Беларускі дзяржаўны тэхналагічны ўніверсітэт

Факультэт інфармацыйных тэхналогій

Кафедра праграмнай інжэнерыі

Лабараторная работа №10

Па дысцыпліне «Асновы алгарытмізацыі і праграмавання»

На тэму «Аднамерныя масівы»

Выканала:

Студэнтка 1 курса 6 группы

Літвінчук Дар'я Валер'еўна

Выкладчык: асс. Андронава М.В.

2023, Мінск

**Варыянт 10**

**Заданне 5**

|  |  |
| --- | --- |
| 10 | 1. Найти в массиве элемент, если он существует, равный среднему арифметическому суммы трех его последних элементов. |

#include<iostream>

#include <ctime>//падключаем пры выкарыстанні функцыі time

int main()

{

setlocale(LC\_CTYPE, "Russian");

using namespace std;

const int N = 100;

int i, size, A[N], rmn = 0, rmx = 99;

int sum, x;

cout << "Увядзiце памер масiва: ";

cin >> size;//уводзім памер масіва

cout << "Масiў А: " << endl;

srand((unsigned)time(NULL));//каб лікі былі рознымі трэба выкарыстоўваць функцыю srand, якая задае пачатковы лік з дапамогай функцыі time, якая вяртае бягучы час у секундах

for (i = 0; i < size; i++)//запаўняем масіў выпадковымі лікамі ад 0 да 99

{

A[i] = (int)(((double)rand() / (double)RAND\_MAX) \* (rmx - rmn) + rmn);

cout << A[i] << " ";//выводзім элементы масіва

}

sum = (A[size - 1] + A[size - 2] + A[size - 3]) / 3;// вылiчаем сярэдняе арыфметычнае трох апошніх элементаў масіва

cout << "" << endl;

cout << "сярэдняе арыфметычнае трох апошнiх элементаў= " << sum << endl;

for (i = size; i > 0; i--)

{

if (A[i] == sum)//шукаем элемент роўны сярэдняму арыфметычнаму трох апошніх элементаў масіва

{

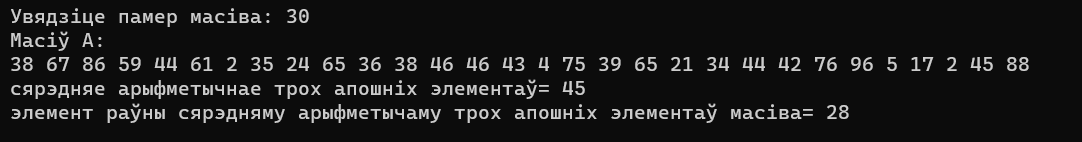
x = i;//прысвойваем нумар элемента

cout << "элемент раўны сярэдняму арыфметычаму трох апошнiх элементаў масiва= " << x << endl;

}

}

}

****

|  |  |
| --- | --- |
| 10 | 2. Задан массив из **k** чисел. Найти числа, входящие в массив только один раз. |

#include<iostream>

#include <ctime>//падключаем пры выкарыстанні функцыі time

int main()

{

setlocale(LC\_CTYPE, "Russian");

using namespace std;

const int N = 100;

int i, k, A[N], rmn = 0, rmx = 99, x;

cout << "Увядзiце памер масiва: ";

cin >> k;//уводзім памер масіва

cout << "Масiў А: " << endl;

srand((unsigned)time(NULL));//каб лікі былі рознымі трэба выкарыстоўваць функцыю srand, якая задае пачатковы лік з дапамогай функцыі time, якая вяртае бягучы час у секундах

for (i = 0; i < k; i++)//запаўняем масіў выпадковымі лікамі ад 0 да 99

{

A[i] = (int)(((double)rand() / (double)RAND\_MAX) \* (rmx - rmn) + rmn);

cout << A[i] << " ";//выводзім элементы масіва

}

cout << "" << endl;

cout << "" << endl;

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

{

int count = 0;

for (int j = 0; j < k; ++j)//правяраем каб лік масіва не паўтараўся

if (i != j && A[i] == A[j])

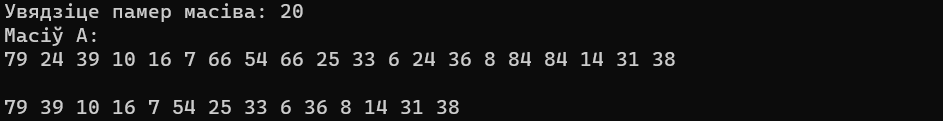
++count;

if (!count)

