ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИСCЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Факультет безопасности информационных технологий

**Дисциплина:**

«Программирование»

**ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1**

**Вариант 5-16**

**Выполнил:**

Студент гр. N3151, Мочеков Семён Сергеевич



**Проверил:**

Гирик Алексей Валерьевич

Санкт-Петербург

2023

1. **Задание.**

Для числа типа uint16\_t, используя арифметические, логические и побитовые операции произвести преобразование:

Назовём рангом тетрады модуль разности количества нулевых и единичных битов в ней. Сделать тетрады с самым высоким рангом старшими в числе, сохранив относительный порядок остальных тетрад.

1. **Makefile**

.PHONY: all clean

APP=lab1ssmN3151

CFLAGS=-Wall -Wextra -Werror -g

all:

gcc -o $(APP) $(CFLAGS) $(APP).c

clean:

rm $(APP)

1. **Работа алгоритма**

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

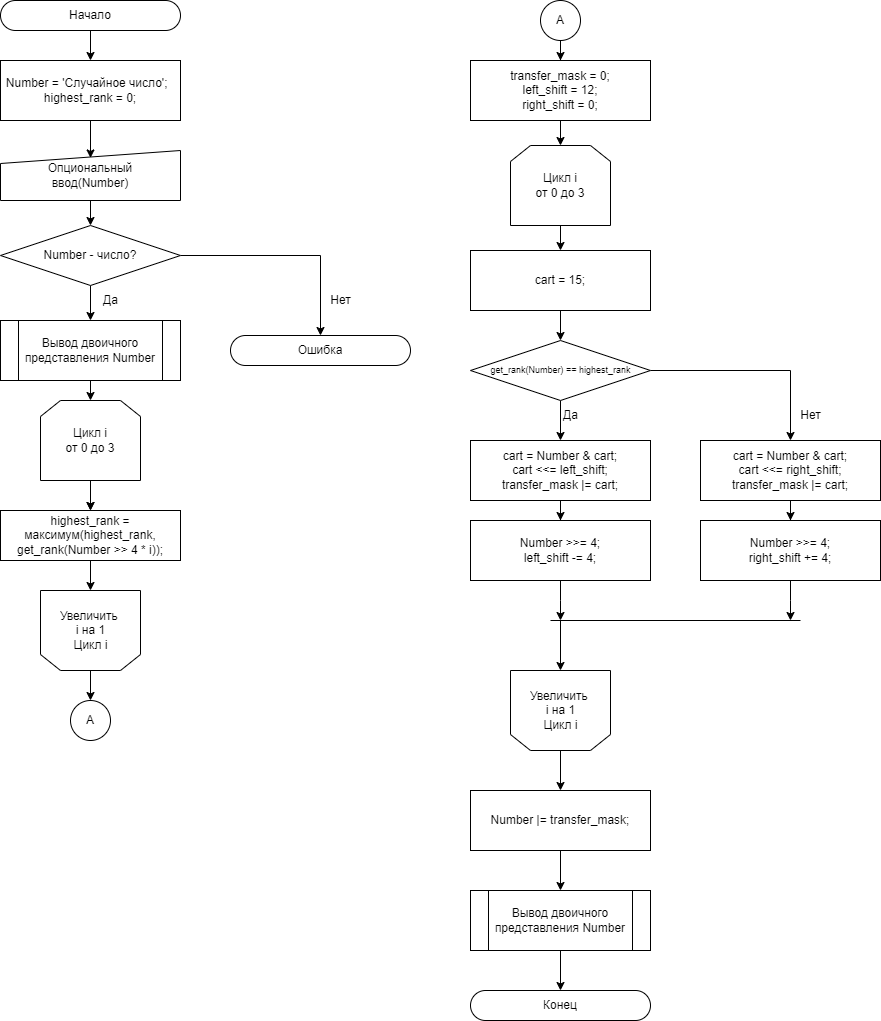
Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

**IV. Блок-схема алгоритма**



**V. Исходный код алгоритма**

#include <stdio.h>

#include <inttypes.h>

#include <stdlib.h>

#include <ctype.h>

#include <time.h>

#define Num\_bits uint16\_t

#define Num\_spicif "%" SCNu16

void binary\_output(Num\_bits num, Num\_bits kolvo) {

// функция двоичного вывода

Num\_bits outer = 1; outer <<= (sizeof(Num\_bits) \* 8 - 1);

for (Num\_bits i = 1; i <= kolvo \* 4; ++i) {

printf("%d", ((num & outer) != 0));

outer >>= 1;

if (i % 8 == 0){

printf(" ");

}

}

printf("\n");

}

Num\_bits get\_rank(Num\_bits nun) {

// функция получения ранга тетрады

Num\_bits result = 0, chuker = 1;

for (Num\_bits j = 0; j < 4; ++j) {

result += ((nun & (chuker)) != 0);

chuker <<= 1;

}

switch (result) {

case 4:

case 0:

return 4;

case 3:

case 1:

return 2;

default:

return 0;

}

}

int main(int argc, char \*argv[]){

srand(time(NULL));

Num\_bits Number = rand(), kolvo\_tetrad = sizeof(Num\_bits) \* 2, highest\_rank = 0,\

transfer\_mask = 0, left\_shift = (kolvo\_tetrad \* 4) - 4, right\_shift = 0,

const\_tetrada\_shift = 4, cart = 0xf, flag = 0;

// проверка ввода пользователя

if (argc == 2) {

if (sscanf(argv[1], Num\_spicif, &Number) == 0){

printf("Ошибка: \'%s\' не является числом\n", argv[1]);

return 0;

} else {

char \*vvod = argv[1];

Num\_bits j = 0;

while (isalnum(vvod[j])){

if (isdigit(vvod[j]) == 0){ printf("Ошибка: \'%s\' \

не является числом\n", argv[1]); return 0;}

j++;

}

}

}

// двоичный вывод

binary\_output(Number, kolvo\_tetrad);

// ранг тетрады определим подсчётом единиц. Если единиц 4/0 -> 4, если 3/1 -> 2,

если 2 -> 0

for (Num\_bits i = 0; i < kolvo\_tetrad; ++i) {

highest\_rank = (highest\_rank < (get\_rank(Number >> 4 \* i))) ? \

(get\_rank(Number >> 4 \* i)) : highest\_rank;

}

// в случае если все тетрады highest\_rank – вернём исходное число

for (Num\_bits j = 0; j < 4; ++j){

flag += ((get\_rank(Number >> 4 \* j)) == highest\_rank);

}

if (flag == kolvo\_tetrad) {

binary\_output(Number, kolvo\_tetrad);

return 0;

}

// узнав highest\_rank, перетасуем тетрады в необходимом порядке

for (Num\_bits i = 0; i < kolvo\_tetrad; ++i) {

cart = 0xf;

if (get\_rank(Number) == highest\_rank) {

cart = Number & cart;

cart <<= left\_shift;

transfer\_mask |= cart;

Number >>= const\_tetrada\_shift;

left\_shift -= const\_tetrada\_shift;

} else {

cart = Number & cart;

cart <<= right\_shift;

transfer\_mask |= cart;

Number >>= const\_tetrada\_shift;

right\_shift += const\_tetrada\_shift;

}

}

Number |= transfer\_mask;

// двоичный вывод

binary\_output(Number, kolvo\_tetrad);

return 0;

}