Лабораторная работа №5

Динамическое программирование

Задание:

Составить программы, решающие следующие задачи. 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 (кроме задачи 6) вывести массив, отметив элементы пути либо другим цветом, либо некоторым другим образом.

Дан двумерный числовой массив размером N1xN2. Найти такой путь от крайнего левого столбца массива к крайнему правому, чтобы сумма чисел по данному пути была минимальной. Из каждой клетки массива допустимо двигаться вверх, вниз, вправо, вправо-вниз или вправо-вверх.

Код программы:

#include <iostream>

#include <fstream>

#include <Windows.h>

#include <string>

using namespace std;

int main() {

HANDLE hConsoleHandle = GetStdHandle(STD\_OUTPUT\_HANDLE); // цвет

setlocale(LC\_ALL, "RU");

int c = 0;

int N1, N2; // размерность массива

ifstream vd("input.txt");

vd >> N1 >> N2;

cout << N1 << " " << N2 << endl;

int\*\* inpt = new int\* [N1];

// заполнение массива

for (int i = 0; i < N1; i++) {

inpt[i] = new int[N2];

for (int j = 0; j < N2; j++) {

vd >> inpt[i][j];

}

}

vd.close();

// вывод массива

for (int i = 0; i < N1; i++) {

for (int j = 0; j < N2; j++) {

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

}

cout << endl;

}

// массив для нахождения пути

int\*\* sc = new int\* [N1];

for (int i = 0; i < N1; i++) {

sc[i] = new int[N2];

for (int j = 0; j < N2; j++) {

sc[i][j] = 0;

}

}

int\*\* da = new int\* [N1];

for (int i = 0; i < N1; i++) {

da[i] = new int[N2];

for (int j = 0; j < N2; j++) {

da[i][j] = 0;

}

}

//массив для вычислений

int\*\* put = new int\* [N1];

for (int i = 0; i < N1; i++) {

put[i] = new int[N2];

}

for (int i = 0; i < N1; i++) {

put[i][0] = inpt[i][0];

sc[i][0] = 4;

}

for (int i = 1; i < N2; i++) {

for (int j = 0; j < N1; j++) {

if (N1 != 1) {

if (j == N1 - 1 || (j != 0 && (put[j - 1][i - 1] < put[j + 1][i - 1]))) {

put[j][i] = inpt[j][i] + put[j - 1][i - 1];

sc[j][i] = 1;

}

else {

put[j][i] = inpt[j][i] + put[j + 1][i - 1];

sc[j][i] = -1;

}

}

}

for (int j = 0; j < N1; j++) {

for (int j = 0; j < N1; j++) {

if (j != 0 && (put[j][i] > inpt[j][i] + put[j - 1][i])) {

put[j][i] = inpt[j][i] + put[j - 1][i];

sc[j][i] = 2;

}

if (j != N1 - 1 && (put[j][i] > inpt[j][i] + put[j + 1][i])) {

put[j][i] = inpt[j][i] + put[j + 1][i];

sc[j][i] = -2;

}

}

if (i!=0 &&(put[j][i] > inpt[j][i] + put[j][i - 1])) {

put[j][i] = inpt[j][i] + put[j][i - 1];

sc[j][i] = 0;

}

}

}

cout << "---------------------------" << endl;

//поиск минимального значения

int min=put[0][N2-1];

int sa=0;// переенная счиатющая конец цикла

for (int i = 0; i < N1; i++) {

if (min > put[i][N2 - 1]) {

sa = i;

min = put[i][N2 - 1];

}

}

cout << "Минимальная сумма пути = " << min << endl;

cout << "---------------------------" << endl;

int N3 = N2 - 1;

while (sc[sa][N3] != 4) {

for (int i = 0; i < N1; i++) {

if (i == sa) {

da[i][N3] = 1;

}

}

if (sc[sa][N3] ==1)

sa -=1;

else if (sc[sa][N3] == -1)

sa +=1;

else if (sc[sa][N3] == 2) {

sa-=1;

N3 ++;

}

else if (sc[sa][N3] == -2) {

sa += 1;

N3++;

}

N3--;

}

for (int i = 0; i < N1; i++) {

if (i == sa) {

da[i][0] = 1; }

else {

da[i][0] = 0; }

}

for (int i = 0; i < N1; i++) {

for (int j = 0; j < N2; j++) {

if (da[i][j] == 1) {

SetConsoleTextAttribute(hConsoleHandle,9);

