**Лабораторная работа №5**

**Динамическое программирование**

Составить программу, решающую следующую задачу. Входные данные находятся в текстовом файле input.txt. В первой строке файла указаны через пробел размеры массива, каждый из которых не превосходит 20. В последующих строках перечислены элементы массива – целые числа. Результатом работы программы должны быть оптимальное значение целевой функции и путь, при котором оно достигается.

Для указания пути вывести массив, отметив элементы пути либо другим цветом, либо некоторым другим образом.

**Вариант 17 (задание №5)**: Дан двумерный числовой массив размером N1xN2, в котором имеются отрицательные числа. Найти такой путь от нижней строки массива к верхней, чтобы сумма чисел по данному пути была максимальной. Из каждой клетки массива допустимо двигаться либо через одну клетку вверх, либо на одну клетку влево-вверх или вправо-вверх.

**Текст программы**

#include <iostream>

#include <fstream>

using namespace std;

int main() {

setlocale(LC\_ALL, "Russian");

ifstream f;

f.open("input.txt");

if (!f.is\_open()) {

cout << "Не удалось открыть файл" << endl;

return -1;

}

int n1, n2;

if (!(f >> n1 >> n2)) {

cout << "Некорректные данные в файле" << endl;

f.close();

return -1;

}

if (n1 > 20 || n2 > 20) {

cout << "Превышен максимально допустимый размер массива!" << endl;

return -1;

}

int\*\* a = new int\* [n1];

for (int i = 0; i < n1; i++)

a[i] = new int[n2];

int\*\* b = new int\* [n1];

for (int i = 0; i < n1; i++)

b[i] = new int[n2];

int\*\* c = new int\* [n1];

for (int i = 0; i < n1; i++)

c[i] = new int[n2];

for (int i = 0; i < n1; i++)

for (int j = 0; j < n2; j++) {

if (!(f >> a[i][j])) {

cout << "Некорректные данные в файле" << endl;

f.close();

return -1;

}

}

f.close();

for (int i = 0; i < n1; i++)

for (int j = 0; j < n2; j++)

c[i][j] = 0;

for (int j = 0; j < n2; j++) { //копируем последнюю строку

b[n1 - 1][j] = a[n1 - 1][j];

}

b[n1 - 2][0] = a[n1 - 2][0] + b[n1 - 1][1]; //записываем предпоследнюю строку

b[n1 - 2][n2 - 1] = a[n1 - 2][n2 - 1] + b[n1 - 1][n2 - 2];

for (int j = 1; j < n2 - 1; j++)

if (b[n1 - 1][j - 1] > b[n1 - 1][j + 1]) b[n1 - 2][j] = a[n1 - 2][j] + b[n1 - 1][j - 1];

else b[n1 - 2][j] = a[n1 - 2][j] + b[n1 - 1][j + 1];

for (int i = n1 - 3; i >= 0; i--) { //записываем все оставшиеся строки

if (b[i + 1][1] > b[i + 2][0]) b[i][0] = a[i][0] + b[i + 1][1];

else b[i][0] = a[i][0] + b[i + 2][0];

if (b[i + 1][n2 - 2] > b[i + 2][n2 - 1]) b[i][n2 - 1] = a[i][n2 - 1] + b[i + 1][n2 - 2];

else b[i][n2 - 1] = a[i][n2 - 1] + b[i + 2][n2 - 1];

for (int j = 1; j < n2 - 1; j++) {

int max = b[i + 2][j];

if (b[i + 1][j - 1] > max) max = b[i + 1][j - 1];

if (b[i + 1][j + 1] > max) max = b[i + 1][j + 1];

b[i][j] = a[i][j] + max;

}

}

int max = b[0][0], maxi = 0, maxj = 0; //находим максимум в верхней строке

for (int j = 0; j < n2; j++) {

if (b[0][j] > max) {

max = b[0][j];

maxj = j;

}

}

c[maxi][maxj] = 1;

cout << "\033[1;33m";

cout << "Максимальная сумма: " << max << endl << endl;

cout << "\033[0m";

while (maxi < n1 - 2) { //проход по массиву в обратном порядке

if (maxj == 0)

if (b[maxi + 2][maxj] > b[maxi + 1][maxj + 1]) {

maxi += 2;

c[maxi][maxj] = 1;

