ЦЕЛЬ РАБОТЫ

Продемонстрировать работу с механизмами динамического выделения и освобождения памяти на примере решения задач с двухмерными массивами.

ПРАКТИЧЕСКОЕ ЗАДАНИЕ

Составить программу, которая:

1. Запрашивает у пользователя размерность массива чисел и динамически выделяет память для хранения указанного количества значений.
2. Запрашивает у пользователя массив чисел и выводит этот массив на экран. Среди чисел должны быть отрицательные, положительные и равные нулю. Дополнительные параметры, если они упоминаются в тексте задания, вводятся с клавиатуры.
3. Вычисляет значения переменных согласно варианту задания.
4. выводит значения всех полученных переменных с пояснениями.

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

Вариант 10

Даны вещественные массивы B[5][4]. Определить значения целочисленного массива D[5] по следующему правилу: D[i]=1, если сумма элементов i-ой строки больше 0, D[i]=0 в противном случае. Значение переменной i изменяется от 0 до 4

БЛОК-СХЕМА АЛГОРИТМА

Блок-схема алгоритма представлена на рис. 1.



*Рисунок 1 Блок-схема алгоритма*

ЛИСТИНГ

#include <iostream>

#include <conio.h>

using namespace std;

int main(void) {

//объявление переменных

int n, m, i, j, k=0, l;

//ввод размерности массива

cout<<"vvedite rasmernost massiva"<<endl;

cin>>n>>m;

cout<<"vvedite massiv"<<endl;

//объявление массивов

float M[n][m], s=0;

int P[k];

//ввод массива

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

{

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

{

cin>>M[i][j];

}

}

//подсчет суммы в строках массива

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

{

s=0;

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

{

s=M[i][j]+s;

}

//условие для добавления 0/1 в одномерный массив

if (s==0)

{

P[k]=0;

k++;

}

else

{

P[k]=1;

k++;

}

}

//вывод получившегося массива на экран

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

{

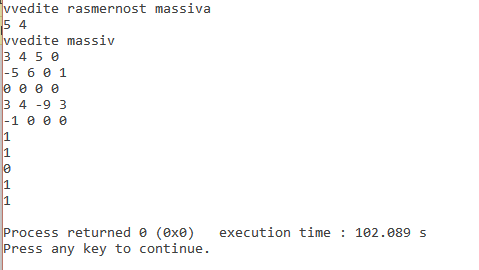
cout<<P[i]<<endl;

}

}

ПРОТОКОЛ ВЫПОЛНЕНИЯ ПРОГРАММЫ

Протокол выполнения программы представлен на рис. 2.



*Рисунок 2 Протокол выполнения программы*