Беларускі дзяржаўны тэхналагічны ўніверсітэт

Факультэт інфармацыйных тэхналогій

Кафедра праграмнай інжэнерыі

 Лабараторная работа 15

Па дысцыпліне «Асновы алгарытмізацыі і праграмавання»

На тэму «Дынамічнае выдзяленне памяці»

Выканала:

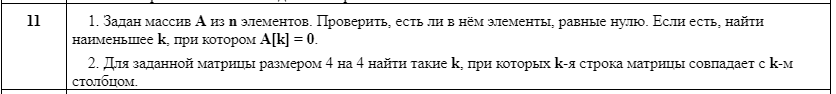
Студэнтка 1 курса 6 группы

Лускіна Вольга Аляксандраўна

Выкладчык: асс. Андронава М.В.

2023, Мінск

Варыянт 11



Заданне 1

#include <iostream>

#include <windows.h>

#include <cstdlib> // Бібліятэка для выкарыстання функцый, звязаных з выдзяленнем і кірававаннем дынамічнай памяці

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int n;

cout << "Увядзіце колькасць элементаў масіва: " << endl;

cin >> n;

cout << "Увядзіце элементы масіва: " << endl;

bool t = false; // Сцяг, каб праверыць наяўнасць нуля

int\* a = (int\*)malloc(n \* sizeof(int)); // Выдзяляем памяць для масіва з выкарыстаннем функцаі malloc

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

cin >> \*(a + i); // Уводзім элементы масіва

}

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

if (\*(a + i) == 0) { // Шукаем элемент роўны нулю

cout << "Элемент з індэксам " << i << " роўны 0" << endl; // Выводзім яго індэкс

t = true; // Адзначаем, што такі элемент ёсць

break;

}

}

if (t == false) { // Калі няма такога элемента

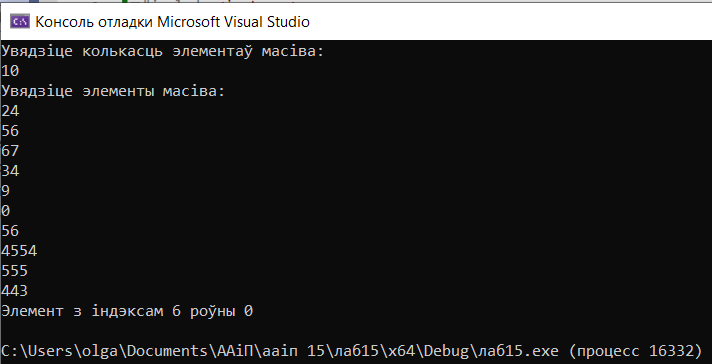
cout << "У масіве няма элементаў роўных 0" << endl;

}

free(a); // Ачышчаем памяць з выкарыстаннем функцыі free

return 0;

}



Заданне 2

#include <iostream>

#include <windows.h>

#include <cstdlib> // Бібліятэка для выкарыстання функцый, звязаных з выдзяленнем і кірававаннем дынамічнай памяці

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int n, s = 0;

cout << "Увядзіце колькасць элементаў масіва: " << endl;

cin >> n;

cout << "Увядзіце элементы масіва: " << endl;

bool t = false;

int\*\* a = new int\* [n]; // Робім масіў

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

a[i] = new int[n]; // Выдзяляем памяць для кожнага радка масіва

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

cin >> a[i][j]; // Уводзім элементы

}

}

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

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

if (a[i][j] != a[j][i]) { // Правяраем равенства элементаў сіметрычных пазіцый

break;

}

else if (a[i][j] == a[j][i]) {

s += 1; // Павелічэнне лічыльніка, калі элементы роўныя

}

}

if (s == n) { // Калі лічыльнік роўны памеру масіва

cout << "Індэкс роўных радка і слупка = " << i << endl; // Вывад індэкса радка і слупка

t = true; // Ставім сцяг, калі знайшлі радок і слупок

}

s = 0; // Збрасваем лічыльнік перад праверкай наступнага радка

}

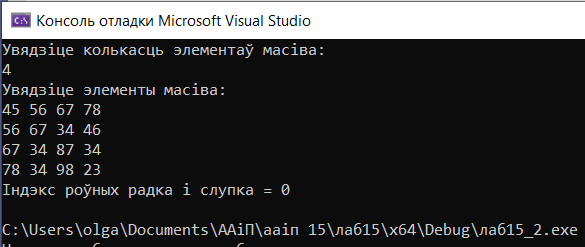
if (t == false) {

cout << "Няма радка і слупка з аднолькавым індэксам" << endl;

