

**ТРЕТО КОНТРОЛНО СЪСТЕЗАНИЕ НА РАЗШИРЕНИЯ
НАЦИОНАЛЕН ОТБОР
6 юни 2021 г.
Група С, 7 – 8 клас**

Задача СК9. РОБОКАРИ

Два робокара са изгубени в склад. Робокарите са номерирани с 1 и 2. Складът представлява мрежа от квадратни полета, подредени в R реда и C колони. Всяко поле е или блокирано, или достъпно за всеки един от робокарите. Робокарите се управляват дистанционно, като получават кратки команди.

Всяка команда се състои от две части: Номер на робокар, към който е адресирана командата (може да има стойност 1 или 2), следван от символ 'U', 'D', 'L' или 'R', указващ посоката (нагоре, надолу, наляво, надясно), в която трябва да се премести робокарът при изпълнение на командата. Двете части на командата са разделени с един интервал.

Ако полето на местоназначението е блокирано, заето от друг робокар или е извън габаритите на склада, нищо не се променя и робокарът остава там, където е бил преди получаване на командата.

Ако полето е свободно, при изпълнение на командата, робокарът се премества в него. Робокарите не могат да рапортуват своето точно местоположение, но, след изпълнение на всяка команда, рапортуват Манхатъновото разстояние, на което се намират един от друг. Ако робокарите се намират в полета с координати (r1, c1) и (r2, c2), тогава Манхатъновото разстояние между тях е $|r1 - r2| + |c1 - c2|$.

След изпълнението на всяка команда, неуспешна или успешна, единствената информация, която получаваме е текущото Манхатъново разстояние между двата робокара. Първоначално двата робокара са разположени в склада на различни места.

Задача

Напишете програма **robo**, съдържаща функцията **robo**, която да издаде поредица от команди, необходими за преместване на двата робокара на указани паркоместа в склада. Тази програма ще се компилира заедно с програмата на журито. Командите се задават чрез извикване на функцията **output**.

Детайли по реализацията

Функцията **robo**, която трябва да реализирате, има следния формат:

```
void robo (int R, int C, char store[][201], int dist);
```

Тя се вика веднъж в началото от програмата на журито с четири параметъра: броя редове, броя колони, матрица с описанието на склада и началното Манхатъново разстояние между двата робокара. Матрицата представлява R низа с по C символа, които описват склада. Възможни са само следните символи:

- '.' - представлява свободно поле;
- '#' - представлява блокирано поле;
- 'R' - местоположение на паркоместа.

Гарантирано е, че ще има точно два знака 'R'. За предаване на команди за движението на робокарите, трябва да викате функцията **output**, която има следния формат:

```
int output (int robot, char direction);
```

Първият параметър е номер на робокар, за който се отнася командата (едно или две), а втория параметър е символ, указващ посоката на преместване (един от символите 'U', 'D', 'L' и 'R'). Функцията връща Манхатъновото разстояние между двата робокара след изпълнението на командата.

Гарантирано е, че всички свободни полета в склада са свързани, както и че има поне едно поле, което не е блокирано и не е заето от робокар.

Вие трябва да предадете към системата файл **robo.cpp**, който съдържа функцията

ТРЕТО КОНТРОЛНО СЪСТЕЗАНИЕ НА РАЗШИРЕНИЯ НАЦИОНАЛЕН ОТБОР

6 юни 2021 г.

Група С, 7 – 8 клас

robo. Той може да съдържа и друг код, и функции, необходими за работата Ви, но не трябва да съдържа главната функция *main*. Също така, не трябва да четете от стандартния вход или да пишете на стандартния изход. Програмата на журито ще преустанови изпълнението и ще го отчете като успешно, когато Вашата функция изпълни успешно мисията. Изпълнението е неуспешно, когато приключи поради изчерпване на лимита от време, невалидна команда или ако Вашата функция приключи без да е изпълнила успешно мисията.

В началото си Вашият файл трябва да съдържа: `#include "robo.h"`.

Ограничения

$2 \leq R, C \leq 200$

В 40% от тестовете: $R, C \leq 25$

Примерна комуникация с програмата на журито

Извикване на функция	Обяснение
<i>robo</i> (4, 5, преобразуван склад, 6)	В началото се вика Вашата функция от програмата на журито, като схемата на <i>склада</i> е показана по-долу. На Вашата програма се подава <i>преобразуван склад</i> , където позициите на робокарите са означени като празни – '.'. Началното Манхатъново разстояние между робокарите е $ 1-4 + 4-1 =6$.
<i>output</i> (1, L)	Вашата функция задава команда за движение на робот 1. Тя го премества наляво, с което той заема едното паркомясто.
<i>output</i> (2, R) <i>output</i> (2, R) <i>output</i> (2, R) <i>output</i> (2, R)	С тази поредица от команди, робокар 2 се премества 4 позиции надясно и така той заема последното паркомясто. След изпълнението на последната команда, програмата на журито прекратява изпълнение и завършва с успех.

Реален склад:

```
4 5
##P1.
.##..
.....
2...P
```

Преобразуван склад:

```
##P..
.##..
.....
....P
```

Локално тестване

За локално тестване е предоставен файла **Lgrader.cpp**. Сложете го в същата папка, в която е Вашият файл **robo.cpp**, и компилирайте *само* **Lgrader.cpp**. Така ще получите програма, с която ще проверите верността на функцията Ви. Програмата ще изисква от стандартния вход следната последователност от данни:

- Първият ред на въвеждане трябва да съдържа две цели числа R и C.
- Следващите R редове трябва да съдържат по C знака всеки. Символите могат да бъдат само '.', '#', 'P', '1' и '2'. Трябва да има точно два знака 'P'. Символите '1' и '2' означават началните места на робокарите. Трябва да има точно по един от тях във входа. Разбира се, символите '1' и '2' ще бъдат преобразувани в '.', когато описанието на склада се предава като параметър към Вашето решение. (на вход на локалния грейдър трябва да въведете *реалния склад* от примера по-горе, а не *преобразувания склад*)

Допълнително на първия ред на локалния грейдър има флаг за работата му – макроса с име DETAILED. По подразбиране (стойност true) отпечатва текущото състояние след всяко подаване на команда. Ако се смени на стойност false, тогава има само съобщение, след приключване на програмата, за това дали е изпълнена успешно мисията.