//Задача 4. Знайти мінімальну з/п, що більша за 3200

// ConsoleApplication56.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <iostream>

using namespace std;

int main()

{

int salariesCount;

printf("Salaries count =");

cin >> salariesCount;

double \* salaries = new double[salariesCount];

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

{

printf("Enter salary %d : ", i + 1);

cin >> salaries[i];

}

//------ І. Підхід з двома етапами

//--------- 1 етап : знаходимо першу з/п, яка більша за 3200

int firstSalary = -1;

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

{

if (salaries[i]>3200)

{

firstSalary = i;

break;

}

}

if (firstSalary==-1)

{

printf("No such");

}

else

{

//---------- 2 етап: порівнюємо з іншими, що також більші за 3200

double m = salaries[firstSalary];

for (int i = firstSalary+1; i < salariesCount; i++)

{

if (salaries[i] > 3200 && salaries[i]<m)

{

m = salaries[i];

}

}

printf("Min=%.2f", m);

}

//-----------ІІ. ПІДХІД-----------

double m = \_I32\_MAX; //Як початкове беремо дуже велике нереальне значення (+нескінченість)

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

{

if (salaries[i] > 3200 && salaries[i]<m)

{

m = salaries[i];

}

}

if (m==\_I32\_MAX)

{

printf("No");

}

else

printf("Min=%.2f", m);

delete[]salaries;

salaries = 0;

system("pause");

return 0;

}

//Задача 5. Знайти кількість заробітних плат, що дорівнюють максимальній з/п

#include "pch.h"

#include <iostream>

using namespace std;

int main()

{

int salariesCount;

printf("Salaries count =");

cin >> salariesCount;

double \* salaries = new double[salariesCount];

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

{

printf("Enter salary %d : ", i + 1);

cin >> salaries[i];

}

double m = salaries[0];

for (int i = 1; i < salariesCount;i++)

{

if (salaries[i] > m)

{

m = salaries[i];

}

}

int k = 0;

for (int i = 1; i < salariesCount;i++)

{

if (abs( m - salaries[i])<0.001 )

{

k++;

}

}

printf("Max count=%.2f",k);

delete[]salaries;

salaries = 0;

}

ПРИКЛАД. ДВОВИМІРНІ МАСИВИ

// ConsoleApplication56.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <iostream>

using namespace std;

int main()

{

//1. Вводимо кількість рядків і стовпців

int n; //Кількість рядків

printf("n=");

cin >> n;

int m; //Кількість стовпців

printf("m=");

cin >> m;

//2.Виділяєм пам"ять для масиву (нехай це будуть дійсні числа - double)

//2.1. Виділяємо пам"яь для масиву адрес рядків

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

//2.2. Виділяємо пам"ять для кожного окремого рядка

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

{

a[i] = new double[m]; //у a[i] зберігаємо адресу і-го рядка

}

//3. Вводимо елементи

for (int i = 0; i < n; i++) //Як змінюються номери рядків

{

for (int j = 0; j < m; j++) //Як змінюються номери стовпців у кожному рядку

{ //Що треба зробити з кожним елементи з і-го рядка та j-го стовпця

printf("Element row=%d, column=%d : ",i,j);

cin >> a[i][j];

}

}

// 4. Знайдемо суму усіх елементів

double sum = 0;

for (int i = 0; i < n; i++) //Як змінюються номери рядків

{

for (int j = 0; j < m; j++) //Як змінюються номери стовпців у кожному рядку

{ //Що треба зробити з кожним елементи з і-го рядка та j-го стовпця

sum+=a[i][j];

}

}

printf("Sum=%f",sum);

// 4. Знайдемо суму усіх елементів (перших 7 рядків)

sum = 0;

for (int i = 0; i < 7; i++) //Як змінюються номери рядків(0-6 (<7))

{

for (int j = 0; j < m; j++) //Як змінюються номери стовпців у кожному рядку

{ //Що треба зробити з кожним елементи з і-го рядка та j-го стовпця

sum += a[i][j];

}

}

printf("Sum=%f", sum);

// 4. Знайдемо суму усіх рядків від 3 до 8

sum = 0;

for (int i = 3; i <=8; i++) //Як змінюються номери рядків

{

for (int j = 0; j < m; j++) //Як змінюються номери стовпців у кожному рядку

{ //Що треба зробити з кожним елементи з і-го рядка та j-го стовпця

sum += a[i][j];

}

}

printf("Sum=%f", sum);

// 4. Знайдемо суму усіх парних рядків

sum = 0;

for (int i = 0; i <n; i+=2) //Як змінюються номери рядків

{

for (int j = 0; j < m; j++) //Як змінюються номери стовпців у кожному рядку

{ //Що треба зробити з кожним елементи з і-го рядка та j-го стовпця

sum += a[i][j];

}

}

printf("Sum=%f", sum);

// 4. Знайдемо суму усіх парних стовпців

sum = 0;

for (int i = 0; i <n; i ++) //Як змінюються номери рядків

{

for (int j = 0; j < m; j+=2) //Як змінюються номери стовпців у кожному рядку

{ //Що треба зробити з кожним елементи з і-го рядка та j-го стовпця

sum += a[i][j];

}

}

// 4. Знайдемо суму усіх непарних стовпців

sum = 0;

for (int i = 0; i <n; i++) //Як змінюються номери рядків

{

for (int j = 1; j < m; j += 2) //Як змінюються номери стовпців у кожному рядку

{ //Що треба зробити з кожним елементи з і-го рядка та j-го стовпця

sum += a[i][j];

}

}

printf("Sum=%f", sum);

// 4. Знайдемо суму усіх додатніх елементів, які знаходяться на перетині парних ряків і стовпців

sum = 0;

for (int i = 0; i <n; i+=2) //Як змінюються номери рядків

{

for (int j = 0; j < m; j += 2) //Як змінюються номери стовпців у кожному рядку

{ //Що треба зробити з кожним елементи з і-го рядка та j-го стовпця

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

sum += a[i][j];

}

}

printf("Sum=%f", sum);

// 4. Знайдемо суму для кожного рядка окремо

for (int i = 0; i <n; i++) //Як змінюються номери рядків

{ //Що зробити для кожного рядка

sum = 0;

for (int j = 0; j < m; j += 2) //Як змінюються номери стовпців у кожному рядку

{ //Що треба зробити з кожним елементи з і-го рядка та j-го стовпця

sum += a[i][j];

}

printf("Sum=%f", sum);

}

system("pause");

return 0;

}