Беларускі дзяржаўны тэхналагічны ўніверсітэт

Факультэт інфармацыйных тэхналогій

Кафедра праграмнай інжэнерыі

Лабараторная работа 15

Па дысцыпліне «Асновы алгарытмізацыі і праграмавання»

На тэму «Дынамічныя масівы»

Выканала:

Студэнтка 1 курса 6 группы

Жучкевіч Кацярына Сяргееўна

Выкладчык: асс. Андронава М.В.

2023, Мінск

**Варыянт 9**

|  |  |
| --- | --- |
| **9** | 1. В одномерном массиве, состоящем из n вещественных элементов, вычислить произведение положительных элементов массива и сумму элементов массива, расположенных до минимального элемента.  2. Найти в матрице первую строку, все элементы которой положительны, и сумму этих элементов. Уменьшить все элементы матрицы на эту сумму. |

#include <stdio.h>

#include <stdlib.h>

void main()//пачатак праграмы

{

int\* ptr, i, n;

int min, start, end, sum = 0, mult = 1;

printf("Input size of massiv, n<30 \n");

scanf\_s("%d", &n);

if (!(ptr = (int\*)malloc(n \* sizeof(int)))) //надаем памяць

{ //і праглядаем, хапае лі месца

puts("Not enough memory");

return;

}

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

{

printf("Input element [%d]\n", i + 1);//уводзім элементы ў масіў

scanf\_s("%d", ptr + i);

}

printf("\nMassiv: \n");

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

printf("%d ", \*(ptr + i));

min = \*(ptr + 0);

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

if (min > \*(ptr + i)) {//шукаем мінімальны элемент

min = i;

}

}

printf("\nMinimal %d\n", min);//выпісваем мінімальны элемент

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

if (\*(ptr + i) > 0) {

mult \*= \*(ptr + i);

}

}

for (i = 0; i < min; ++i) {

sum += \*(ptr + i);

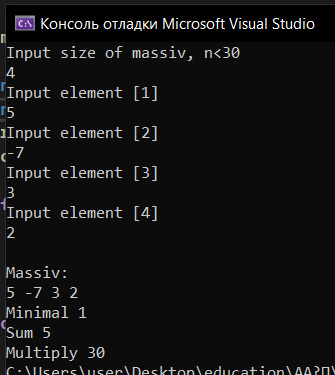
}

printf("Sum %d", sum);

printf("\nMultiply %d", mult);

free(ptr); //вызваляем дынаміч памяць

}

****

#include <iostream>

using namespace std;

int main()

{

int size, \*\* A, rowSum = 0 , check;

cout << "Enter size ";

cin >> size;

A = new int\* [size]; //надаем памяць

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

A[i] = new int[size]; //надаем памяць

for (int j = 0; j < size; j++) {//уводзім масіў

cin >> A[i][j];

}

}

for (int i = 0; i < size; i++)//пераўтвараем масіў

{

cout << "\n";

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

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

}

for (int i = 0; i < size; i++) { //двайны цыкл, у якім сэнс практыкавання

check = 1;

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

if (A[i][j] <= 0) {

check = 0;

rowSum = 0;

break;

}

rowSum += A[i][j];

}

if (check == 1) {

break;

}

}

cout << "\n";

for (int i = 0; i < size; i++)//адымаем суму

{

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

A[i][j] -= rowSum;

}

for (int i = 0; i < size; i++)//выводзім тое, што атрымалі

{

cout << "\n";

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

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

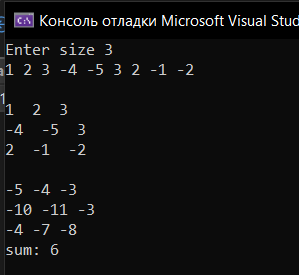
}

cout << "\nsum: " << rowSum << endl;

delete[] A;//вызваляем памяць

return 0;

}

****

|  |  |
| --- | --- |
| **8** | 1. В одномерном массиве, состоящем из **k** целых элементов, вычислить количество положительных элементов массива и сумму элементов массива, расположенных после последнего элемента, равного нулю.  2. Дана целочисленная прямоугольная матрица. Определить номер первого из столбцов, содержащих хотя бы один нулевой элемент. |

#include <stdio.h>

#include <stdlib.h>

void main()//пачатак праграмы

{

int\* ptr, i, k;

int count = 0, ind, sum = 0;

printf("Input size of massiv, k<30 \n");

scanf\_s("%d", &k);

if (!(ptr = (int\*)malloc(k \* sizeof(int)))) //надаем памяць

{ //і праглядаем, хапае лі месца

puts("Not enough memory");

return;

}

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

{

printf("Input element [%d]\n", i + 1);//уводзім элементы ў масіў

scanf\_s("%d", ptr + i);

}

printf("\nMassiv: \n");

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

printf("%d ", \*(ptr + i));

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

if (\*(ptr + i) > 0) {//шукаем, колькі станоўчых элементаў

count++;