}

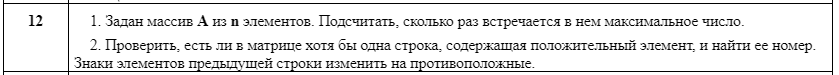
delete[] a; // Ачышчаем памяць, якую выдзялілі для двумернага масіва

}



Дадатковыя заданні:

Варыянт 12



Заданне 1

#include <iostream>

#include <windows.h>

#include <cstdlib> // Бібліятэка для выкарыстання функцый, звязаных з выдзяленнем і кірававаннем дынамічнай памяці

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int n, kol = 0;

cout << "Увядзіце колькасць элементаў масіва: " << endl;

cin >> n;

cout << "Увядзіце элементы масіва: " << endl;

int\* a = (int\*)malloc(n \* sizeof(int)); // Выдзяляем памяць для масіва з выкарыстаннем функцаі malloc

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

cin >> \*(a + i); // Уводзім элементы масіва

}

int max = \*a; // Прысвойваем зменнай max значэнне першага элемента

for (int i = 0; i < n; i++) { // Шукаем максімальны элемент у масіве

if (\*(a + i) > max) {

max = \*(a + i);

}

}

for (int i = 0; i < n; i++) { // Падлічваем, колькі разоў сустракаецца максімальны элемент у масіве

if (\*(a + i) == max) {

kol += 1;

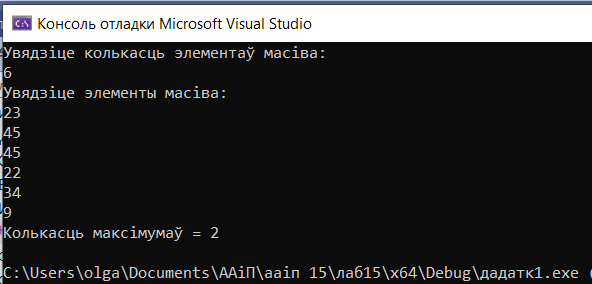
}

}

cout << "Колькасць максімумаў = " << kol << endl;

free(a); // Ачышчаем памяць з выкарыстаннем функцыі free

}



Заданне 2

#include <iostream>

#include <windows.h>

#include <cstdlib> // Бібліятэка для выкарыстання функцый, звязаных з выдзяленнем і кірававаннем дынамічнай памяці

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int n, k, num = 0;

bool t = false;

cout << "Увядзіце памер масіва: ";

cin >> n >> k;

int\*\* a = new int\* [n]; // Робім масіў

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

a[i] = new int[k]; // Вылзяляем памяць для кожнага радка масіва

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

cin >> a[i][j];

}

}

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

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

if (a[i][j] > 0) { // Шукаем дадатны элемент

t = true; // Ставім сцяг, калі ёсць хаця б 1 дадатны лік

num = i; // Адзначаем першы радок, у якім ёсць дадатны элемент

cout << "У радку з індэксам " << i << " ёсць хаця б адзін дадатны лік" << endl;

break;

}

}

if (t == true) { // Калі знайшлі, перарываем цыкл

break;

}

}

if (t == false) { // Калі такіх элементаў не знайшлі

cout << "У ніводным радку няма дадатных лікаў" << endl;

}

else { // Калі зсё ж ёсць, працягваем выкананне задання

if (num == 0) {

num = n - 1;

}

else {

num -= 1;

}

for (int i = 0; i < k; i++) { // Змяняем у мінулам радку знакі, а для першага радка змяняем у апошняга

a[num][i] = -a[num][i];

}

cout << "Новы масіў:" << endl;

for (int i = 0; i < n; i++) { // Вывад новага масіва

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

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

}

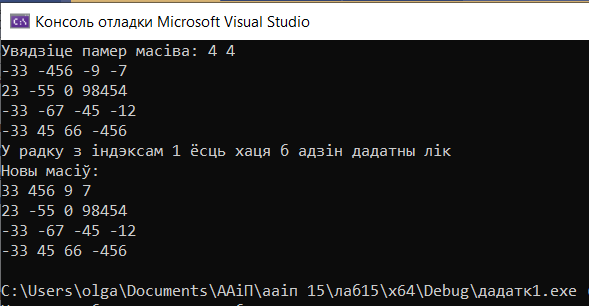
cout << endl;

