## Задача 1:

Създайте следната функция:

**int printf\_time(const char \****format***, ...);**

Функцията работи по същия начин като **printf** с единствената разлика, че преди да извика **printf**, за да запише подадените аргументи в стандартния изход, изпечатва текущото време в подходящ формат.

### Примерно извикване:

**printf\_time(“%d %d”, 10, 20);**

### Изход:

**[01.08.2023 – 19:19:23]**

## Задача 2:

Направете програма, в която потребителят въвежда от стандартния вход цяло число **n**. След това да се създаде масив от реални числа от **n** на брой елементи в динамичната памет. Елементите на масива да се инициализират с произволни стойности между 0,0 и 1.0. След това се въвежда ново число **m** и масивът се да се разшири с нови **m** елемента. Новите елементи да се инициализират с числа между 1.0 и 2.0. След това да се въведе цяло число **p**. Масивът да се разшири с нови **p** елемента. Новите елементи да се инициализират с числа между 2.0 и 3.0. Накрая да се изведе масива в стандартния изход. Да се освободи заетата памет.

Да се следи дали всяка функция успешно заделя памет.

<https://www.geeksforgeeks.org/dynamic-memory-allocation-in-c-using-malloc-calloc-free-and-realloc/>

## 

## Задача 3:

Направете имплементация на функцията от стандартната библиотека **printf:**

**int my\_printf(const char \*fmt, …);**

Функцията приема символен низ, определящ формата чрез **fmt.** Като минимум във **fmt** могат да се срещнат форматните спецификатори **%d, %f, %c.** Колкото форматни спецификатори са предадени,толкова аргумента от командия ред трябва да бъдат изпечатани от **my\_printf.** Във **fmt** могат да се срещнат и символи, които не са част от форматен спецификатор.

За печатане на символи на стандартния изход, използвайте **putc (или fptuc).**

Функцията връща броя на символите, които е изпечатала.

Направете примерни извиквания на **my\_printf**.