

CS Games 2017



GSOFT - Team Software Engineering

Participants	3
Workstations	3
Value	6%
Duration	3 hours

At GSOFTE, we developed a love for coffee. Lately, we thought we could give Sébastien (Our Barista) an helping hand with our coffee orders. So we equipped the espresso machines with an Arduino and we hacked a Roomba to automate the distribution of coffee in the map.

Your objective is to create a program that takes as input a map and a list of orders, and outputs a series of commands that will allow the Roomba and the espresso machines to prepare coffees and deliver them to the right person.

The program must meet the following input specifications:

```
<your_program> <map> <missions>
```

You can use the programming language of your choice to accomplish this task. You will need to provide all necessary documentation to compile and run your program.

Input

Map

The provided card is in an ASCII format:

```
##### Legend
#1      #      #      A: Audren
#  #####      #      B: Bastien
#  ###B      #      C: Carl
# @ ###      #      ...
#          #      Z: Zoey
#####2      #
#A          #      1...9: Espresso machines
#####      @: Roomba
          #: Wall
```

- The letters A to Z correspond to the persons in the map, the numbers correspond to each espresso machine available in the map.
- The starting position of the Roomba is represented by the @.
- It is possible to move only in empty spaces.

Orders

Coffee orders can be done 2 ways, either by passing a detailed description of the order in JSON or with the name of a coffee that is available in the menu of the coffee GSOFTE.

An order is in the following format:

```
<person> <coffee>
```

The `coffee` attribute is a Base64-encoded UTF8 string that contains either a JSON object for a customized coffee or the name of the coffee to be ordered from the GSOFTE menu.

A custom order contains the following JSON structure:

```
{
  "size": <"solo", "doppio", "triplo">,
  "length": <"ristretto", "normale", "lungo">,
  "inverted": <true, false>,
  "ingredients":
    "milk": <0-500>,
    "foam": <0-500>,
    "hot_water": <0-500>
}
```

- The `size` attribute corresponds to whether it is a single, double or triple espresso.
- The `length` attribute corresponds to whether it is a short, normal or elongated espresso.
- The `inverted` attribute is `true` when the ingredients are to be put in the cup before the coffee.
- The ingredients are `milk`, `foam` and `hot_water`.
- The ingredients are in milliliters (ml).
- The ingredients are always in steps of 25.
- The menu of the GSOFTE coffee is available in the file `menu.json`.

Roomba

The Roomba understands the following commands:

Command	Meaning	Description
F	Forward	Move the Roomba forward by a square according to the orientation of the Roomba
R	Right	Rotate the Roomba 90 'clockwise
L	Left	Rotate the Roomba 90 'counterclockwise
T	Take the cup	Take the Cup from the adjacent espresso machine
G	Give	Gives the right coffee to the adjacent person
	"Wait"	No command

F.A.Q. on the Roomba

- The Roomba can only carry 2 coffees at a time.
- The initial orientation of the Roomba is facing north.
- The Roomba is considered adjacent to an espresso machine or a person when it is North, East, South or West of that object's square.
- The Roomba can interact with an espresso machine or a person regardless of its orientation.
- A cup transfer from the espresso machine to the Roomba must be synchronized. The Roomba must send the "T" command at the same time as the espresso machine sends the "G"

Espresso machine

Each espresso machine understands the following commands:

Command	Meaning	Description
S	"Insert a Small Cup"	Place an empty 50ml cup in the espresso machine
M	"Insert a Medium Cup"	Place an empty cup of 250ml in the espresso machine
L	"Insert a Large Cup"	Place an empty cup of 750ml in the espresso machine
B	"Grind 1 serving of coffee beans"	Grind a serving of coffee beans in the portafilter
I	"Insert Portafilter"	Insert the portafilter into the espresso machine
E	"Extract"	Extract the ground coffee in the portafilter
K	"Milk"	Pour 25ml of milk into the cup
F	"Foam"	Pour 25ml of milk foam into the cup
W	"Water"	Pour 25ml of hot water into the cup
G	"Give"	Give the cup to the adjacent Roomba
R	"Remove Portafilter"	Remove the portafilter from espresso machine
C	"Clean Portafilter"	Cleans the portafilter
	"Wait"	No command

