Звіт

Автор: Чугунов В.Ю

КІТ-119а

Лабораторна робота №9

**ВИКЛЮЧЕННЯ**

**Мета:** Навчитись розробляти програми з реалізацією виключень.

1 ЗАВДАННЯ ДО РОБОТИ

**Загальне завдання.** У файлі розміщена інформація про N масивів.  
 В першому рядку міститься інформація про кількість масивів, у кожній наступній – інформація про кількість елементів в кожному масиві та власне дані масиву.  
 Дії, що має виконувати програма:  
 - введення з клавіатури назви вхідного файлу з даними;  
 - читання даних з файлу;  
 - виконання індивідуального завдання;  
 - введення з клавіатури імені вихідного файлу;  
 - запис результату операції у файл;  
 - доступ до елемента за індексом.

**Індивідуальне завдання.** Знайти у масиві кількість елементів з максимальним значенням. Результат операції – масив з кількостей максимальних елементів кожного з вхідних масивів.

2 РОЗРОБКА АЛГОРИТМІВ ФУНКЦІЙ

**Змінні:**  
string filename – назва файлу  
int \*\*InArray – покажчик на вхідний двовимірний масив, що буде зчитано зі вхідного файлу  
int \*OutArray – покажчик на вихідний масив, що буде записано у вихідний файл  
int \*sizes – покажчик на масив з кількостей елементів кожного масиву вхідного масиву  
int size – кількість елементів вхідного масиву  
int index – індекс, який потрібен для отримання доступу до елемента

**Функції:**  
 **int main ( )** Призначення: головна функція.  
 Опис роботи:  
 1. Запис назви вхідного файлу;  
 2. Створення вхідного двовимірного масиву та масиву з кількостей елементів;  
 3. Читання даних зі вхідного файлу;  
 4. Створення вихідного масиву;  
 5. Пошук максимальних значень, запис їх у вихідний масив;  
 6. Запис назви вихідного файлу;  
 7. Запис результату операції у вихідний файл;  
 8. Отримання елемента вхідного масиву за індексом;  
 9. Вивільнення динамічно-виділеної пам’яті.  
 Схема алгоритму подана на рис. 1.

**string SetInFilePath( )** Призначення: отримання назви вхідного файлу.  
 Опис роботи: введення з клавіатури назви вхідного файлу.  
 Схема алгоритму подана на рис. 2.

**string SetOutFilePath ( )** Призначення: отримання назви вихідного файлу.  
 Опис роботи: введення з клавіатури назви вихідного файлу.  
 Схема алгоритму подана на рис. 3.

**int ReadFromFile ( int \*\*&in, int \*&s, string path )** Призначення: читання даних з файлу.  
 Опис роботи: функція пробує відкрити файл за вказаним у змінній path шляхом. Якщо файлу не існує, функція викидає виключення. А якщо є, функція зчитує дані з файлу.  
 Повертає кількість елементів вхідного масиву.  
 Аргументи: in – вхідний масив;  
 s – масив кількостей елементів;  
 path – зміна, що зберігає назву файлу.  
 Схема алгоритму подана на рис. 4.

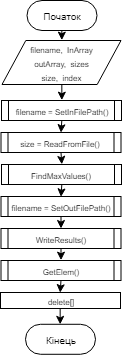
**int max ( int \*& arr, int size )**  
 Призначення: пошук максимального значення в масиві.  
 Опис роботи: використовуючи цикл і умову, функція знаходить максимальний елемент масиву.  
 Повертає максимальний елемент масиву.  
 Схема алгоритму подана на рис. 5.

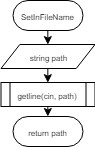
**void FindMaxValues ( int \*\*& in, int \*& out, int \*& s, int size )** Призначення: пошук максимальних значень вхідного масиву, запис результатів у вихідний масив.  
 Опис роботи: у циклі: функція присвоює кожному елементу вихідного масиву результат роботи функції *max*.  
 Схема алгоритму подана на рис. 6.

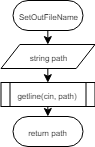
**void WriteResults ( int \*& out, int size, string path )** Призначення: запис результату операції у вихідний файл.  
 Опис роботи: за вказаним у змінній *path* шляхом функція створює файл, у який згодом, використовуючи цикл, записує дані вихідного масиву.  
 Схема алгоритму подана на рис. 7.

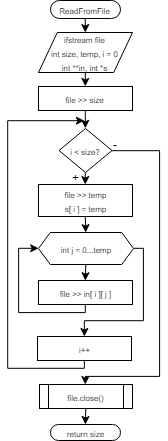
**int GetElem ( int index, int \*\*& in, int \*& s, int size )** Призначення: отримання елемента вхідного масиву за індексом.  
 Опис роботи: функція перевіряє, якщо індекс менший за 0 або більший за загальну кількість елементів, то вона викидає виключення. В іншому випадку функція використовує цикл та умови, знаходить потрібний елемент і повертає його як результат роботи функції.  
 Схема алгоритму подана на рис. 8.