}

else {

maxi++;

maxj++;

c[maxi][maxj] = 1;

}

else if (maxj == n2 - 1)

if (b[maxi + 2][maxj] > b[maxi + 1][maxj - 1]) {

maxi += 2;

c[maxi][maxj] = 1;

}

else {

maxi++;

maxj--;

c[maxi][maxj] = 1;

}

else {

int k = 0;

max = b[maxi + 2][maxj];

if (b[maxi + 1][maxj - 1] > max) {

max = b[maxi + 1][maxj - 1];

k = 1;

}

if (b[maxi + 1][maxj + 1] > max) {

max = b[maxi + 1][maxj + 1];

k = 2;

}

switch (k) {

case 0: maxi += 2;

break;

case 1: {

maxi++;

maxj--;

}

break;

case 2: {

maxi++;

maxj++;

}

break;

}

c[maxi][maxj] = 1;

}

}

if (maxi != n1 - 1) {

if (maxj == 0) {

maxi++;

maxj++;

c[maxi][maxj] = 1;

}

else if (maxj == n2 - 1) {

maxi++;

maxj--;

c[maxi][maxj] = 1;

}

else {

if (b[maxi + 1][maxj - 1] > b[maxi + 1][maxj + 1]) {

maxi++;

maxj--;

c[maxi][maxj] = 1;

}

else {

maxi++;

maxj++;

c[maxi][maxj] = 1;

}

}

}

cout << "\033[1;33m";

cout << "Исходный массив:" << endl;

cout << "\033[0m";

for (int i = 0; i < n1; i++) {

for (int j = 0; j < n2; j++) {

if (c[i][j] == 1) cout << "\033[1;31m";

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

if (c[i][j] == 1) cout << "\033[0m";

}

cout << endl;

}

cout << endl;

cout << "\033[1;33m";

cout << "Массив с наибольшими суммами для каждой ячейки" << endl;

cout << "\033[0m";

for (int i = 0; i < n1; i++) {

for (int j = 0; j < n2; j++) {

if (c[i][j] == 1) cout << "\033[1;34m";

cout << b[i][j] << ' ';

if (c[i][j] == 1) cout << "\033[0m";

}

cout << endl;

}

cout << endl;

cout << "\033[1;33m";

cout << "Массив пути:" << endl;

cout << "\033[0m";

for (int i = 0; i < n1; i++) {

for (int j = 0; j < n2; j++) {

if (c[i][j] == 1) cout << "\033[1;32m";

cout << c[i][j] << ' ';

if (c[i][j] == 1) cout << "\033[0m";

}

cout << endl;

}

cout << endl;

for (int i = 0; i < n1; i++) {

delete[] a[i];

delete[] b[i];

delete[] c[i];

}

delete[] a;

delete[] b;

delete[] c;

system("pause");

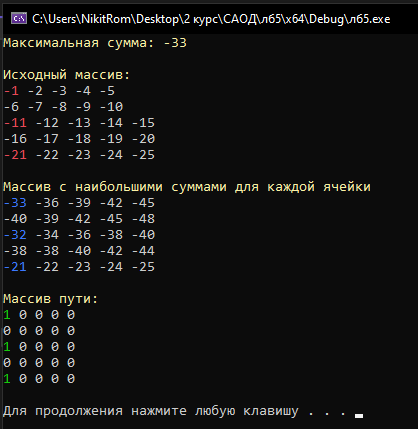
return 0;

