

СУ „Св. Климент Охридски“, ФМИ

СПЕЦИАЛНОСТ „СОФТУЕРНО ИНЖЕНЕРСТВО“

Увод в програмирането, 2020-2021 г.

Задачи за домашно № 4

1. Напишете функция $F()$ ¹ с параметри масив от естествени числа, дължина на масива M и естествено число N . Функцията да връща като резултат булева стойност дали масивът (редицата от числа образувана от елементите на масива) може да се превърне в нарастващ със смяната на най-много N на брой числа в него с кои да е други естествени числа (без да се извършват размествания в масива). Напишете функция `main()`, в която потребителят последователно въвежда N , M и редица от M естествени числа и извежда резултатът от изпълнението на $F()$ върху така въведените параметри.

Примери:

Вход	Изход
2 6 8 3 9 2 12 15	1
2 5 10 3 2 7 6	0
2 5 1 2 2 19 14	1

2. Разполагате с шахматна дъска с размери $N \times M$. Шахматната фигура кон започва от движението си от координати (p, q) , $p < N$, $q < M$. Напишете функция $F()$ ¹, която при дадени координати $p1$ и $q1$, както и число k , връща като резултат булева стойност дали конят може да се намира на позиция $(p1, q1)$ след не повече от k на брой ходове. Напишете функция `main()`, според указанията по-долу, с която да демонстрирате работата на $F()$.

¹ Сами определете подходящо име на функцията.

Приемаме стандартните правила на играта Шах, при които конят за един ход се движи във формата на буквата "Г". Например от стартова позиция с координати (5,5), той може да се придвижи на някоя от следните позиции: (3,4), (3,6), (4,7), (6,7), (7,6), (7,4), (6,3), (4,3), но само при условие, че всеки от валидните координати е в рамките на шахматната дъска. Задачата да се реши с рекурсивен подход.

Формат на входа: От първия ред на стандартния вход се прочитат числата N и M - броя редове и колони на шахматната дъска. От втория ред на стандартния вход се прочитат числата p и q. От третият ред на стандартния вход се прочитат числата p1 и q1. От четвъртия ред на стандартния изход се прочита число k.

Ограничения: $3 \leq N, M, p, q, p1, q1 \leq 200$, $k \leq 30$.

Формат на изхода: На единствения ред на стандартния изход изведете "true", ако е възможно след не повече от k на брой хода да бъде достигната позиция (p1, q1) или "false" в противен случай.

Примери:

Вход	Изход	Пояснение
6 8 1 2 4 5 2	True	<div> <p>На изображението е показана дъска с размери 6x8. Стартовата позиция (1, 2) е маркирана с червено. С розово са отбелязани клетките, които кон може да достигне след 1 ход. С лилаво - клетките, които кон може да достигне след 2 хода.</p> <p>Позиция (4, 5) е оцветена в лилаво, което означава, че тя може да бъде достигната за 2 хода (и по-общо, за не повече от 2 хода). Позиция (3, 0) е оцветена в бяло - не може да бъде достигната.</p> </div>
6 8 1 2 3 0 2	False	

3. Да се напише функция $F()$, която да пресмята аритметична операция на две обикновени дроби. За целта функцията да приема като параметри 4 цели числа, които представляват съответно числителя и знаменателя на първата дроб и числителя и знаменателя на втората дроб и символ, който представлява аритметична операция – $\{+, -, *, /\}$. Функцията да връща като резултат масив от две цели числа – на първо място да е числителят на резултата от операцията с дробите, а на второ място – знаменателя.

Напишете функция $\text{main}()$, която да чете от клавиатурата:

- Две цели числа в интервала $[-10^8, 10^8]$ - числителя и знаменателя на първата дроб
- Аритметична операция- символ $\{+, -, *, /\}$
- Две цели числа в интервала $[-10^8, 10^8]$ - числителя и знаменателя на втората дроб

След което да извежда като резултат в конзолата две числа - числителя и знаменателя на опростената дроб, която се получава в резултат на изчислението.

Примери:

Вход	Изход
1 2 + 1 3	5 6
3 9 * 2 4	1 6

Пояснения:

1. Задачи 1 и 3 носят по 3 точки, а задача 2 - 4 точки.
2. Всички задачи ще бъдат проверени автоматично за преписване. Файловете с голямо съвпадение ще бъдат проверени ръчно от лектора и при установено плагиатство ще бъдат анулирани.
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. **За решаване на задачи 1 и 3 не се допуска използването на `vector` и/или `string`.**
5. Предадените от вас решения трябва да могат да се компилират успешно на Visual C++ или GCC.
6. Всяка задача от домашното трябва да бъде решена в точно един, отделен файл. Името на файла трябва да бъде в следния формат:

`fnXXXXX_d4_N_CC.cpp`, където:

- XXXXX е вашият факултетен номер
- N е номерът на задачата
- CC указва кой компилатор сте използвали. Стойността му може да бъде "gcc" за GCC или "vc" за Visual C++.

7. Архивирайте всички файлове, които предавате в един архивен файл, компресиран в стандартен zip формат, със следното име:

`UP_20-21_fnXXXXX_d4.zip`, където XXXXX е вашият факултетен номер

8. 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.
9. Всички предадени програми трябва следят за некоректно въведени входни данни от потребителя, в зависимост от условието на задачата.
10. Файловете с решенията може да съдържат само стандартните символи с кодове от 0-127 (не се разрешава използване на кирилица, например в стринговете или коментарите!).
11. Първото нещо във всеки от файловете, които предавате, трябва да бъде коментарен блок, който носи информация за съдържанието на файла. Този коментар 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 4
 * Introduction to programming course
 * Faculty of Mathematics and Informatics of Sofia University
 * Winter semester 2020/2021
 *
 * @author <вашето име>
 * @idnumber <вашият факултетен номер>
 * @task <номер на задача>
 * @compiler <използван компилатор - GCC или VC>
 *
 */

```

Например един попълнен блок за студент с име Иван Иванов, ф.н. 12345, който предава задача 2, компилирана с GCC, трябва да изглежда така:

```

/**
 *
 * Solution to homework assignment 4
 * Introduction to programming course
 * Faculty of Mathematics and Informatics of Sofia University
 * Winter semester 2020/2021
 *
 * @author Ivan Ivanov
 * @idnumber 12345
 * @task 2
 * @compiler GCC
 *
 */

```

12. Предадени домашни, които не отговарят на условията от точки 4-10 ще бъдат оценени с 0 точки.