

Задача 1. (1 т.) Да се напише програма, която въвежда кръстословица с размери 4x4 от стандартния вход и извежда на екрана колко думи се съдържат в нея.

За кръстословицата НЕ е позволено използването на масив. Всеки от символите в кръстословицата да се дефинира като отделна променлива. В кръстословицата могат да участват само малки латински букви и символът ‘-’.

Забележка: Да се реализира валидация на входа.

Дума в кръстословицата представлява последователност от 3 или повече последователни латински букви, които могат да се прочетат по хоризонтала или вертикална. Думите в кръстословицата са разделени от празни квадратчета, отбелязани със символа ‘-’.

Вход:

| | | | |
|---|---|---|---|
| - | d | o | g |
| r | e | a | l |
| i | e | - | o |
| o | r | - | w |

↓ ↓ ↓ ↓
1 1 0 1

Изход: 5

→ 1
→ 1
→ 0
→ 0

Задача 2. (1.5 т.) Да се дефинира масив от цели числа с максимален капацитет 1024.

Две числа x_i и x_j образуват **дистрибутирана двойка**, ако $i < j$ и $x_j - x_i = j - i$ (разликата между стойностите съвпада с разликата между техните **индекси**). Да се напише програма, която извежда **индексите** на всички елементи, които:

1. участват в поне една дистрибутирана двойка, и
2. се намират между други два индекса на елементи, които също участват в дистрибутирани двойки.

Ако няма нито един такъв индекс, да се изведе “**none**”.

Забележка: Да се реализира валидация на входа.

| Вход | Изход | Обяснение |
|---|--------------|---|
| 5 3 <u>5</u> 9 7 <u>11</u> | 2 3 | (5,7) и (9, 11) са дистрибутирани двойки. |
| 7 <u>10</u> 100 12 7 9 <u>20</u> 21 | 2 5 | (10,12) и (20, 21) са дистрибутирани двойки |
| 4 1 100 50 4 | none | |

Задача 3. (1.5 т.) Вариация на дадено естествено число x се нарича това число, получено след като в двоичното му представяне при срещане на еднакви съседни битове всички освен първия се заменят с тяхното отрицание. Да се напише програма, която намира вариацията на дадено число.

*Забележка: Всички преобразования да се правят **намясто** (*in-place*). Използването на побитови операции не е задължително. Не е позволено използването на масиви.*

Пример:

Вход: 36

Изход: 45

Обяснение: $36_{(10)} \rightarrow 100100_{(2)} \rightarrow 101101_{(2)} \rightarrow 45_{(10)}$

Вход: 57

Изход: 35

Обяснение: $57_{(10)} \rightarrow 111001_{(2)} \rightarrow 100011_{(2)} \rightarrow 35_{(10)}$

Предложение за задача:

Аня и Ванката искат да засадят **M** дървета покрай една улица. На тази улица са предвидени **N** ($1 \leq N \leq 1024$) възможни места за засаждане. Те искат да засадят дърветата така че минималното разстояние между всеки две дървета да е максимално голямо. Помогнете на Аня и Ванката като напишете програма, която смята кое е това разстояние.

Пример:

Вход:

$N = 5; M = 2$

Разстояния от началото на улицата до всяко от местата: 5, 8, 12, 32, 1.

Изход: 31

Вход:

$N = 7; M = 3$

Разстояния от началото на улицата до всяко от местата: 1, 15, 35, 10, 59, 60, 28.

Изход: 34