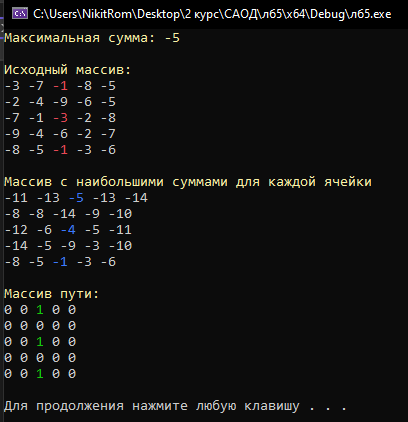
}

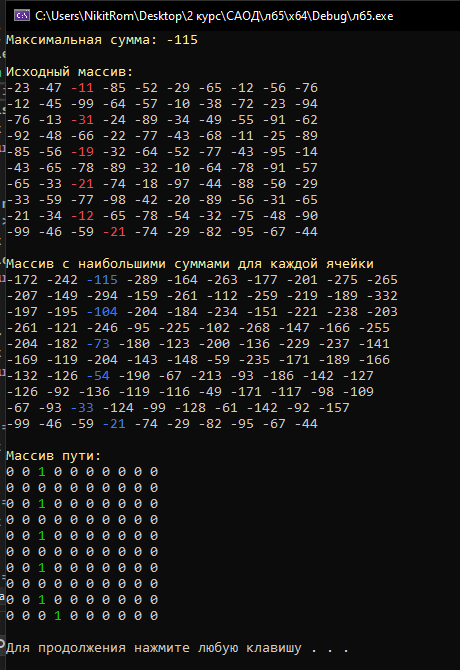
**Описание программы**

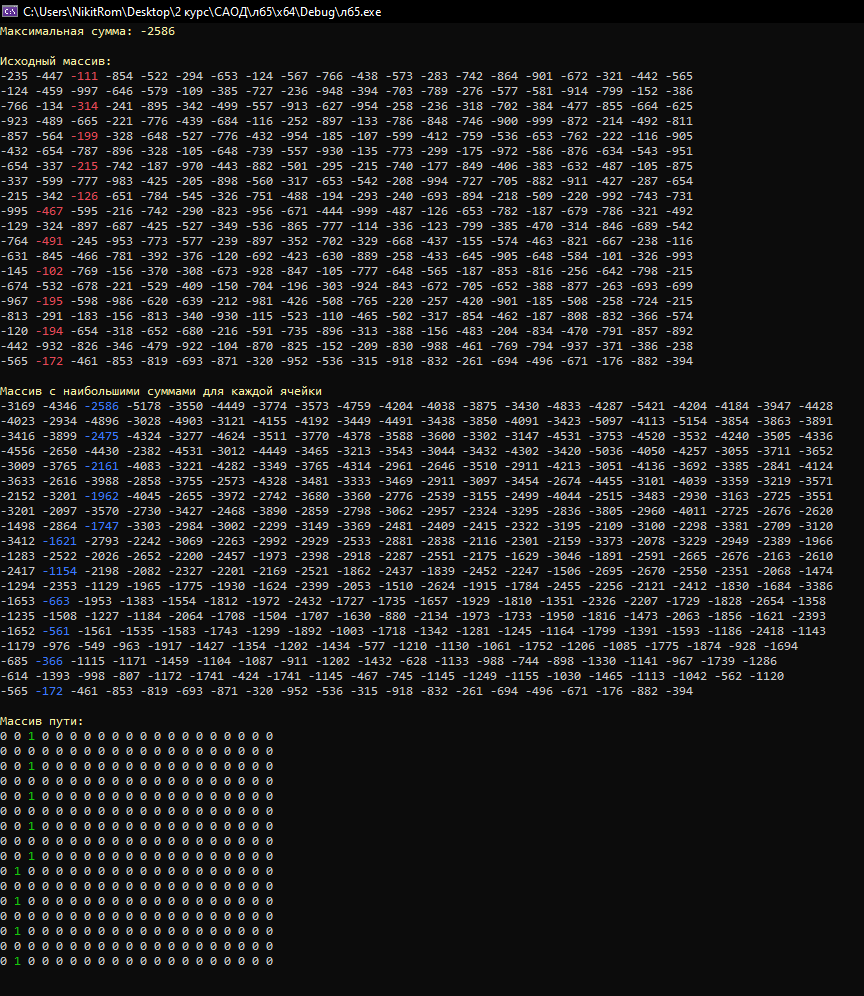
1. Создается объект **f** типа **ifstream**. Открывается файл **input.txt**
2. Если такого файла нет, на экран выводится сообщение «*Не удалось открыть файл*», выполнение программы завершается досрочно
3. Данные файла проверяются на корректность (соответствие условию). Если не соответствуют – выводится сообщение «*Некорректные данные в файле*», выполнение программы завершается досрочно
4. Инициализируются переменные **n1**, **n2** – размеры массива. Их значение считывается из файла (первые 2 числа).
5. Инициализируются двумерные динамические массивы **a[n1][n2]**, **b[n1][n2]**, **c[n1][n2]**
6. Если **n1 > 20** и/или **n2 > 20**, выводится сообщение «Превышен максимально допустимый размер массива», выполнение программы завершается досрочно
7. Считываются данные из файла, записываются в массив **a[n1][n2]**
8. Файл f закрывается
9. Массив **c[n1][n2]** заполняется нулями
10. Значения в строке n1-1 массива **a[n1][n2]** копируются в последнюю строку массива **b[n1][n2]**
11. Вычисляется значение ячеек предпоследней строки массива **b[n1][n2]** по принципу: если верхняя левая ячейка больше верхней правой, то значение действующей ячейки равно сумме значения ячейки с такими же индексами в массиве **a[n1][n2]** и значения той из 2 ячеек, что больше
12. Вычисляется значение ячеек оставшихся строк по принципу: значение ячейки равно сумме значения ячейки с такими же индексами в массиве **a[n1][n2]** и значения наибольшей ячейки из трех: левой верхней, правой верхней и верхней ячейки через одну
13. Инициализируются переменные **max**, **maxi**, **maxj**. Переменной max присваивается наибольшее значение из 1 строки массива **b[n1][n2]**, а переменным **maxi** и **maxj** – значения индексов этой ячейки
14. На экран выводится значение переменной **max**
15. Производится обратный описанному в пунктах 11-12 проход по массиву **b[n1][n2]**, от ячейки **[maxi][maxj]**, равной **max**, до последней строки. При промежуточных значениях, в массиве **c[n1][n2]** ячейкам с соответствующими индексами присваивается значение 1
16. На экран выводятся 3 массива – a (исходный), b (массив с наибольшими суммами), c (массив пути). При этом путь (индексы ячейек, которые в массиве **c[n1][n2]** равны 1) выделяется цветом во всех 3 массивах.
17. Удаляются динамические массивы **a**, **b**, **c**

