



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
КАФЕДРА МЕДИЧНОЇ КІБЕРНЕТИКИ ТА ТЕЛЕМЕДИЦИНИ

Лабораторна робота № 8

з дисципліни «Алгоритмізація та програмування»

на тему: «Функції користувача»

Варіант 16

Виконав:

Студент групи БС-32

Хоміщевич Микита

Перевірила:

викл.

Матвійчук А.О.

Київ-2014

Задание

16	Написати функцію, яка для заданого цілого числа x обчислює кількість пар (розташованих один за одним) нульових бітів в цьому числі.
----	---

☐ Лабораторна робота виконана без зауважень

☐ Лабораторна робота має зауваження:

☐ присутні зауваження до блок-схем:

☐ виконані не за стандартом

☐ не відповідають коду

☐ присутній код

☐ інші зауваження:

☐ присутні зауваження до коду:

☐ задача завдання вирішена хибно

☐ код програми не компілюється

☐ використано глобальні змінні

☐ недостатня декомпозиція на функціонуванні користувача

☐ статичні змінні при роботі з масивами

☐ оформлення коду

☐ присутні зайві символи «{» та «}»

☐ інші зауваження:

☐ невірні відповіді на запитання:

☐ №1 ☐ №2 ☐ №3

☐ мають інші зауваження:

Программный код

```
#include <iostream>

#include <conio.h>

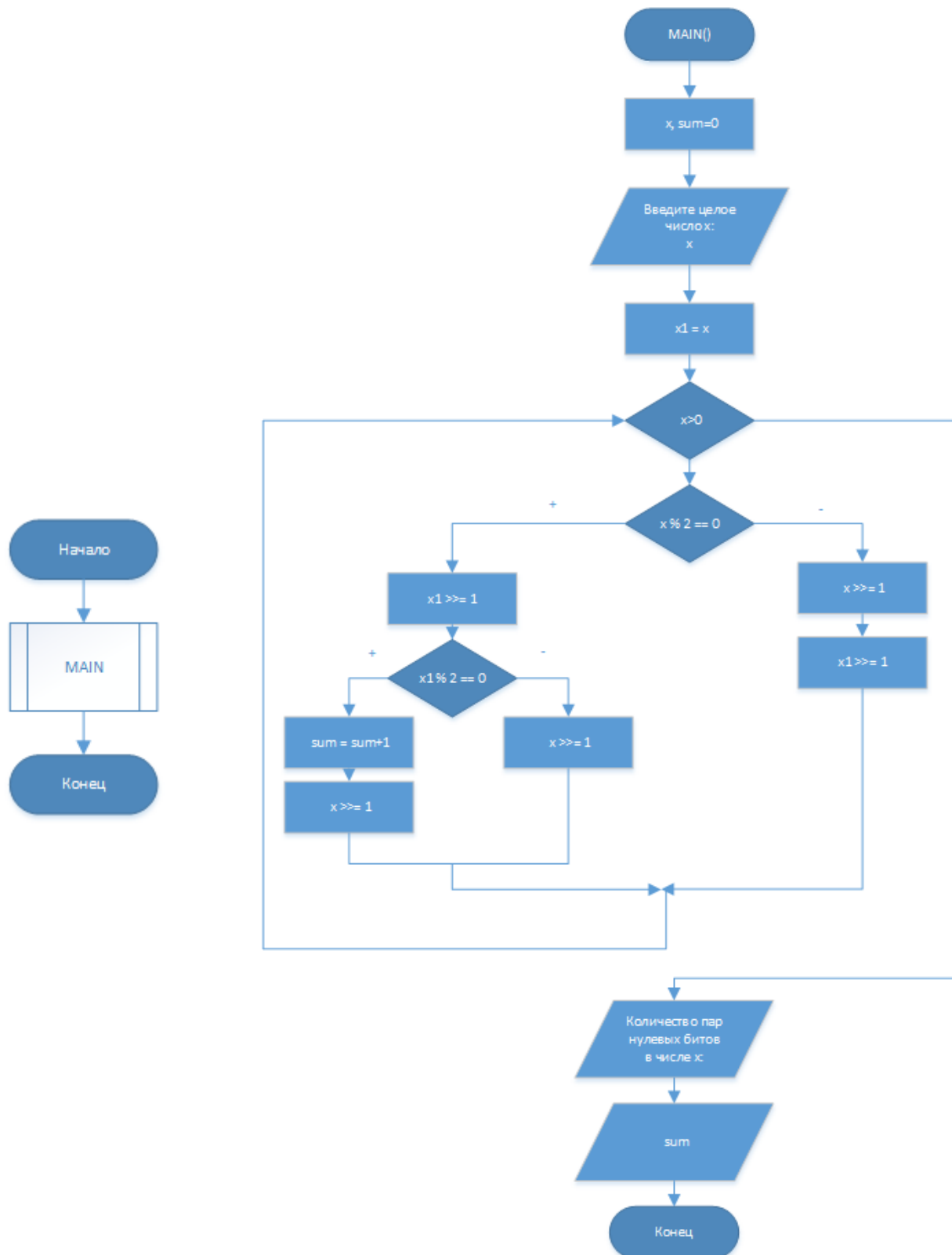
#include <locale.h>

using namespace std;

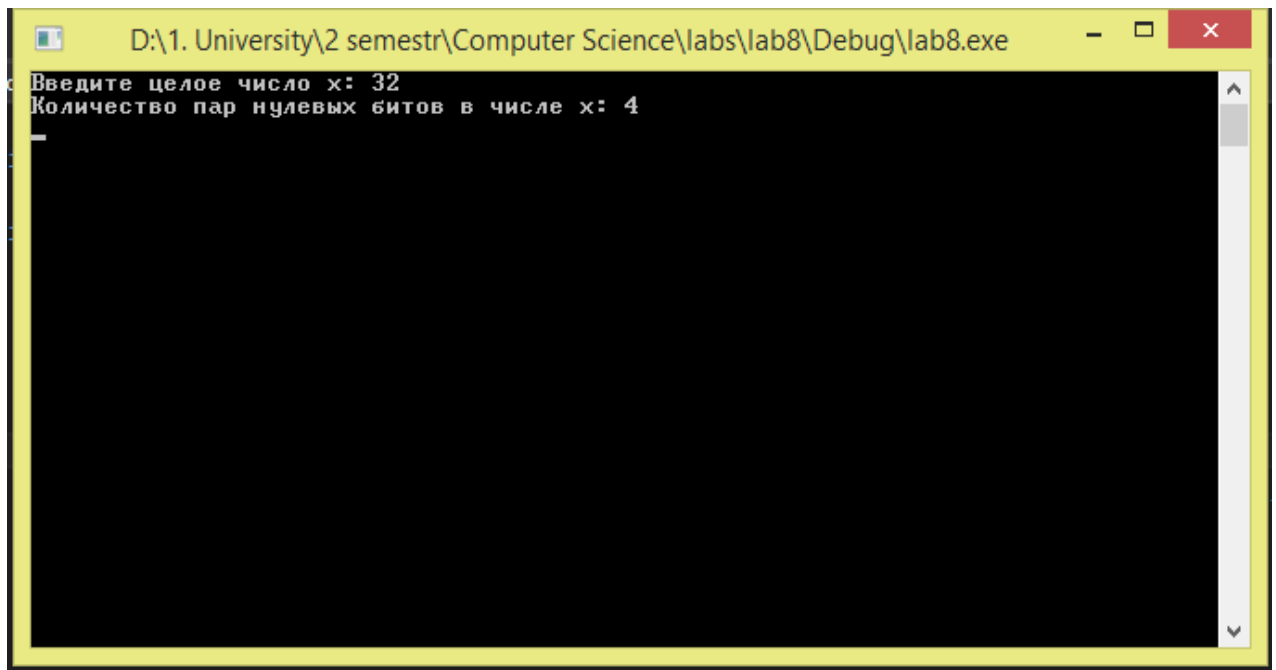
void MAIN() {
    setlocale(LC_ALL, "RUS");
    int x, sum = 0;
    cout << "Введите целое число x: ";
    cin >> x;
    int x1 = x;
    while (x > 0) {
        if (x % 2 == 0) { x1 >>= 1; if (x1 % 2 == 0) { ++sum; x >>= 1; } else { x >>=
1; } }
        else { x >>= 1; x1 >>= 1; }
    }
    cout << "Количество пар нулевых битов в числе x: " << sum << endl;
}

int main(void) {
    MAIN();
    _getch();
}
```

