

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ
УНИВЕРСИТЕТ им. В. Г. Шухова»
(БГТУ им. В. Г. Шухова)**

Кафедра программного обеспечения вычислительной
техники и автоматизированных систем

Лабораторная работа №19.7
по дисциплине: «Побитовые операции»

Выполнил/а: ст. группы ВТ-231
Кисиль Николай Владимирович

Проверили:
Черников Сергей Викторович
Новожен Никита Викторович

Белгород, 2023 г.

Цель работы:

получение навыков работы с побитовыми операциями

Содержание работы

Задача 1: Вывести восьмеричное представление записи числа x	4
Задача 2: Напишите функцию <i>deleteOctNumber</i> , которая удаляет цифру <i>digit</i> в записи данного восьмеричного числа x . Вывод результата можно произвести в любой системе счисления	5
Задача 3: Напишите функцию <i>swapP airBites</i> , которая меняет местами соседние цифры пар в двоичной записи данного натурального числа. Обмен начинается с младших разрядов. Непарная старшая цифра остается без изменения.	6
Задача 4: Напишите функцию <i>invertHex</i> , которая преобразует число x , переставляя в обратном порядке цифры в шестнадцатеричном представлении данного натурального числа.	7
Задача 5: Напишите функцию <i>isBinPoly</i> , которая возвращает значение 'истина', если число x является палиндромом в двоичном представлении, иначе - 'ложь'	8
Задача 6: Даны два двухбайтовых целых $sh1$ и $sh2$. Получить целое число, последовательность четных битов которого представляет собой значение $sh1$, а последовательность нечетных – значение $sh2$	9
Задача 7: Определить максимальную длину последовательности подряд идущих битов, равных единице в двоичном представлении данного целого числа.	10
Задача 8: ** Выполнить циклический сдвиг в двоичном представлении данного натурального числа x на k битов влево.	11
Задача 9: ** Дано длинное целое неотрицательное число. Получить число, удалив каждую вторую цифру в двоичной записи данного числа, начиная со старших цифр	12

Задача 10: ** Дано целое неотрицательное число. Получить число перестановкой битов каждого байта данного числа в обратном порядке. 13

Задача 11: ** Пакеты с монетами (1037A)..... 14

Задача 1: Вывести восьмеричное представление записи числа X

Пример тестовых данных:

№	Входные данные	Выходные данные
1	12	14
2	166	246
3	0	0

Спецификация функции countOctDigits:

1. Заголовок `int countOctDigits(unsigned x)`.
2. Назначение: возвращает число цифр в восьмеричное записи x

Спецификация функции printOct:

1. Заголовок `void printOct(const unsigned x)`
2. Назначение: выводит восьмеричное представление записи числа x

Код:

```
int countOctDigits(unsigned x) {
    int digits = 1;

    while (x > 7) {
        digits++;
        x >>= 3;
    }
    return digits;
}

void printOct(const unsigned x) {
    int digits = countOctDigits(x);
    for (size_t i = digits; i > 0; i--) {
        int digit = x >> BITS_IN_OCT_DIGIT * (i - 1) & 7;
        printf("%d", digit);
    }
}
```

Задача 2: Напишите функцию *deleteOctNumber*, которая удаляет цифру *digit* в записи данного восьмеричного числа *x*. Вывод результата можно произвести в любой системе счисления

Пример тестовых данных:

№	Входные данные	Выходные данные
1	3179 1	653
2	9 1	0
3	37 1	45

Спецификация функции *deleteOctNumber*:

1. Заголовок `deleteOctNumber(const unsigned x, const int d)`
2. Назначение: удаляет цифру *digit* в записи данного восьмеричного числа *x*.

Код:

```
void deleteOctNumber(const unsigned x, const int d) {
    int digits = countOctDigits(x);
    for (size_t i = digits; i > 0; i--) {
        int digit = x >> BITS_IN_OCT_DIGIT * (i - 1) & 7;
        if (digit != d) {
            printf("%d", digit);
        }
    }
}
```

Задача 3: Напишите функцию *swapPairBites*, которая меняет местами соседние цифры пар в двоичной записи данного натурального числа. Обмен начинается с младших разрядов. Непарная старшая цифра остается без изменения.

Пример тестовых данных:

№	Входные данные	Выходные данные
1	6	5
2	5	6
3	77	78

Спецификация функции *swapPairBites*:

1. Заголовок `unsigned int swapPairBites(int x)`
2. Назначение: меняет местами соседние цифры пар в двоичной записи данного натурального числа

Код:

```
unsigned int swapPairBites(int x) {
    int length = 0;
    int copy_number = x;
    while (copy_number > 0) {
        copy_number >>= 1;
        length++;
    }
    unsigned result;
    if (length % 2 == 0) {
        result = ((x & ODD_BIN_MASK) << 1) | ((x & EVEN_BIN_MASK) >> 1);
    }
    else{
        x = x - (1 << (length - 1));
        result = (((x & ODD_BIN_MASK) << 1) | ((x & EVEN_BIN_MASK) >> 1)) +
(1 << (length - 1));
    }
    return result;
}
```

Задача 4: Напишите функцию *invertHex*, которая преобразует число *x*, переставляя в обратном порядке цифры в шестнадцатеричном представлении данного натурального числа.

Пример тестовых данных:

№	Входные данные	Выходные данные
1	77	212
2	2732	3242

Спецификация функции *invertHex*:

1. Заголовок `long long invertHex(int n)`
2. Назначение: преобразует число *x*, переставляя в обратном порядке цифры в шестнадцатеричном представлении данного натурального числа.

Код:

```
long long invertHex(int n) {  
    long long res = 0;  
    while (n) {  
        res <<= 4;  
        res |= n & 0xF;  
        n >>= 4;  
    }  
    return res;  
}
```

Задача 5: Напишите функцию *isBinPoly*, которая возвращает значение 'истина', если число *x* является палиндромом в двоичном представлении, иначе - 'ложь'.

