Задача 1:

Направете функция, която изчислява корен трети, без да използвате math.h или друга външна библиотека.

Задача 2:

Да се реализира функцията:

void printValue(const void* valuePtr, uint8_t flag)

Нейната роля е да изведе в стандартния изход съответната стойност, сочена от указателя **valuePtr**. Типът на променливата да се определи от флагова променлива **flag** с предварително дефинирани константи - напр. **#define TINT 1** ....

Да се поддържа работата за следните типове: **int, char, double, float, uint8_t, uint16_t, uint32_t, uint64_t**.

Да се направи извикване на функцията в main() с подходящи примери.

Примерно извикване:	**Примерен изход:**
int num = 23; printValue(&num, TINT);	Value: 23
double num = 3.14; printValue(&num, TDOUBLE);	Value: 3.14
char symbol = ‘A’; printValue(&symbol, TCHAR);	Value: A

Задача 3:

Да се реализира функцията:

```
void filter_and_map(
     const int arr[],
     size_t n,
     int (*filter_f)(int),
     int (*map_f)(int),
     int dest[],
     size_t* dest_cnt
);
```

Функцията има следните параметри:

- **arr** - масив
- **n -** брой на елементите в **arr**
- **`int (*filter_f)(int)` –** указател към функция, която приема int и връща **Истина(1)** или **Лъжа(0)**
- **`int (*map_f)(int)` –** указател, който приема **int**, променя го и връща резултата
- **`int dest[]` –** масив, в който се запазва резултатът
- **`size_t* dest_cnt` –** указател, чрез който ще се запази броят на елементите добавени в **dest.**

Функцията преминава през всеки елемент на **arr** и определя дали даден елемент да бъде изхвърлен на базата на резултата от **filter_f.** Елементите, за които **filter_f** връща **Истина**, биват подадени на **map_f.** След това резултатът се запазва в масива **dest**.

Ако за определен елемент **filter_f** върне **Лъжа,** то той не се **map-ва** и не се запазва в **dest.**

Накрая - функцията запазва броят на елементите, които са запазени в **dest** на адреса, сочен от **dest_cnt.** Ако **dest** е **NULL**, то резултатите не се запазват.

Примерно извикване

- **`int isPositive(int a)`** - функция, която връща 1, ако **a** е положително и 0, иначе.
- **`int addOne(int a)`** - функция, която прибавя 1 към подадения аргумент и връща резултата.
- **`int arr[8] = {1, 2, 3, 4, -1, -2, 11, -100};`**
- **`int dest[10];`**
- **`size_t new_size;`**

```
filter_and_map(arr, 8, isPositive, addOne, dest, &new_size);
```

След горното извикване:

- `new_size = 5`
- `dest = {2, 3, 4, 5, 12};`

Задача 4:

Извиквайте функцията **`filter_and_map`** с подходящи аргументи, така че да получите следните функционалности:

- Да се изпечатат само четните числа от масив
- Да се намери квадратът само на простите числа от масив
- Да се намери броят на битовете, които са 1 само за положителните елементи на масив.