

## Домашно 3 – леки задачи

**Задача 1.** Всяко цяло положително число, което няма други прости делители освен числата 2, 3 и 5 се нарича число на Хеминг. Напишете програма, която въвежда от клавиатурата естествено число  $n$  и извежда първите  $n$  числа на Хеминг.

| Вход | Изход           |
|------|-----------------|
| 3    | 1 2 3           |
| 8    | 1 2 3 4 5 6 8 9 |

**Задача 2.** От “*FMlvalut*” искат да раздадат награди на някои от техните потребители в знак на благодарност за оказаното доверие. Тъй като създателите са от ФМИ, то им се струва твърде тривиално да подаряват на принципа - 100-тен потребител, 1000-ден потребител и т.н. Затова те са решени да съберат малко повече статистика, чрез която да измислят математически модел за награждаването. За целта, ще трябва да напишете програма, която:

1. Прочита от стандартния вход цяло положително число  $n$  в интервала  $[1; 31]$ . (**Валидирайте** входа)
2. След това, последователно се прочитат  $n$  на брой идентификатори на карти - 9-цифрени числа (**ще бъдат валидни** идентификатори). Получаваме редицата  $id(0)$  - първо прочетено число,  $id(1)$  - второ и т.н до  $id(n-1)$ .
3. От програмата се очаква да отпечата на стандартния си изход следната статистика:
  - а) За  $id(0)$  винаги се извежда 1. След това за всеки следващ член  $id(j)$  извежда 1 ако той е  $\geq$  от последния известен член  $id(i)$  ( $0 \leq i < j$ ), такъв че за  $id(i)$  сме извели 1. За  $id(j)$  извеждаме 0 в противен случай.
  - б) Разглеждаме редицата от 0-ли и 1-ци получена от а), “медианна” 1-ца ще наричаме тази 1-ца, която е на средна позиция измежду всички единици (при нечетен брой) и **по-лявата** от двете 1-ци (при четен брой). Съответно тя се намира на позиция  $k$  в цялата поредица от 0-ли и 1-ци ( $k$  е в интервала  $[1; n]$ ). Програмата трябва да определи в кой от двата подинтервала определени от  $k$  -  $[1; k]$  или  $(k; n]$  има повече на брой 0-ли и колко точно са те. На стандартния изход изведете **Left side**: - ако повече 0-ли има в  $(1; k]$ , **Right side**: - ако повечето попадат в  $(k; 1]$  или **Both sides**: - ако 0-лите са по равен брой в двата подинтервала. След думите да се отпечата число - броят на съответните нули.

**Пример** (разглеждаме едноцифрени числа за опростяване):

| Вход:              | Изход:                         |
|--------------------|--------------------------------|
| 7<br>1 4 5 6 3 7 2 | 1 1 1 1 0 1 0<br>Right side: 2 |

**Обяснение за изхода:** 1, 4, 5, 6 са сортирани възходящо, затова на техните индекси се извеждат 1-ци. 3-ката не е част от наредбата - за нея се отпечата 0. 7-ката, пък участва в сортираната редица - изведено е 1, 2-ката не участва, на нейно място се извежда 0.

В редицата 1 1 1 1 0 1 0 - има 5 единици. “Медианната” единица е **удебелена**, от двете си страни тя има по 2 други единици. От лявата ѝ страна има само 1-ци, съответно броят на нулите е 0, от дясната страна имаме два броя 0-ли. Затова е отпечатано “Right side: 2”.

Съвети:

- **Няма** нужда да се използват масиви в задачата.
- Помислете колко да/не стойности можем да запишем в 1 байт? Ами в 4 байта?