

## СУ „Св. Климент Охридски“, ФМИ

СПЕЦИАЛНОСТ „СОФТУЕРНО ИНЖЕНЕРСТВО“

## Увод в програмирането, 2023-2024 г.

## Задачи за домашно № 1

## Задача 1:

Да се напише програма, която получава от стандартния вход цели положителни числа  $n$  и  $k$ . Програмата да извежда на стандартния изход броя двойки цели числа  $(x, y)$ , такива че  $1 \leq x \leq y \leq n$ , които удовлетворяват равенството:  $k \cdot d(x) \cdot d(y) = x \cdot y$ , където  $d(x)$  е броят делители на  $x$  и  $d(y)$  – броят делители на  $y$ .

*Пояснение: Търси се броят на двойките числа, за които произведението на броя на делителите им е точно  $k$  пъти по-малко от произведението на самите числа.*

## Ограничения:

$$1 \leq x \leq y \leq n$$

## Пример:

Вход	Изход	Обяснение
8 3	2	Двете двойки, които удовлетворяват равенството са (3, 8) и (6, 8). (3, 8): Числото 3 има два делителя – 1 и 3, и следователно $d(3)=2$ . Числото 8 има четири делителя – 1, 2, 4 и 8, и $d(8)=4$ . Заместваме в $k \cdot d(x) \cdot d(y) = x \cdot y$ . $\leftrightarrow$ $3 \cdot 2 \cdot 4 = 3 \cdot 8$ (6, 8): Числото 6 има четири делителя – 1, 2, 3 и 6, и следователно $d(6)=4$ . Числото 8 има четири делителя – 1, 2, 4 и 8, и $d(8)=4$ . Заместваме в $k \cdot d(x) \cdot d(y) = x \cdot y$ $\leftrightarrow$ $3 \cdot 4 \cdot 4 = 6 \cdot 8$
25 9	6	Двойките числа са (9, 9), (9, 18), (9, 24), (18, 18), (18, 24) и (24, 24)

**Задача 2:**

В държавата Упландия живеят хора с всевъзможен цвят на косата. Веднъж боядисали косата си в даден цвят, те никога не го променят. Друг интересен факт е, че всеки жител знае точния брой на хората, които имат същия цвят коса като неговия. Те обаче имат проблем - не знаят колко души живеят в държавата.

Вашата задача е да им помогнете като определите колко най-малко биха били хората, живеещи там. Разполагате със следната информация:

Когато срещнете гражданин на Упландия, можете да го попитате колко още човека имат същия цвят коса. Гарантирано е, че той ще каже истината.

Да се напише програма, която приема от стандартния вход брой запитвания  $n$ , след което  $n$  на брой отговори от граждани на Упландия. Програмата да извежда на стандартния изход минималния брой на гражданите на Упландия.

**Ограничения:**

$$1 \leq n \leq 1000$$

$$0 \leq \text{arr}[i] \leq 1000$$

**Пример:**

Вход	Изход	Обяснение
3 1 1 2	5	Срещнали сте трима граждани. Първият и вторият са казали, че има още 1 с коса като тяхната. Това означава, че е възможно да са с еднакъв цвят на косата (например <b>синя</b> ) Третият казва, че има още двама като него. Той няма как да е със синя коса, защото тогава първите двама са излъгали. Нека третият е с <b>червена</b> коса. Тогава има още двама, които са с червена коса, но не сте ги срещнали. Така най-малкият брой на жителите е 5: 3, които сте попитали + 2, които не сте срещнали.
3 10 10 10	11	
4 2 2 2 3	7	
4 1 1 1 1	4	Срещнали сте четирима граждани. Всеки от тях твърди, че има още един гражданин със същия цвят на косата като текущо попитания. Минималният брой граждани ще се постигне, ако двама от тях са с цвят на косата <b>A</b> , а другите двама са с цвят на косата <b>B</b>
1	1	

0		
---	--	--

**Задача 3:**

Имате последователност от  $n$  разноцветни топчета, подредени в редица. Можете да изстрелвате топчета с определен цвят към редицата от топчета. Изстреляното топче попада точно след топчето, към чиято позиция е било изстреляно. Новото топче измества всички топчета на позиции по-големи от него с една позиция в дясно. Ако нововъведеното топче е част от последователност от топчета с еднакъв цвят с дължина поне 3, цялата последователност се изтрива, след което дясната част се долепя до лявата. Ако и от двете страни на мястото на слепяне, седят топчета от един и същ цвят и са част от редица отново с дължина поне 3, процедурата се повтаря.