Preparing a coffee must be done by following these steps:

- Insert a cup of the correct size (See formula to determine the size of a coffee)
- Grind the right amount of coffee beans
 - Solo = 1 serving
 - Doppio = 2 serving
 - Triplo = 3 serving
- Insert the portafilter
- If it is a coffee of type *inverted*, add the water, milk and/or milk foam before the extraction of the grinded coffee beans
- Extract the right amount of coffee
 - Ristretto = 1 extraction
 - Normale = 2 extraction
 - Lungo = 3 extraction
- If it is not a coffee of type *inverted*, add the water, milk and/or milk foam before the extraction of the grinded coffee beans
- Remove the portafilter
- Clean the portafilter
- Give the cup to the Roomba

The size of a coffee is determined by the following formula:

$$(\text{size} * 25\text{ml}) * (\text{length} * 0.5) + \text{SUM}(\text{ingredients})$$

F.A.Q. on the Espresso Machines

- The order of the ingredients is important, you always have to go in this order:
hot_water, milk and foam.
- The machine must be cleaned before preparing another coffee.
- Once the coffee is finished, it can be immediately given to the Roomba and cleaned afterwards.
- The espresso machine is limited to one coffee at a time. It has to give the current coffee to the Roomba before preparing another coffee.
- A cup transfer from the espresso machine to the Roomba must be synchronized. The Roomba must send the "T" command at the same time as the espresso machine sends the "G"

Output

The output of the program must be the series of commands to be executed by the Roomba and the espresso machines for the map and orders provided as input.

Each line must be identified first by the actor of the commands (Roomba or the espresso machine number) followed by commands separated by spaces.

Example

```
1 S B I E K G R C
@ R F F F L F F T R F F F G
```

Example with Solution

Map

```
#####
#1 @ #
### #
#A #
#####
```

Orders

Alexandre
eyJzaXplIjoiZG9wcGlvIiwibGVuZ3RoIjoibm9ybWFsZSIzImVudmVydGVkIjpmYWxzZSwiaW5ncmVkaWVudHMlOnsibWlsayI6NTAsImZvYW0iOiUwfx0 =

Output

```
1 M B B I E E K K F F G R C
@ L F R R T F F R F F R F F G
```


Explanation

The order corresponds to the following payload:

```
{
  "size": "doppio",
  "length": "normal",
  "inverted": false,
  "ingredients": {
    "milk": 50,
    "foam": 50
  }
}
```

The size of the coffee is 150ml so it requires a medium cup.

$$(2 * 25\text{ml}) * (2 * 0.5) + (50 + 50) = 150\text{ml}$$

It is not a coffee of type `inverted`, so you have to extract the coffee beans then add the ingredients.

1 M B B I E E K K F F G R C

For its part the Roomba must go to the espresso machine #1 and wait after the order. Since it begins facing north, it must turn to the left once and advance one square to be adjacent to the espresso machine. While waiting for the order, it can rotate in order to be in the right direction for his departure.

@ L F R R T F F R F F R F F G

The espresso machine can remove the filter holder and clean while the Roomba goes to Alexandre to give him his coffee once adjacent to him.

Evaluation

General (10%)

2.5% - Your code includes a file "run.sh" that takes the input parameters, compiles and executes your solution

2.5% - The code provided compiles and runs without errors

2.5% - Your program has a help menu (--help)

2.5% - Your program handles missing or invalid inputs

Espresso Machine (25%)

2.5% - Coffees are prepared with the right size of cup

2.5% - Coffees can be prepared in the "inverted" style

5.0% - The GSOFTE Menu coffees are prepared correctly

7.5% - Coffees are prepared correctly

7.5% - Prepared coffees do not wait at the espresso machine (so that they stay hot as long as possible!)

Roomba (20%)

2.5% - The Roomba retrieve synchronously coffee from the espresso machines

10.0% - Roomba visits all those who ordered

7.5% - The Roomba does not hit obstacles

Performance (45%)

5.0% - Easy Map

10.0% - Medium Map

15.0% - Hard Map

15.0% - Correction Map

The team that completes a problem with the least amount of commands will have 100% Bonus Performance. The other teams will be evaluated according to the best team.