# MMS Star Camp 17

## КОНТРОЛНО №1

## Задача 1 - 10т.

Да се направи програма, която изчислява лицето на околната повърхнина и обемът на цилиндър по дадени радиус на основата **r** и височина **h**.

Да се създаде следната функция:

**int cylinder(double r, double h, double\* V, double\* S);**

При невалидни параметри (<= 0) функцията връща -1. При валидни параметри, функцията пресмята лицето и обема и ги запазва чрез указателите V и S.

Работата на програмата прочита по следния начин:

* От стандартния вход се прочитат радиус и височината на цилиндъра (на един ред, разделени с интервали).
* При некоректни данни се извежда съобщение за грешка на стандартния изход за грешки без да се прекратява програмата.
* При коректни данни се изпечатват лицето на околната повърхнина и обемът на цилиндъра в стандартния изход.

Тези стъпки се изпълняват докато не се въведат две нули от стандартния вход.

#### **Примерен вход и изход:**

**./cylinder**

**5 6**

**V = 471.24, S = 188.49**

**3 -2**

**Invalid data**

**3.2 1.6**

**V = 51.47, S = 32.16**

**0 0**

## Задача 2 - Побитови маски - 10т.

**2.1** **(3т.)** Създайте следната функция:

**unsigned onesCount(uint64\_t mask1, uint32\_t mask2, uint16\_t mask3, uint8\_t mask4);**

Функцията връща общия брой на битовете, които са ‘**1**’ във всички маски. Направете примерно извикване на функцията.

**2.2 (3т.)** Създайте следната функция:

**void flipOddBits(uint64\_t\* mask);**

Функцията инвертира (преобръща) всеки нечетен бит от маската **mask**. Броенето на битовете да се брои като най-младшия бит се приеме за номер 0. Направете примерно извикване на функцията.

**Напр.** 0b0000000000000000000000000000000000000000000000000000000000000000 -> 0b101010101010101010101010101010101010101010101010101010101010

**2.3** **(4т.)** Създайте следната функция:

**void mirrorBits(uint16\_t\* mask);**

Функцията обръща огледално битовете на маската **mask**. Направете примерно извикване на функцията.

***Напр****. 0b1111111100000000 -> 0b0000000011111111*

## Задача 3 - Работа с масиви - 10т.

Създайте следната функция:

**void subseq(int arr[], int n, int dest[], int \*new\_n);**

Функцията намира най-дългата последователна намаляваща подредица в масива **arr**.

* Намерената подредица се запазва в **dest.**
* Броят на елементите в **dest** се предава чрез указателя **new\_n.**

От стандартния вход се прочита **N** и след това **N** на брой цели числа. Използвайте **subseq**,за да намерите най-дългата последователна намаляваща подредица. Изпечатайте подредицата и нейния размер на стандартния изход.

#### **Примерен вход:**

**10**

**5 4 3 2 1 10 11 2 3 4**

#### **Примерен изход:**

**5**

**5 4 3 2 1**