На първия ред се въвежда  $n$  - броя на топчетата. На следващите  $n$  реда се въвеждат цветовете на топчета, които са символи. След това се въвежда  $s$  – брой изстрели. На следващите  $s$  реда се въвеждат:  $i$  - индекс на топчето, към което стреляте, и  $c$  - цвят на изстреляното топче.

Програмата да изведе на стандартния изход състоянието на редицата след всичките стрелания по редицата. Ако тя е празна, да се изведе -1.

При невалиден индекс, изстреляното топче изчезва.

**Ограничения:**

$n \in (0, 256)$

$s \in [0, 127]$

$c \in [a-z]$

**Примери:**

Вход	Изход	Обяснение
15 b b a a a b b b b c c c a a b 4 10 c 6 b 1 a 3 b	-1	След първи изстрел: b b a a a b b b b (c c c) a a b – редицата от c-та се унищожава След втория: b b a a a (b b b b b) a a b – редицата от b-та се унищожава След слепянето: b b (a a a a a) b – редицата от a-та се унищожава След второто слепяне: (b b b) – редицата от b-та се унищожава и не остава нищо На следващите две изстрелвания, не се случва нищо понеже редицата вече е празна и индексите 1 и 3 са невалидни

10 a a a b b a a a c c 5 1 d 10 e 10 f 4 b 14 p	a a d c c f e	След първия изстрел: a a d a b b a a a c c След втория: a a d a b b a a a c c e След третия: a a d a b b a a a c c f e След четвъртия: a a d a (b b b) a a a c c f e – редицата от b-та се унищожава След слепянето: a a d (a a a a) c c f e -редицата от a-та се унищожава След слепянето: a a d c c f e След петия: Остава същата понеже индексът е невалиден
8 a a a c c b b b 3 3 c 0 a 0 b	-1	След първия изстрел: a a a (c c c) b b b => a a a b b b След втория изстрел: (a a a) b b b => b b b След третия изстрел: (b b b b) => []
11 c s a a b b b a a c c 3 3 b 4 f 6 j	-1	След първия изстрел: c s a a (b b b b) a a c c => c c (a a   a a) c c => (c c   c c) => [] където   е мястото на слепяне  Индексите на останалите изстрели са невалидни.

**Задача 4:**

Да се напише програма, която разделя интервал между две дати на по-малки интервали. Програмата приема като вход началото и края на интервала в реда година, месец, ден (като допустимите дати са между 1 януари 1600 г. и 31 декември 2100 г. Началната дата трябва да е преди крайната, както и число  $n \in [1, 365]$  – максималната допустима дължина на интервал, измерена в дни.

Първият интервал в резултата трябва да започва от началната дата, а последният – да завършва с крайната. Всеки интервал трябва да съдържа точно  $n$  дни, с изключение на последния, който може да е по-кратък. Всеки интервал трябва да започва с деня, който идва след последния ден от предходния интервал.

Използваме опростен модел на датите, който има следните особености:

- През 1916 г. България преминава от Юлианския календар (“стар стил”) към Григорианския календар (“нов стил”), при което се “изпускат” 13 дни. Денят след 31 март 1916 г. е 14 април 1916 г. (датите 1 до 13 април 1916 г. не съществуват)
- В Юлианския календар (дати до 31 март 1916 г.), високосна е всяка година, която се дели на 4.
- В Григорианския календар (дати от 14 април 1916 г. нататък), високосна е всяка година, която се дели на 400, както и тези години, които се делят на 4 и не се делят на 100.

**Ограничения:**

Упоменати в условието.