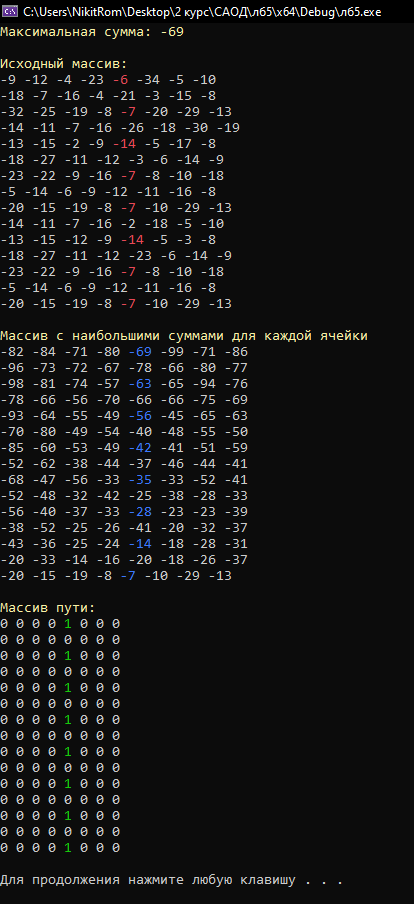
**Результаты тестов**











Файл с буквами вместо чисел:



Файл с неверными размерами массива:



Файл с массивом размером 21х21:



Указание несуществующего файла:



**Вывод**

Мною была написана программа, находящая оптимальное значение целевой функции и путь, при котором она достигается, с помощью методов динамического программирования.

Программа берет входные данные (массив отрицательных чисел) из файла. Программа ищет такой путь от нижней строки массива к верхней, чтобы сумма чисел по данному пути была максимальной, а затем выводит массив, отметив элементы пути другим цветом.