих алгоритмів під час складання програмних специфікацій. В ході роботи було створено алгоритм для обробки довільної матриці 5x8 і масиву, що створюється з певних елементів матриці.