

1. Дано целое число n от 1 до 10^9 . Найти произведение ненулевых цифр.

```
int digits(int n)
```

| | | | |
|-------|-------|-----|-----------|
| Ввод | 12045 | 999 | 100200300 |
| Вывод | 40 | 729 | 6 |

Пояснение: $\text{digits}(12045) = 5 * \text{digits}(1204)$.

2. Даны целые числа от -100 до 100, ввод заканчивается нулём. Найти сумму чисел.

```
int sumOfTail(void)
```

| | |
|-------|-----------------|
| Ввод | 1 2 3 4 -1 -2 0 |
| Вывод | 7 |

3. Даны целые числа от 1 до 100, ввод заканчивается нулём. Вывести числа в обратном порядке.

```
void readAndPrintTail(void)
```

| | |
|-------|-----------------|
| Ввод | 1 2 3 4 -1 -2 0 |
| Вывод | -2 -1 4 3 2 1 |

4. Даны целые числа n от 1 до 100 и k от 2 до 10. Вывести n в k -ичной системе счисления.

```
void printOneDigit(int n, int k)
```

| | | | |
|-------|------|------|---------|
| Ввод | 11 2 | 32 9 | 2016 10 |
| Вывод | 1011 | 35 | 2016 |

Пояснение: рекурсивная функция выводит одну цифру числа.

5. Даны целые числа от 1 до 10^9 , ввод заканчивается нулём. Найти наибольший общий делитель всех чисел.

```
int GCD(int a, int b); int multiGCD()
```

| | | | |
|-------|---------|-----------|-----|
| Ввод | 4 8 6 0 | 21 9 12 0 | 5 0 |
| Вывод | 2 | 3 | 5 |

Пояснение: рекурсивная функция `GCD` вычисляет наибольший общий делитель 2 целых чисел, а рекурсивная функция `multiGCD` считывает число и находит НОД между этим числом и всеми остальными, которые идут после.

6. Дано слово (длины не более 100), состоящее только из строчных латинских букв. Проверить, является ли это слово палиндромом. Выведите YES или NO соответственно.

```
int check(int n, char a[], int i)
```

| | | | |
|-------|-------|------|-----|
| Ввод | label | abba | gag |
| Вывод | NO | YES | YES |

Пояснение: `check(n, a, i)` возвращает 1, если подстрока `a[i..n-1-i]` является палиндромом, и 0 — иначе.

7. Даны целые числа a от 1 до 10^9 , n от 1 до 10^{18} и p от 2 до 10^9 . Вывести $a^n \bmod p$ с помощью бинарного возведения в степень

```
long long powerMod(long long a, long long n)
```

| | | | |
|-------|----------|-------|---------|
| Ввод | 11 2 100 | 5 4 9 | 2 64 10 |
| Вывод | 21 | 4 | 6 |

Пояснение: $a^n = b * b * a$ при нечетном n , и $a^n = b * b$ иначе, где $b = a^{n/2}$.

8. Даны последовательность символов, описывающих формулу вида:

формула := (формула знак формула) | цифра

цифра := 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

знак := + | - | *

Вычислить ее.

```
int check()
```

| | | | |
|-------|-----------|---------------|---|
| Ввод | (1+(2*3)) | ((1+2)*(1-3)) | 5 |
| Вывод | 6 | -6 | 5 |

Пояснение:

- 1) если считанный символ — цифра, то вернуть ее значение, иначе это — открывающаяся скобка;
- 2) вернуть результат вычислений формулы:
- 2а) получить значение из рекурсии;
- 2б) считать знак;
- 2в) получить значение из рекурсии;
- 2г) считать закрывающуюся скобку.

9. Даны n и k от 1 до 100. Распечатать всевозможные представления n в виде суммы k натуральных слагаемых.

```
void makesum(int sum, int number)
```

| | |
|-------|--|
| Ввод | 5 3 |
| Вывод | 1 1 3 1 2 2 1 3 1 2 1 2 2 2 1 3 1 1 |

10. Дано целое число n от 1 до 100 — размер массива. Далее n различных целых чисел — массив. Отсортировать массив сортировкой слиянием `mergesort`.

```
void sort(int n, int a[], int left, int right)
```

| | | | |
|-------|----------------|----------------|----------------|
| Ввод | 5 2 1 5 3 4 | 5 5 4 3 2 1 | 5 1 2 3 4 5 |
| Вывод | 1 2 3 4 5 | 1 2 3 4 5 | 1 2 3 4 5 |

Пояснение:

- 1) отсортируем подмассивы `a[left..middle]` и `a[middle+1..right]`;
- 2) объединим данные массивы в один отсортированный буфер:
- 2а) 2 указателя `l = left`; `r = middle`; двигаем вправо до тех пор, пока не закончится один из подмассивов них, при этом меньший из двух элементов будем переносить в буфер;
- 2б) копируем буфер в исходный массив.