Задача. Дано масив виручки магазина протягом N днів. Виконати одну із дій за бажанням користувача:

1) знайти загальний об’єм виручки;

2)знайти середній об’єм виручки;

3)знайти найбільший об’єм виручки;

=================================

4)вивести на екран номери тих днів, у які кількість виручки була більшою за середнє;

5)вивести на екран номери тих днів, у які кількість виручки була меншою за середнє;

6) вивести на екран номери тих днів, у які кількість виручки дорівнювала максимальному значенню виручки.

// ConsoleApplication64.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include<iostream>

using namespace std;

//1) знайти загальний об’єм виручки;

double getTotalIncome(double\* incomes, int daysCount)

{

double sum = 0;

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

{

sum += incomes[i];

}

return sum;

}

//2)знайти середній об’єм виручки;

double getAverageIncome(double\* incomes, int daysCount)

{

return getTotalIncome(incomes, daysCount) / daysCount;

}

//3)найти найбільший об’єм виручки;

double getMaxIncome(double\* incomes, int daysCount)

{

double max = incomes[0];

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

{

if (max<incomes[i])

{

max = incomes[i];

}

}

return max;

}

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

bool isGreate(double c1, double c2)

{

return c1 > c2;

}

bool isLower(double c1, double c2)

{

return c1 < c2;

}

bool isEqual(double c1, double c2)

{

return c1 == c2;

}

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

void printByRule(double\* incomes, int daysCount, bool(\*rule)(double, double), double value)

{

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

{

if (rule(incomes[i],value))

{

printf("%d : %f\n",i, incomes[i]);

}

}

}

int main()

{

double\* incomes = new double[4]{5000,3200,7000,8500};

printf("1) >average\n");

printf("2) <average\n");

printf("3) == max\n");

printf("Your answer :");

int answer;

cin >> answer;

double average = getAverageIncome(incomes, 4);

double max = getMaxIncome(incomes, 4);

switch (answer)

{

case 1:printByRule(incomes, 4, isGreate, average);

break;

case 2:printByRule(incomes, 4, isLower, average);

break;

case 3:printByRule(incomes, 4, isEqual, max);

break;

}

system("pause");

return 0;

}