}

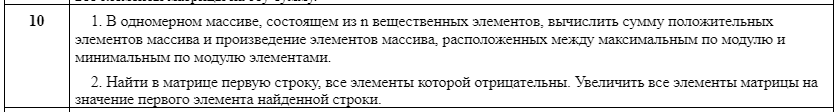
}

delete[] a; // Ачышчаем памяць, якую выдзялілі для двумернага масіва

}



Варыянт 10



Заданне 1

#include <iostream>

#include <windows.h>

#include <cstdlib> // Бібліятэка для выкарыстання функцый, звязаных з выдзяленнем і кірававаннем дынамічнай памяці

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int n;

cout << "Увядзіце колькасць элементаў масіва: " << endl;

cin >> n;

cout << "Увядзіце элементы масіва: " << endl;

double\* a = (double\*)malloc(n \* sizeof(double)); // Выдзяляем памяць для масіва з выкарыстаннем функцаі malloc

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

cin >> \*(a + i); // Уводзім элементы масіва

}

int maxIndex = 0;

int minIndex = 0;

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

if (abs(\*(a + i)) > abs(\*(a + maxIndex))) {

maxIndex = i; // Знаходзім індэкс максімальнага элемента

}

if (abs(\*(a + i)) < abs(\*(a + minIndex))) {

minIndex = i; // Знаходзім індэкс мінімальнага элемента

}

}

int sum = 0;

int product = 1;

int start, end;

if (maxIndex < minIndex) { // Параўноўваем дзве зменныя

start = maxIndex + 1;

end = minIndex;

}

else {

start = minIndex + 1;

end = maxIndex;

}

for (int i = start; i < end; i++) {

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

sum += \*(a + i); // Падлічваем суму дадатных элементаў

}

product \*= \*(a + i); // Падлічваем множанне элементаў паміж максімальным і мінімальным па модулю

}

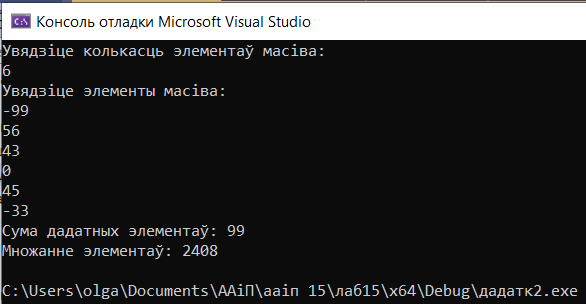
cout << "Сума дадатных элементаў: " << sum << endl;

cout << "Множанне элементаў: " << product << endl;

free(a); // Ачышчаем памяць з выкарыстаннем функцыі free

return 0;

}



Заданне 2

#include <iostream>

#include <windows.h>

#include <cstdlib> // Бібліятэка для выкарыстання функцый, звязаных з выдзяленнем і кірававаннем дынамічнай памяці

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int n, k, num = -1;

cout << "Увядзіце памер масіва: ";

cin >> n >> k;

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

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

a[i] = new int[k]; // Вылзяляем памяць для кожнага радка масіва

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

cin >> a[i][j];

}

}

bool foundNegative = false; // Зменная для пошуку адмоўнага ліка ў радку

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

foundNegative = false; // Збрасваем сцяг у бягучым радку

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

if (a[i][j] >= 0) {

foundNegative = true; // Ставім сцяг, калі знайшлі адмоўны лік

break;

}

}

if (!foundNegative) {

num = i; // Запамінаем індэкс радка без адноўных лікаў

break;

}

}

if (num == -1) {

cout << "У ніводным радку няма адмоўных лікаў" << endl;

}

else {

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

a[num][i] += a[num][0]; // Павялічанне усіх элементаў знлйдзеннага радка на значэнне яе першага элемента

}

cout << "Новы масіў:" << endl;

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

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

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

}

cout << endl;