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

}

else {

SetConsoleTextAttribute(hConsoleHandle, 4);

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

}

}

cout << endl;

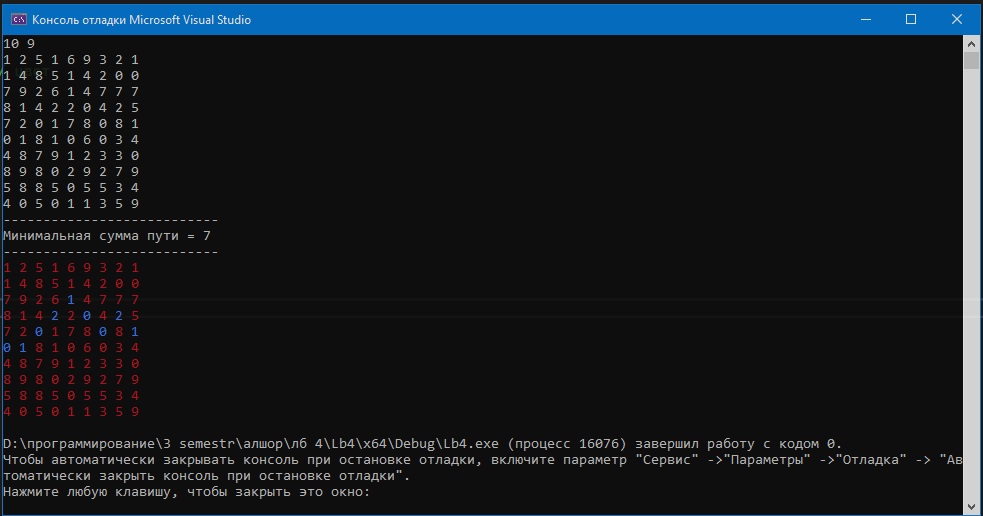
}

SetConsoleTextAttribute(hConsoleHandle, 7);

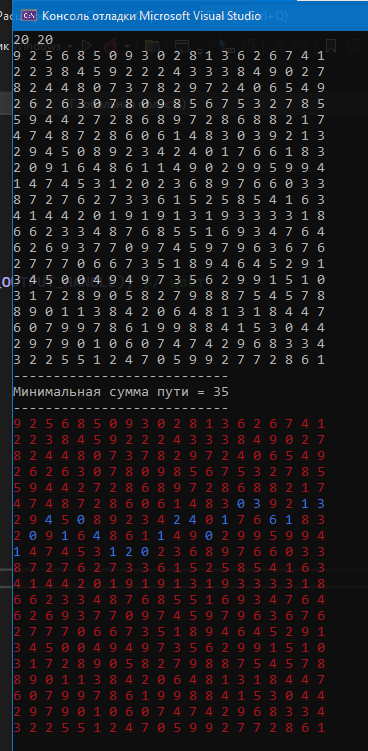
}

Тесты

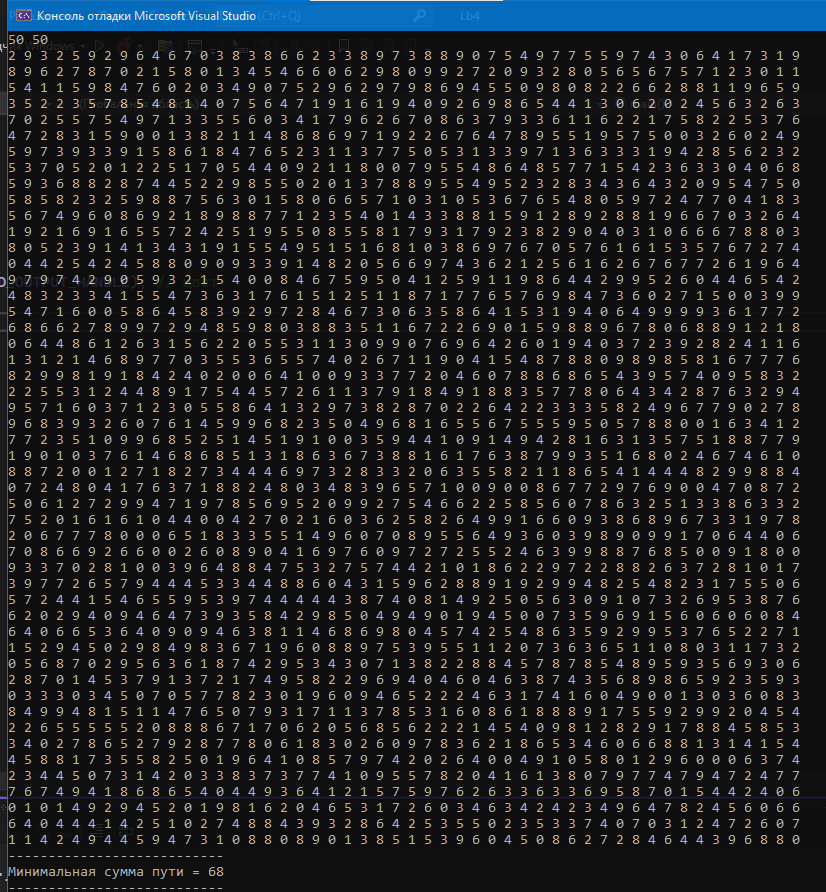
1)