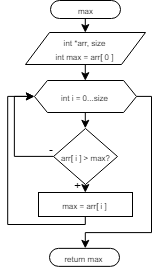
**Схеми алгоритмів функцій та методів.**

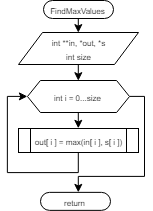
  
Рисунок 1 – Схема алгоритму функції main

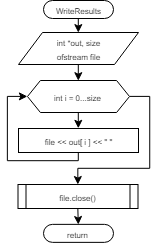
  
Рисунок 2 – Схема алгоритму функції SetInFileName

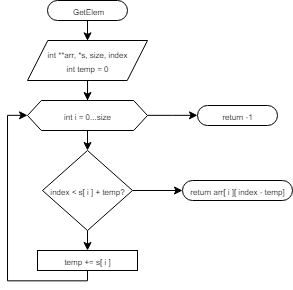
  
Рисунок 3 – Схема алгоритму функції SetOutFileName

  
Рисунок 4 – Схема алгоритму функції ReadFromFile

  
Рисунок 5 - Схема алгоритму функції маx

  
Рисунок 6 - Схема алгоритму функції FindMaxValues

  
Рисунок 7 – Схема алгоритму функції WriteResults

  
Рисунок 8 – Схема алгоритму функції GetElem

3 ТЕКСТ ПРОГРАМИ

**Текст файлу main.cpp**  
#include "Functions.h"

using namespace std;

int main()

{

setlocale(LC\_ALL, "rus");

string filename = SetInFilePath(); // Записываем название входного файла

int \*\*InArray = { 0 }; // Входной массив

int \*sizes = { 0 }; // Кол-во эл-тов входного массива

int size = ReadFromFile(InArray, sizes, filename); // Чтение данных для массива из файла

Print(InArray, sizes, size); // Вывод массива на экран

int \*OutArray = { 0 }; // Выходной массив

FindMaxValues(InArray, OutArray, sizes, size); // Поиск максимальных значений

filename = SetOutFilePath(); // Записываем название выходного файла

WriteResults(OutArray, size, filename); // Записываем данные выходного массива в файл

cout << "Введите индекс: ";

int index;

cin >> index;

cout << GetElem(index, InArray, sizes, size); // Получение элемента по индексу

delete[]OutArray;

delete[]sizes;

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

delete[]InArray[i];

delete[]InArray;

cout << "\nПрограмма завершила свою работу.\nПроверьте содержимое файла: " << filename << endl << endl;

system("PAUSE");

return 0;

}

**Текст файлу Functions.h**

#pragma once

#include <iostream>

#include <string>

#include <fstream>

using namespace std;

//Ввод названия входного файла

string SetInFilePath();

//Ввод названия выходного файла

string SetOutFilePath();

//Чтение входного файла

int ReadFromFile(int\*\* &in, int\* &s, string path);

int max(int\* &arr, int size);

//Поиск максимальных значений

void FindMaxValues(int\*\* &in, int\* &out, int\* &s, int size);

//Запись результатов в выходной файл

void WriteResults(int\* &out, int size, string filename);

//Получение значения массива по индексу

int GetElem(int index, int\*\* &arr, int\* &s, int size);

//Вывод содержимого массива на экран

void Print(int\*\* &arr, int\* &s, int size);

**Текст файлу Functions.cpp**

#include "Functions.h"

string SetInFilePath()

{

string path;

cout << "Введите путь ко входному файлу: ";

getline(cin, path);

return path;

}

string SetOutFilePath()

{

string path;

cout << "Введите путь к выходному файлу: ";

getline(cin, path);

return path;

}

int ReadFromFile(int\*\* &in, int\* &s, string path)

{

ifstream file;

file.exceptions(ifstream::badbit | ifstream::failbit);

try

{

file.open(path);

}

catch(const ifstream::failure& err)

{

cout << "Ошибка открытия файла.\n";

cout << err.what() << endl;

cout << err.code() << endl;

}

int size = 0;

file >> size;

in = new int\* [size];

s = new int[size];

int temp = 0;

int i = 0;

while (i < size)

{

file >> temp;

s[i] = temp;

in[i] = new int[temp];

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

{

file >> in[i][j];

}

i++;

}

file.close();

return size;

}

int max(int\* &arr, int size)

{

int max = arr[0];

for (int i = 1; i < size; i++)

if (arr[i] > max)

max = arr[i];

return max;

}

void FindMaxValues(int\*\* &in, int\* &out, int\* &s, int size)

{

out = new int[size];

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

{

out[i] = max(in[i], s[i]);

}

}

void WriteResults(int \*& out, int size, string filename)

{

ofstream file(filename);

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

{

file << out[i] << " ";

}

file.close();

}

int GetElem(int index, int \*\*& arr, int \*& s, int size)

{

int Size = 0;

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

Size += s[i];

try

{

if (index < 0 && index >= Size)

throw "Bad index.";

}

catch (char\* Msg)

{

cout << "Error: " << Msg << endl;

}

int temp = 0;

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

{

if (index < s[i] + temp)

return arr[i][index-temp];

else

temp += s[i];

}

}

void Print(int\*\* &arr, int\* &s, int size)

{

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

{

for (int j = 0; j < s[i]; j++)

{S

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

}

}

cout << endl;

}

4 РЕЗУЛЬТАТ РОБОТИ ПРОГРАМИ

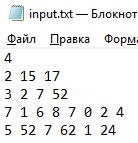
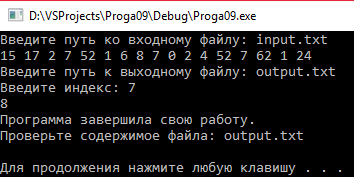
  

Рисунок 9 – Результат роботи програми.

ВИСНОВКИ

При виконанні лабораторної роботи написано програму, яка зчитує дані зі вхідного файлу, виконує індивідуальне завдання та записує результати у вихідний файл.  
 У функціях зчитування та отримання елемента за індексом використано обробку виключеннь, які перевіряють на коректність вхідні дані. При виявлені помилок, викидають виключення.  
 Набуто практичних навичок з розроблення програм із використанням обробки виключних ситуацій.