Задача. За допомогою покажчиків на функції реалізувати обчислення виразу

Де f або функція знаходження добутку чисел, або функція знаходження суми двох чисел.

Задача 1. Створити функцію, яка б могла для масиву знаходити одну із сум:

1)суму додатних;

2)суму парних;

3)суму елементів, кратних 3.

// ConsoleApplication24.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include<iostream>

using namespace std;

//Щоб не писати дві такі функції (наведені нижче)

int getSumPositive(int\*a, int length) //Сума додатніх

{

int s = 0;

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

{

if (a[i] > 0)

s += a[i];

}

return s;

}

int getSumEven(int\*a, int length) //Сума парних

{

int s = 0;

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

{

if (a[i] %2 == 0)

s += a[i];

}

return s;

}

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

//--- Правила перевірки

bool isPositive(int number)

{

return number > 0;

}

bool isEven(int number)

{

return number %2 == 0;

}

bool isDividedBy3(int number)

{

return number % 3 == 0;

}

//---- Узагальнена функція на основі правил

int getSumByRule(int\*a, int length, bool (\*rule)(int))

{

int s = 0;

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

{

if (rule(a[i]))

s += a[i];

}

return s;

}

int main()

{

int\* a = new int[4]{32,1,-61,9};

int choise;

printf("1. Positive \n");

printf("2. Even \n");

printf("3. Divided by 3\n");

cin >> choise;

switch (choise)

{

case 1: printf("Sum positive =%d", getSumByRule(a,4,isPositive));

break;

case 2: printf("Sum even =%d", getSumByRule(a, 4, isEven));

break;

case 3: printf("Sum 3 =%d", getSumByRule(a, 4, isDividedBy3));

break;

default:

break;

}

return 0;

}

Задача 2. 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.