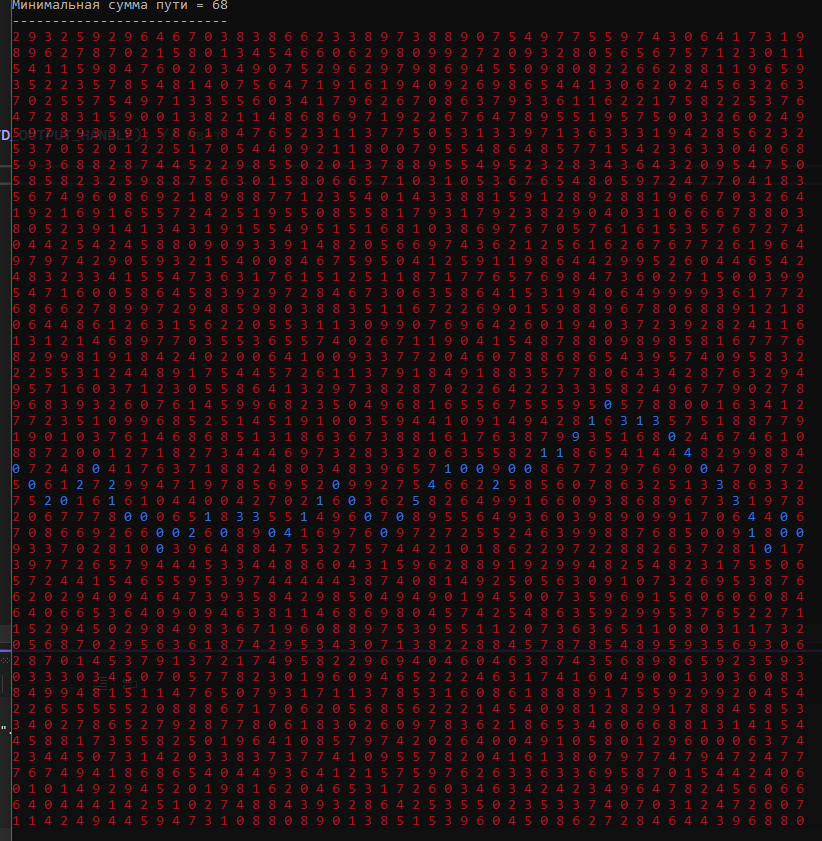


2)



3)





**Описание программы:**

* Инициализация переменной c равной 0;
* Объявление переменной N1 (кол-во строк) и N2 (кол-во столбцов);
* Создаём объект vd класса ifstream для открытия и чтения файла “input.txt”;
* Считываем размерность массива из файла в переменные N1 и N2 и выводим их на экран;
* Объявление двумерного динамического массива inpt размером N1 x N2;
* C помощью двух циклов for заполняем массив inpt числами из файла;
* Закрываем файл.
* Выводим массив inpt на экран.
* Объявление двумерного динамического массива sc размером N1 x N2 (будет использоваться для обозначения шагов), а также с помощью 2 циклов for заполнение массива sc нулями;
* Объявление двумерного динамического массива da размером N1 x N2 (будет использоваться для обозначения пути), а также с помощью 2 циклов for заполнение массива da нулями;
* Объявление двумерного динамического массива put размером N1 x N2(будет использоваться для вычислений кротчайшего пути;
* C помощью цикла for заполняем первый столбец массива put значениями первого столбца массива inpt, и заполняем первый столбец массива sc четверками.
* С помощью цикла for по i (от 1 до N2) перебираем элементы массива по столбцам;
* С помощью цикла for по j (от 0 до N1) перебираем элементы массива по строкам;