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

}

}

****

**Дадатковыя варыянты :**

**2. Подсчитать количество пар соседних элементов массива с одинаковыми значениями.**

#include<iostream>

#include<stdio.h>

int main()

{

setlocale(LC\_CTYPE, "Russian");

using namespace std;

const int N = 100;

int i, size, A[N], count = 0;

cout << "Увядзiце памер масiва: ";

cin >> size;//уводзім памер масіва

cout << "Масiў А: " << endl;

for (i = 0; i < size; i++)//запаўняем масіў з клавіятуры

{

cin >> A[i];

}

cout << " " << endl;

for (i = 0; i < size; i++)//правяраем кожны элемент масіва

{

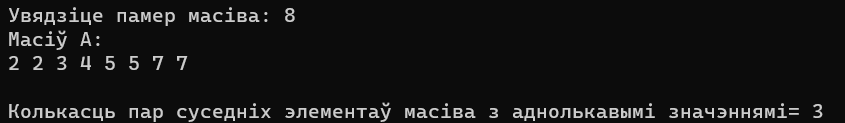
if (A[i] == A[i + 1])

count++;//калі значэнне элемента роўна значэнню суседняга элемента лічыльнік улічваем на 1

}

cout << "Колькасць пар суседнiх элементаў масiва з аднолькавымi значэннямi= " << count;

}



**3. Найти в массиве наибольшее число подряд идущих одинаковых элементов.**

#include<iostream>

#include<stdio.h>

int main()

{

setlocale(LC\_CTYPE, "Russian");

using namespace std;

const int N = 100;

int i, size, A[N], count = 0, max\_count = 0;

cout << "Увядзiце памер масiва: ";

cin >> size;//уводзім памер масіва

cout << "Масiў А: " << endl;

for (i = 0; i < size; i++)//запаўняем масіў з клавіятуры

{

cin >> A[i];

}

cout << " " << endl;

for (i = 0; i < size; i++)//правяраем кожны элемент масіва

{

if (A[i] == A[i + 1])

count++;//калі значэнне элемента роўна значэнню суседняга элемента лічыльнік улічваем на 1

else//інакш параўноўваем значэнні counter і max\_count

{

if (count > max\_count)

{

max\_count = count;

count = 0;

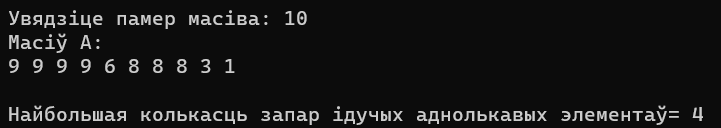
}

}

}

cout << "Найбольшая колькасць запар iдучых аднолькавых элементаў= " << ++max\_count;

}

****

**6. Найти непрерывную последовательность положительных чисел, сумма элементов в которой максимальная. Максимальный размер массива 100 элементов. Диапазон значений от -100 до 100.**

#include<iostream>

#include<stdio.h>

#include<ctime>

int main()

{

setlocale(LC\_CTYPE, "Russian");

using namespace std;

const int N = 100;

int A[N], k, sum = 0, mx = 0;

bool znak = false;//будзе паведамляць аб дадатных лiках

cout << "Увядзiце памер масiва: ";

cin >> k; cout << endl;

for (int i = 0; i < k; i++)//масіў запаўняецца выпадковымі лікамі ў дыяпазоне ад -100 да 100

{

A[i] = -100 + rand() % 200;

}

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

{

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

if (A[j] > 0)//калі элемент дадатны, то znak прысвойвае значэнне true

{

sum += A[j];

znak = true;

}

if (znak == true && A[j] < 0)

{

mx = sum;

sum = 0;

znak = false;

}

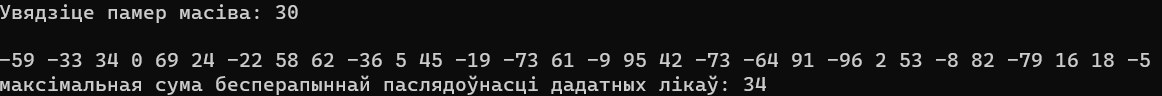
if (sum > mx) mx = sum;

}

cout << endl;

cout << "максiмальная сума бесперапыннай паслядоўнасцi дадатных лiкаў: " << mx;

}

****