**Примери:**

Вход	Изход
2023 10 1 2023 10 18 5	2023-10-1 - 2023-10-5 2023-10-6 - 2023-10-10 2023-10-11 - 2023-10-15 2023-10-16 - 2023-10-18
2023 2 23 2023 3 4 5	2023-2-23 - 2023-2-27 2023-2-28 - 2023-3-4

2020 2 23 2020 3 4 5	2020-2-23 - 2020-2-27 2020-2-28 - 2020-3-3 2020-3-4 - 2020-3-4
1916 3 30 1916 4 15 5	1916-3-30 - 1916-4-15
1916 3 30 1916 4 15 1	1916-3-30 - 1916-3-30 1916-3-31 - 1916-3-31 1916-4-14 - 1916-4-14 1916-4-15 - 1916-4-15
1900 2 27 1900 3 4 4	1900-2-27 - 2023-3- <b>1</b> 1900-3- <b>2</b> - 1900-3-4
2100 2 27 2100 3 4 4	2100-2-27 - 2100-3-2 2100-3-3 - 2100-3-4

## Допълнителни инструкции за решаване и предаване на домашното:

1. Всички коректно решени задачи носят по 2,5 точки.
2. Обърнете внимание, че в условието на всички задачи е посочено множеството от допустимите стойности на входните данни. Ако входът който се подаде е извън това множество, програмата трябва да изведе в конзолата следното съобщение: **"Incorrect input"** и да завърши своето изпълнение.
3. Опитайте се да напишете максимално ефективен код, както по отношение на брой редове, така и по отношение на време за изпълнение. Помислете къде може да се намали броят на повторенията на циклите или да се намали броят на променливите, които използвате за решаване на задачата.
4. Всички задачи ще бъдат проверени автоматично за преписване. Файловете с голямо съвпадение ще бъдат проверени ръчно от преподавателите и при установено плагиатство ще бъдат анулирани.
5. **Всички решения на задачите от домашното трябва да бъдат предадени преди сряда, 6 декември, 23:59 ч. Линкът за предаване на задачите ще бъде наличен няколко дни преди крайния срок.**
6. За решаване на задачите **не се допуска използване на STL функции.**
7. **Предадените от вас решения трябва да са написани на езика C/C++ и да могат да се компилират успешно на GCC.**
8. **Всяка задача от домашното трябва да бъде решена в точно един, отделен файл.** Името на файла трябва да бъде в следния формат:

fnXXXXX\_d1\_N.cpp, където:

- XXXXX е вашият факултетен номер
  - N е номерът на задачата.
9. Предаването на домашното се извършва в Github, според линка за предаване посочен в Moodle, като задачите се commit-ват в главната директория на repository-то. В Actions може да видите дали предадени задачи се компилират и дали примерните входове от условието се изпълняват успешно. След края на срока за предаване задачите ще бъдат тествани допълнително и с други тестове.
  10. Файловете с решенията, които предавате трябва да са оформени съгласно добрите практики за оформяне на кода, за които се говори по време на лекции и упражнения. Ще се отнемат точки за неинформативни имена на променливи, неизползване на подходящи константи и т.н.
  11. Файловете с решенията може да съдържат само стандартните ASCII символи с кодове от 0-127 (не се разрешава използване на кирилица, например в стринговете или коментарите!).
  12. Всички предадени програми трябва следят за некоректно въведени входни данни от потребителя, в зависимост от условието на задачата.
  13. В началото на всеки един от файловете, които предавате, трябва има коментарен блок, който носи информация за съдържанието на файла. Този коментарен блок



трябва да изглежда точно така, както е показано по-долу, като в него попълните информация за Вас. За улеснение, просто копирайте дадения по-долу блок и попълнете в него необходимите данни, вместо текста, маркиран с ъглови скоби. Обърнете внимание, че на първия ред след наклонената черта има две звезди и че във файловете не може да се съдържат символи на кирилица.

```
/**
 *
 * Solution to homework assignment 1
 * Introduction to programming course
 * Faculty of Mathematics and Informatics of Sofia University
 * Winter semester 2023/2024
 *
 * @author <вашето име>
 * @idnumber <вашият факултетен номер>
 * @task <номер на задача>
 */
```

Например един попълнен блок за студент с име Хикс Хиксов, ф.н. 3MI1234567, който предава задача 2, трябва да изглежда така:

```
/**
 *
 * Solution to homework assignment 1
 * Introduction to programming course
 * Faculty of Mathematics and Informatics of Sofia University
 * Winter semester 2023/2024
 *
 * @author Hiks Hiksov
 * @idnumber 3MI1234567
 * @task 2
 */
```

14. Предадени домашни, които не отговарят на условията от точки 5-12 от инструкциите описани по-горе ще бъдат оценени с 0 точки.