Пример тестовых данных:

№	Входные данные	Выходные данные
1	27	1
2	454	0

Спецификация функции *isBinPoly*:

1. Заголовок `int isBinPoly(unsigned int x)`
2. Назначение: возвращает значение 'истина', если число *x* является палиндромом в двоичном представлении, иначе - 'ложь'

Код:

```
int isBinPoly(unsigned int x) {
    unsigned int bin_x = x;
    unsigned int reversed_bin_x = 0;
    while (bin_x != 0) {
        unsigned int last_digit = bin_x & 1;
        reversed_bin_x <<= 1;
        reversed_bin_x |= last_digit;
        bin_x >>= 1;
    }
    return x == reversed_bin_x;
}
```


Задача 6: Даны два двухбайтовых целых *sh1* и *sh2*. Получить целое число, последовательность четных битов которого представляет собой значение *sh1*, а последовательность нечетных – значение *sh2*.

Пример тестовых данных:

№	Входные данные	Выходные данные
1	12 18	420
2	32512 0	715784192

Спецификация функции createNum:

1. Заголовок `int createNum(int sh1, int sh2)`
2. Назначение: возвращает целое число, последовательность четных битов которого представляет собой значение *sh1*, а последовательность нечетных – значение *sh2*.

Код:

```
int createNum(int sh1, int sh2){
    int new_num = 0;
    int index = 0;
    while (sh1 > 0 || sh2 > 0){
        new_num += (sh2 & 1) << index++;
        sh2 >>= 1;
        new_num += (sh1 & 1) << index++;
        sh1 >>= 1;
    }
    return new_num;
}
```

Задача 7: Определить максимальную длину последовательности подряд идущих битов, равных единице в двоичном представлении данного целого числа.

Пример тестовых данных:

№	Входные данные	Выходные данные
1	61454	4
2	11	2

Спецификация функции `getMaxLengthBits`:

1. Заголовок `unsigned int getMaxLength(unsigned int x)`
2. Назначение: возвращает максимальную длину последовательности подряд идущих битов, равных единице в двоичном представлении данного целого числа

Код:

```
unsigned int getMaxLengthBits(unsigned int x) {
    int curr_length = 0;
    int max_length = 0;

    while (x >= 1) {
        unsigned int digit = x & 1;
        if (digit == 1) {
            curr_length++;
            if (curr_length > max_length)
                max_length = curr_length;
        } else
            curr_length = 0;
        x >>= 1;
    }

    return max_length;
}
```

Задача 8: ** Выполнить циклический сдвиг в двоичном представлении данного натурального числа x на k битов влево.

Пример тестовых данных:

№	Входные данные	Выходные данные
1	27 1	23
2	27 2	15
3	42 1	21

Спецификация функции shift:

1. Заголовок `unsigned int shift(unsigned int x, unsigned int k)`.
2. Назначение: циклический сдвиг в двоичном представлении данного натурального числа x на k битов влево

Код:

```
unsigned int shift(unsigned int x, unsigned int k) {  
    int digitsCounter = ceil(log2(x + 1));  
  
    for (; k > 0; k--) {  
        x = (x << 1) | (x >> (digitsCounter - 1));  
        x &= ~(1 << digitsCounter);  
    }  
    return x;  
}
```

Задача 9: ** Дано длинное целое неотрицательное число. Получить число, удалив каждую вторую цифру в двоичной записи данного числа, начиная со старших цифр

Пример тестовых данных:

№	Входные данные	Выходные данные
1	3	1
2	10	3
3	40	6

Спецификация функции `deleteEverySecondDigit`:

1. Заголовок `unsigned long long deleteEverySecondDigit(unsigned long long x)`
2. Назначение: возвращает число, удалив каждую вторую цифру в двоичной записи данного числа, начиная со старших цифр

Код:

```
unsigned long long deleteEverySecondDigit(unsigned long long x) {
    unsigned long long result;
    if (x == 1 || x == 2)
        result = x >> (x - 1);
    else {
        int digits_count = ceil(log2(x + 1));
        result = x >> (digits_count - 1);
        for (int i = digits_count - 3; i >= 0; i -= 2) {
            result <<= 1;
            result |= ((x >> i) & 1);
        }
    }

    return result;
}
```

Задача 10: ** Дано целое неотрицательное число. Получить число перестановкой битов каждого байта данного числа в обратном порядке.

Пример тестовых данных:

№	Входные данные	Выходные данные
1	61455	4080
2	43605	21930

Спецификация функции `numberWithPermutation`:

1. Заголовок `unsigned long long numberWithPermutation(unsigned long long x)`
2. Назначение: возвращает число перестановкой битов каждого байта данного числа в обратном порядке

Код:

```
unsigned long long numberWithPermutation(unsigned long long x) {  
    x = (x >> 8) | (x << 8);  
    for (int i = 23; i >= 16; i--) {  
        x &= ~(1 << i);  
    }  
    return x;  
}
```

Задача 11: ** Пакеты с монетами (1037A)

Код:

```
#include <stdio.h>

int main() {
    long long n;
    scanf("%lld", &n);



    int min_packages = 0;

    while (n > 0) {
        n >>= 1;
        min_packages++;
    }

    printf("%d", min_packages);

    return 0;
}
```

Результат тестирующей программы:

Статус соревнования 									
№	Когда	Кто	Задача	Язык	Вердикт	Время	Память		
233423415	19.11.2023 16:21	n1kuso	1037A - Пакеты с монетами	GNU C11	Полное решение	15 мс	0 КБ		

Вывод: почтили навыки написания функций с побитовыми операциями.