Блок-схема



Скриншот работы программы



Ответы на контрольные вопросы

1. Что такое прототип (сигнатура) функции, зачем он нужен?

Прототип функции (сигнатура) – это объявление функции, что не содержит тела функции, но указывает имя функции, аргументы, а так же типы аргументов и возвращаемый тип данных.

В основном сигнатуры используются программистами для проектирования функции «на будущее» - как описание интерфейса.

Пример:

```
int function(int n);
```

2. Чем отличается определение функции от объявления ее прототипа?

Когда мы создаем прототип функции, мы не описываем саму ф-ию, т.е. не создаем тело функции, где идет описание работы данной функции. Так же в прототипе функции мы можем не указывать имена переменных, которые будут использоваться, а можем лишь указать тип используемых параметров.

3. Может ли в одной ф-ии встречаться несколько операторов return?

Да. Может. К примеру, когда мы используем рекурсивную ф-ию и через условие возвращаем разные значения.

4. Сколько значений может вернуть ф-ия?

Одно.

5. Как следует организовывать ф-ии, если нужно вернуть значения нескольких переменных в точку вызова этой ф-ии?

Через стандартный способ возврата значения из функции в языке Си никак, но есть возможность использовать (в языке C++) передачу аргументов в ф-ию через ссылки. Тогда мы можем негласно вернуть сколько угодно значений.

Пример:

```
#include <iostream>

void lal(int &a, int &b) {
    a *= 100; b *= 200; }

int main() {
    int a, b;

    std::cin >> a >> b; //a = 12, b = 20
```

```
lal(a, b);

std::cout << a << " " << b << std::endl; // вернет a = 1200, b = 4000;

}
```

6. Что такое формальные параметры функции?

Это параметры, имена которых используются в теле функции.

7. Что такое фактические параметры функции?

Это параметры, которые передаются в функцию при ее вызове.

8. Какие особенности передачи параметров в функцию в языке C++?

Если, к примеру, мы передаем в функцию какие-то фактические параметры в виде обычной переменной, то далее, изменяя в функции уже данный фактический параметр мы при выходе из функции не поменяем значение фактического параметра.

А если передадим указатель на данную переменную (или ссылку), то данное значение можем изменять в функции, и соответственно, менять вне функции

9. Какие есть способы вызова функции?

- 1) Вызов функции через написание имя функции и в скобках указание параметров ф-ии
- 2) Рекурсивно
- 3) Через указатель на функцию
- 4) Вызов ф-ии в отдельном потоке

10. Значения какого типа возвращает ф-ия, если тип не указан?

Т.к. Си (и с++) язык со статической типизацией, то при вызове ф-ии обязательно должен быть указан тип возвращаемого значения. Но если, к примеру, мы вызываем функцию через указатель, то можно воспользоваться безтиповым (универсальным) значением “**void***”. Тогда тип возвращаемого значения будет определен по типу указателя.