// ConsoleApplication53.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <iostream>

#include <string>

using namespace std;

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

template<typename MyType1, typename MyType2>

MyType1 getMax(MyType1 a, MyType2 b)

{

if (a>b)

{

return a;

}

else

{

return b;

}

}

int main()

{

int max1 = getMax(23, 12);

double max2 = getMax<double, int>(23.0, 12);

max2 = getMax<double, int>(23.0, 12);

printf("Random int =%d \n", getRandomInt(3,12,2,5,8));

system("pause");

return 0;

}

Задача 1.1. Створити функцію, яка б могла знаходити суму або додатніх, або від’ємних чисел.

// ConsoleApplication53.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <iostream>

#include <string>

using namespace std;

bool isPositive(int number)

{

return number > 0;

}

bool isNegative(int number)

{

return number < 0;

}

bool isEven(int number)

{

return number % 2 == 0;

}

int sumByRule(int\* a, int n, bool (\*selectionRule)(int))

{

int s = 0;

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

{

if (selectionRule(a[i]))

{

s += a[i];

}

}

return s;

}

int main()

{

int\* arr = new int[] {2,-1,4,-6,23,11,-9};

int sumP = sumByRule(arr, 7, isPositive);

int sumN= sumByRule(arr, 7, isNegative);

int sumE = sumByRule(arr, 7, isEven);

system("pause");

return 0;

}

Задача 1. Створити функцію, яка б могла виконати одну із дій над одновимірним масивом:

1)до кожного елемента додати 2;

2)кожен елемент замінити сумою цифр;

3)кожен елемент помножити на 7.

Задача 2. Створити функцію, яка б виводила на екран усі числа, які задовольняють одному із правил:

1)парні;

2)числа, у яких остання цифра дорівнює 7;

4)числа, які більші за 20.

// ConsoleApplication53.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <iostream>

#include <string>

using namespace std;

bool isLastNumber\_7(int number)

{

return number%10==7;

}

bool isGreateThan\_20(int number)

{

return number>20;

}

bool isEven(int number)

{

return number % 2 == 0;

}

bool \_true(int number)

{

return true;

}

void printByRule(int\* a, int n, bool (\*selectionRule)(int))

{

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

{

if (selectionRule(a[i]))

{

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

}

}

cout << endl;

}

int main()

{

int\* arr = new int[7] {2,-1,4,-6,23,11,-9};

printf("1. >20\n");

printf("2. Even \n");

printf("3. Last digit is 7\n");

printf("4. All\n");

printf("Your choise: ");

int c;

cin >> c;

switch (c)

{

case 1:printByRule(arr, 7, isGreateThan\_20);

break;

case 2:printByRule(arr, 7, isEven);

break;

case 3:printByRule(arr, 7, isLastNumber\_7);

break;

case 4:printByRule(arr, 7, \_true);

break;

}

system("pause");

return 0;

}

Приклад. Нараховувати премію за одним з правил

// ConsoleApplication62.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <iostream>

#include <string>

using namespace std;

//----- Правила нарахування премії

//Правило 1 (20% якщо з/п <8000 15% інакше)

double rule1(double salary)

{

if (salary < 8000)

return salary\*0.2;

else

return salary\*0.15;

}

//Правило 2 (0% якщо з/п <1000 15% (1000-5000), 25% - інакше)

double rule2(double salary)

{

if (salary < 1000)

return 0;

if (salary < 5000)

return salary\*0.15;

return salary\*0.25;

}

//----- Узагальнена функція, для нарахування премії за заданим правилом

void setBonuses(double\* salaries, int length, double(\*rule)(double))

{

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

{

salaries[i] += rule(salaries[i]);

}

}

int main()

{

double\* salaries = new double[4]{5000,7500,4300,7800};

printf("What is the rule? (rule 1 or rule 2)");

int answer;

cin >> answer;

switch (answer)

{

case 1: setBonuses(salaries, 4, rule1);

break;

case 2: setBonuses(salaries, 4, rule2);

break;

}

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

for (int i = 0; i < 4; i++)

{

printf("Salary N%d %.2f\n",i,salaries[i]);

}

system("pause");

return 0;

}