**Федеральное государственное автономное образовательное учреждение**

**высшего образования**

**"Национальный исследовательский университет**

**"Высшая школа экономики"**

Московский институт электроники и математики НИУ ВШЭ  
Департамент компьютерной инженерии

**Курс: Информатика и программирование**

**ОТЧЕТ**  
**по лабораторной работе № 6**

|  |  |  |  |
| --- | --- | --- | --- |
| **Раздел** | **Max оценка** | **Итог.**  **Оценка**  **1** | **Итог.**  **Оценка**  **2** |
| Тесты | 1 |  |  |
| Работа программы | 1 |  |  |
| Алгоритм | 3 |  |  |
| Вопросы | 2 |  |  |
| Доп. Задание | 3 |  |  |

**Студент: Жалкова Наталия Евгеньевна**  
**Группа: БИВ153, 1 подгруппа**   
**Вариант: 76**  
**Руководитель: Хруслова Диана Владимировна  
Оценка:**

**МОСКВА 2016**

# Содержание

[Содержание 1](#_Toc451802533)

[Условия задач 2](#_Toc451802534)

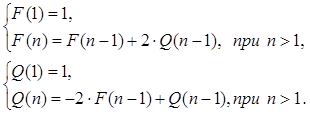
[Распечатка программы 3](#_Toc451802535)

[Тестовые примеры 7](#_Toc451802536)

# Условия задач

I. Написать программу, включающую рекурсивную функцию для вычисления:

Алгоритмы вычисления значений функции F(n) и Q(n), где n – натуральное число, заданы следующими соотношениями:

****

II. Написать программу, которая создает бинарное дерево, состоящее из целых чисел, которые вводятся с клавиатуры. Затем программа выполняет преобразования в соответствии с номером варианта и печатает все числа исходного и преобразованного бинарного дерева (или другой результат работы программы).

**Находит сумму всех нечетных элементов дерева.**

# Распечатка программы

**1 программа**

#include <stdio.h>

#include <string.h>

#include <cstdlib> //для atoi

#include <ctype.h> //для isdigit

void proverka(int \*prov){

int k;

char sprov[20],\*s;

do

{ k=0; //количество цифр

gets(sprov); //ввод строки

s=sprov; //указатель на строку

while(\*s) //пока существует символ

{ if (isdigit(\*s)) //если это цифра, счетчик увеличивается

k++;

s++;//идем дальше

}

if (k!=strlen(sprov)) //если количество цифр не равно длине строки

printf("Enter positive integer\r\n");

}

while (k!=strlen(sprov)); //пока количество цифр не равно длине строки вводим строку снова и снова

\*prov=atoi(sprov); //вышли из цикла, ввод верен, привели строку к типу int

}

int F(int n); //прототип

int Q(int n); //прототип

int main (){

int n;

do

{ printf("\r\nEnter natural n: \r\n");

proverka(&n);

}

while (n<1);

printf("F(%d) = %d\r\n",n,F(n));

printf("Q(%d) = %d\r\n",n,Q(n));

return 0;

}

int F(int n){

int sum;

if (n==1)

return (1);

else

{ sum = F(n-1) + 2\*Q(n-1);

return(sum);

}

}

int Q(int n){

int sum;

if (n==1)

return (1);

else

{ sum = (-2)\*F(n-1)+Q(n-1);

return(sum);

}

}

**2 программа**

#include <stdio.h>

struct node

{ int info;

node \*left, \*right;

};

//новый узел с числом p

node \*newn(int p)

{

node \*uk = new node;

uk->info = p;

uk->left = uk->right = NULL;

return(uk);

}

//присоединение нового узла c числом x к узлу p слева

void setleft(node \*p, int x)

{ p->left=newn(x);

}

//присоединение нового узла c числом x к узлу p справа

void setright(node \*p, int x)

{ p->right=newn(x);

}

node \*form()

{ int n;

node \*der = NULL, \*next, \*tek;

printf("Input integer numbers\nletter means that input is over");

if (scanf("%d",&n)==1)

{ der = newn(n);

while (scanf("%d",&n)==1)

{ next = tek = der;

while (next!=NULL)

{ tek = next;

if (n<tek->info)

next = tek->left;

else

next = tek->right;

}

if(n<tek->info)

setleft(tek,n);

else

setright(tek,n);

}

}

return der;

}

//вывод дерева на экран

void print\_tree(node \*der, int h)

{ if(der)

{ print\_tree(der->left, h+1);

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

printf(" ");

printf("%d\r\n",der->info);

print\_tree(der->right, h+1);

}

}

int summary(node \*der)

{

if (der == NULL) return 0;

if (der != NULL)

{

if (der->info%2==1)

return (summary(der->right) + summary(der->left) + der->info);

else

return (summary(der->right) + summary(der->left));

}

}

int main()

{node \*der;

printf("Building tree\n");

der = form();

if (!der)

printf("Tree is empty");

else

{ print\_tree(der, 0);

printf("Sum of tree's odd elements = %d", summary(der));

}

return 0;

}

# Тестовые примеры

|  |  |  |
| --- | --- | --- |
| **№** | **Исходные данные** | **Результат** |
| **1** | N=3 | F(3)=1; Q(3)=-7 |
| **2** | x | Tree is empty |
| **3** | 1 2 8 7 3 r | Sum=11 |
| **4** | 2 x | Sum=0 |