







// ConsoleApplication79.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <iostream>

using namespace std;

//Функція, яка у випадку перевищення прожиткового мінімуму відрахує 20%

void podatok(double & zarplata, double prozMin)

{

if (zarplata > prozMin)

zarplata = zarplata - zarplata \* 20 / 100;

}

int main()

{

double zarplata;

cout << "Zarplata =";

cin >> zarplata;

podatok(zarplata, 1475);

cout << "ZArpata bez podatku=" << zarplata << endl;

system("pause");

return 0;

}

Приклад. Дано одновимірний масив. Знайти середнє значення додатніх елементів, якщо такі є.

// ConsoleApplication79.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <iostream>

using namespace std;

void vvod(double\*&a, int&n)

{

cout << "n=";

cin >> n;

a = new double[n];

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

{

printf("a[%d]=",i);

cin >> a[i];

}

}

void vuod(double\* a, int n)

{

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

{

printf("a[%d]=%f", i,a[i]);

}

}

bool getSrDod(double\* a, int n, double& serDod)

//Будемо повертати true якщо додатні є (а середнє будемо повертати через формальний параметр посилання)

{

int kDod = 0;

double sumDod = 0;

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

{

if (a[i] > 0)

{

sumDod += a[i];

kDod++;

}

}

if (kDod > 0)

{

serDod = sumDod / kDod;

return true;

}

else

return false;

}

int main()

{

double\* b;

int nb;

vvod(b, nb);

double sr;

if (getSrDod(b, nb, sr) == true)

{

cout << "sr=" << sr << endl;

}

else

cout << "dod nema" << endl;

system("pause");

return 0;

}

Дано масив знайти два найбільших елемент

// ConsoleApplication79.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <iostream>

using namespace std;

void vvod(double\*&a, int&n)

{

cout << "n=";

cin >> n;

a = new double[n];

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

{

printf("a[%d]=",i);

cin >> a[i];

}

}

void vuod(double\* a, int n)

{

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

{

printf("a[%d]=%f", i,a[i]);

}

}

bool getSrDod(double\* a, int n, double& serDod)

//Будемо повертати true якщо додатні є (а середнє будемо повертати через формальний параметр посилання)

{

int kDod = 0;

double sumDod = 0;

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

{

if (a[i] > 0)

{

sumDod += a[i];

kDod++;

}

}

if (kDod > 0)

{

serDod = sumDod / kDod;

return true;

}

else

return false;

}

//===================================

void get\_2\_max(double \* a, int n, double& max1, double&max2) //Будемо вважати, що max1<=max2

{

if (a[0] < a[1])

{

max1 = a[0];

max2 = a[1];

}

else

{

max1 = a[1];

max2 = a[0];

}

for (int i = 2; i < n; i++)

{

if (a[i] > max1)

{

if (a[i] > max2)

{

max1 = max2;

max2 = a[i];

}

else

max1 = a[i];

}

}

}

int main()

{

double\* b;

int nb;

vvod(b, nb);

double sr;

if (getSrDod(b, nb, sr) == true)

{

cout << "sr=" << sr << endl;

}

else

cout << "dod nema" << endl;

//------------------------------------

double m1,m2;

get\_2\_max(b, nb, m1, m2);

printf("max1=%f, max2=%f\n",m1,m2);

system("pause");

return 0;

}

