

Контролна работа № 1 по Функционално програмиране
КН, II курс, I поток, 09.11.2024 г.

Решенията трябва да са готови за автоматично тестване. Важно е програмният код да бъде добре форматиран. Предайте решенията на задачите в **един** файл с име **k1_<FN>.rkt**, където **<FN>** е Вашият факултетен номер.

Решения предадени като архив **няма** да се оценяват.

Приятна работа и успех!

Нека е дадена картата на пътищата в игрално поле, представено като правоъгълна таблица от клетки. Всяка от клетките може да е свързана с някои от нейните четири съседни клетки (горна, долна, лява и дясна клетка). Картата се състои от низ с по един символ за всяка клетка, като:

- низът "|" означава, че клетката е свързана с горната и долната клетка;
- низът "-" означава, че клетката е свързана с лявата и дясната клетка;
- низът "└" означава, че клетката е свързана с горната и дясната клетка;
- низът "┐" означава, че клетката е свързана с горната и лявата клетка;
- низът "└" означава, че клетката е свързана с лявата и долната клетка;
- низът "┐" означава, че клетката е свързана с дясната и долната клетка;
- низът "." означава, че от тази клетка не може да се продължи към друга;
- низът "S" означава началната позиция на играч, като тази клетка е свързана със всичките си съседни клетки.

Всяка карта се състои от много пътища, които не образуват цикъл.

Картата има представянето на матрица от низове, т.е. списък от списъци от низове.

Задача 1. (5 точки) Да се дефинира процедура (**find-start player-map**), която приема карта и връща като точкова двойка координатите на началната позиция на играча (клетката в картата с низ "S"). Индексирането е от 0 и започва от горния ляв ъгъл на картата. Първата стойност в двойката е индекс на ред, а втората - индекс на колона.

Примери:

```
(find-start '(("." "." ".")
              ("." "." ".")
              ("." "S" ".")
              ("." "." ".")))
→ '(2 . 1)
```

```
(find-start '("." "." "F" "J")
            ("S" "-" "|" ".")
            ("." "." "J" ".") )
→ '(1 . 0)
```

Задача 2. (10 точки) Да се дефинира процедура (**get-next-steps player-map current-position**), която приема карта и точкова двойка с координати на текуща позиция на играч и връща списък от точкови двойки, които съответстват на координатите на следващите възможни позиции, на които играчът може да се премести.

Примери:

```
(get-next-steps '("." "." ".")
                ("." "." ".")
                ("." "S" ".")
                ("." "." ".") )
→ '(2 . 1))
→ '((1 . 1) (3 . 1) (2 . 2) (2 . 0))
```

```
(get-next-steps '("." "." "F" "J")
                ("S" "-" "|" ".")
                ("." "." "J" ".") )
→ '(1 . 2))
→ '((0 . 2) (2 . 2))
```

```
(get-next-steps '("." "." "F" "J")
                ("S" "-" "|" ".")
                ("." "." "J" ".") )
→ '(0 . 3))
→ '((0 . 2))
```

Задача 3. (15 точки) Да се дефинира процедура (**num-steps-farthest player-map**), която приема карта и намира броя на стъпките в най-дългия път, който започва от стартовата позиция "S". Гарантирано е, че в подадената карта няма да има пътища, които да образуват цикъл.

Примери:

```
(num-steps-farthest '("." ".")
                    ("S" ".")
                    ("." ".") )
→ 1
```

```
(num-steps-farthest '("." "." ".")
                    ("S" "-" "|" ".")
                    ("." "." ".") )
```

→ 3

```
(num-steps-farthest ' (("." "." "F" "J")
                      ("S" "-" "|" ".")
                      ( "." " ." "J" " .") ) )
```

→ 4

```
(num-steps-farthest ' (("." "." " ." ".")
                      ( "." " ." "F" "J")
                      ("S" "-" "|" ".")
                      ( "." " ." "J" " .") ) )
```

→ 5

```
(num-steps-farthest ' (("." "." " ." "|")
                      ( "." " ." "F" "J")
                      ("S" "-" "|" ".")
                      ( "." " ." "J" " .") ) )
```

→ 5

```
(num-steps-farthest ' ( ("F" "-" "7" " ." " ." "|")
                      ("|" " ." "|" " ." "F" "J")
                      ("|" " ." "S" "-" "|" " .")
                      ("|" " ." " ." " ." "|" " .")
                      ("L" "-" "-" " ." "J" " .") ) )
```

→ 11