}

}

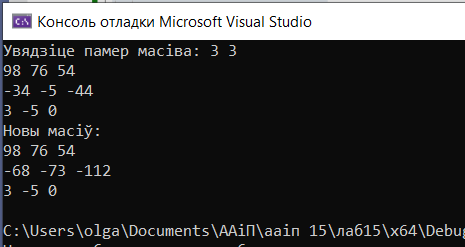
for (int i = 0; i < n; i++) {

delete[] a[i];

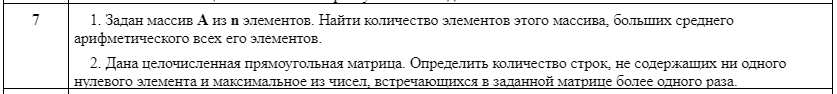
}

delete[] a; // Ачышчаем памяць, якую выдзялілі для двумернага масіва

}



Варыянт 7



Заданне 1

#include <iostream>

#include <windows.h>

#include <cstdlib> // Бібліятэка для выкарыстання функцый, звязаных з выдзяленнем і кірававаннем дынамічнай памяці

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int n, kol = 0;

cout << "Увядзіце колькасць элементаў масіва: " << endl;

cin >> n;

cout << "Увядзіце элементы масіва: " << endl;

int\* a = (int\*)malloc(n \* sizeof(int)); // Выдзяляем памяць для масіва з выкарыстаннем функцаі malloc

double sum = 0.0; // Зменная для захоўвання сумы ўсіх эдементаў масіва

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

cin >> \*(a + i);

sum += \*(a + i); // Дадаем бягучы элемент да сумы

}

double average = sum / n; // Падлік сярэдняга арфметычнага

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

if (\*(a + i) > average) { // Параўнозваем бягкчы элемент з сярэднім арфметычным

kol += 1; // Павялічваем лічыльнік, калі элемент большы за сярэдняе арыфметычнае

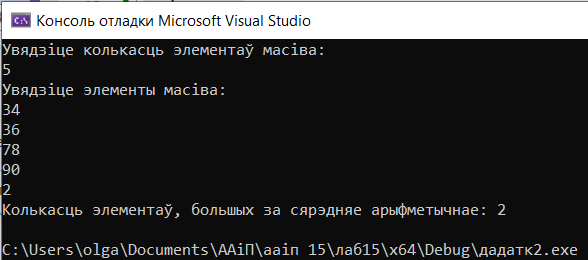
}

}

cout << "Колькасць элементаў, большых за сярэдняе арыфметычнае: " << kol << endl;

free(a); // Ачышчаем памяць з выкарыстаннем функцыі free

}



Заданне 2

#include <iostream>

#include <windows.h>

#include <cstdlib> // Бібліятэка для выкарыстання функцый, звязаных з выдзяленнем і кірававаннем дынамічнай памяці

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int n, k;

cout << "Увядзіце памер масіва: ";

cin >> n >> k;

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

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

a[i] = new int[k]; // Выдзяляем памяць для кожнага радка масіва

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

cin >> a[i][j];

}

}

int padlikRadkouBezNuleu = 0; // Лічыльнік радкоў без нулявых элементаў

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

bool znaishliNol = false; // Сцяг для пошуку нулявых элементаў у бягучым радку

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

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

znaishliNol = true; // Ставім сцяг, калі знайшлі нулявы элемент

break;

}

}

if (znaishliNol == false) {

padlikRadkouBezNuleu++; // Павялічваем лічыльнік, калі радок не мае нулявых элементаў

}

}

int maxCount = 0; // Максімальная колькасць паўтораў ліка

int maxNumber = a[0][0]; // Максімальны лік, які сустракаецца больш за раз

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

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

int BiaguchiNumder = a[i][j];

int count = 0; // Лічыльнік паўтораў ліка ў матрыцы

for (int x = 0; x < n; x++) {

for (int y = 0; y < k; y++) {

if (a[x][y] == BiaguchiNumder) {

count++; // Павялічваем лічыльнік, калі знайшлі паўтор ліка

}

}

}

if (count > 1 && count > maxCount) {

maxCount = count;

maxNumber = BiaguchiNumder;

}

}

}

cout << "Колькасць радкоў, якія не маюць нулявыя элементы: " << padlikRadkouBezNuleu << endl;

cout << "Максімальны лік, які сустракаецца больш за раз: " << maxNumber << endl;

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

delete[] a[i];

}

delete[] a; // Ачышчаем памяць, якую выдзялілі для двумернага масіва

}