(Ищем лучший путь в ячейку по диагонали)

* + Используя условный оператор if проверяем количество строк;
  + Если N1 не равно 1, выполняет следующие действия:
  + С помощью оператора if проверяем последняя это строка ИЛИ (строка не нулевая И ( значение верней левой ячейки массива put меньше чем значение нижней левой строки массива put);
    - В ячейку массива put[j][i] записываем сумму значений ячейки inpt[j][i] и верней левой ячейки массива put;
    - В ячейку sc[j][i] записываем значение шага равное 1(пришли в текущую ячейку из левой верхней);
    - Иначе в ячейку массива put[j][i] записываем сумму значений ячейки inpt[j][i] и нижней левой ячейки массива put;
    - В ячейку sc[j][i] записываем значение шага равное

-1(пришли в текущую ячейку из нижней левой);

(Ищем лучший путь в ячейку по горизонтали и вертикали)

* + Используя цикл for по j (от 0 до N1) перебираем элементы массива по строкам;
    - Используя цикл for по j (от 0 до N1) перебираем элементы массива по строкам
      * C помощью оператора if проверяем: не нулевая данная строка (j!=0) И ( значение ячейки массива put[j][i] больше суммы ячейки массива inpt[j][i] и верхней ячейки массива put);
      * Если условие выполняется, то в ячейку массива put[j][i] записывается сумма ячейки inpt[j][i] и верхней ячейки массива put;
      * В ячейку массива sc[j][i] записываем значение шага равное 2(пришли в текущую ячейку из верхней);

* + - * C помощью оператора if проверяем: не последняя данная строка (j!=N1-1) И ( значение ячейки массива put[j][i] больше суммы ячейки массива inpt[j][i] и нижней ячейки массива put);
      * Если условие выполняется, то в ячейку массива put[j][i] записывается сумма ячейки inpt[j][i] и нижней ячейки массива put;
      * В ячейку массива sc[j][i] записываем значение шага равное -2(пришли в текущую ячейку из нижней);
    - С помощью оператора if проверяем: данный столбец не нулевой (i!=0 И (значение ячейки put[j][i] больше суммы ячейки inpt[j][i] и левой ячейки массива put);
      * Если условие выполняется, то в ячейку массива put[j][i] записывается сумма ячейки inpt[j][i] и левой ячейки массива put;
      * В ячейку массива sc[j][i] записываем значение шага равное 0(пришли в текущую ячейку из левой);
* Инициализируем переменную min равную значению последнего элемента в первой строке массива put;
* Инициализируем переменную sa равную 0;
* C помощью цикла перебираем все элементы из последнего столбца массива put;
* C помощью оператора if ищем минимальное значение в последнем столбце массива put и присваиваем номер строки переменной sa;
* Выводим минимальную сумму пути;
* Инициализируем переменную N3 равную N1- 1;
* Используя цикл while перебираем элементы массива sc пока не встретится элемент со значением 4;
  + Используя цикл for и оператор if ищем строку в которой будет конец пути (i==sa), когда её нашли записываем в ячейку массива da[i][N3] единицу;
  + С помощью оператора условия if проверяем значение элемента массива sc[sa][N3], равно ли оно 1;
    - Если условие выполняется, то значение sa уменьшается на 1;
  + Иначе если значение элемента массива sc[sa][N3] равно -1, то:
    - Значение sa увеличивается на 1;
  + Иначе если значение элемента массива sc[sa][N3] равно 2, то:
    - Значение sa уменьшается на 1;
    - Значение N3 увеличивается на 1;
  + Иначе если значение элемента массива sc[sa][N3] равно -2, то:
    - Значение sa увеличивается на 1;
    - Значение N3 увеличивается на 1;
* С помощью цикла for перебираем элементы массива da;
  + С помощью оператора if проверяем i равно sa;
  + Если да, то da[i][0] приравниваем к 1;
  + Иначе da[i][0] приравниваем к 0;
* С помощью двух циклов for перебираем элементы массива da;
  + С помощью условного оператора if проверяем значение ячейки da[i][j] равно 1 или нет;
  + Если да, то выводим элемент массива inpt меняя его цвет на синий;
  + Иначе выводим элемент массива inpt меняя его цвет на красный;