}

}

for (i = 0; i < k; i++) {//шукаем індэкс апошнега нуля

if (\*(ptr + i) == 0) {

ind = i;

}

}

for (i = 1; i < k; ++i) {//шукаем суму пасля нуля

sum += \*(ptr + i);

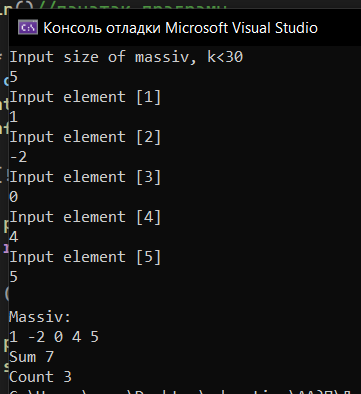
}

printf("\nSum %d", sum);

printf("\nCount %d", count);

free(ptr); //вызваляем дынаміч памяць

}

****

#include <iostream>

using namespace std;

int main()

{

int indJ, r, c;

int\*\* A;

cout << "Enter size (row): ";

cin >> r;

cout << "Enter size (column): ";

cin >> c;

A = new int\* [r]; //надаем памяць

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

A[i] = new int[c]; //надаем памяць

for (int j = 0; j < c; j++) {//уводзім масіў

cin >> A[i][j];

}

}

for (int i = 0; i < r; i++)//пераўтвараем масіў

{

cout << "\n";

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

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

}

for (int i = 0; i < r; i++) { //двайны цыкл, у якім сэнс практыкавання

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

if (A[i][j] == 0) {

indJ = j;

break;

}

}

}

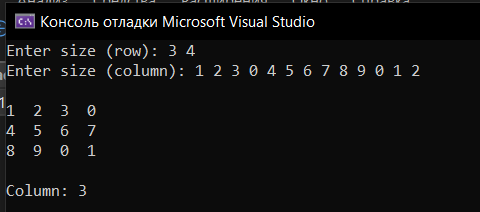
cout << "\n";

cout << "\nColumn: " << indJ + 1 << endl;

delete[] A;//вызваляем памяць

return 0;

}

****

|  |  |
| --- | --- |
| **11** | 1. Задан массив **A** из **n** элементов. Проверить, есть ли в нём элементы, равные нулю. Если есть, найти наименьшее **k**, при котором **A[k] = 0**.  2. Для заданной матрицы размером 4 на 4 найти такие **k**, при которых **k**-я строка матрицы совпадает с **k**-м столбцом. |

#include <stdio.h>

#include <stdlib.h>

void main()//пачатак праграмы

{

int\* ptr, i, k;

int indi;

printf("Input size of massiv, k<30 \n");

scanf\_s("%d", &k);

if (!(ptr = (int\*)malloc(k \* sizeof(int)))) //надаем памяць

{ //і праглядаем, хапае лі месца

puts("Not enough memory");

return;

}

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

{

printf("Input element [%d]\n", i + 1);//уводзім элементы ў масіў

scanf\_s("%d", ptr + i);

}

printf("\nMassiv: \n");

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

printf("%d ", \*(ptr + i));

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

if (\*(ptr + i) == 0) {//шукаем першы нуль

indi = i;

break;

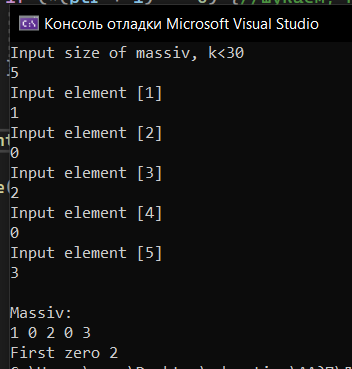
}

}

printf("\nFirst zero %d", indi + 1);

free(ptr); //вызваляем дынаміч памяць

}

****

#include <iostream>

using namespace std;

int main()

{

int size = 4, k;

int\*\* A;

cout << "Matrix: ";

A = new int\* [size]; //надаем памяць

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

A[i] = new int[size]; //надаем памяць

for (int j = 0; j < size; j++) {//уводзім масіў

cin >> A[i][j];

}

}

for (int i = 0; i < size; i++)//пераўтвараем масіў

{

cout << "\n";

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

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

}

cout << "\n";

for (int i = 0; i < size; i++) { //двайны цыкл, у якім сэнс практыкавання

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

if (i == j) {

k = i + 1;

cout << k << " ";

}

}

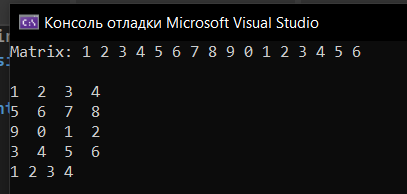
}

cout << "\n";

delete[] A;//вызваляем памяць

return 0;

}

****

**(Поняла это задание как задание, где просят найти индекс строки, которая совпадает с индексом колонки. То есть, все подходят (: )**