JUMPYRINTH - CODING 100

"While doing his mission preparation tests, R-boy notices in the file he's reading that the data has been inserted in a mysterious order. Read the text with him and discover what's behind it."

The challenge provide us a *RULES.txt* file which contains the information about a language and a text file with the *labyrinth*.

To solve the challenge we wrote an interpreter for the described language, which follows the information in the RULES.txt file and print the final flag.
The only detail to which we must be careful is that the start symbol **$** appears more than once in the labyrinth

```
import string

grid = open("2c464e58-9121-11e9-aec5-34415dec71f2.txt", "r").readlines()

def readup(y, x):
  out = ""
  y = y-1
  while y >= 0:
    if grid[y][x] not in string.digits:
      break
    out = out + grid[y][x]
    y = y-1
  return int(out)

def readdown(y, x):
  out = ""
  y = y+1
  while y < len(grid):
    if grid[y][x] not in string.digits:
      break
    out = out + grid[y][x]
    y = y+1
  return int(out)

def readright(y, x):
  out = ""
  x = x+1
  while x < len(grid[0]):
    if grid[y][x] not in string.digits:
      break
    out = out + grid[y][x]
```

```
    x = x+1
  return int(out)

def readleft(y, x):
  out = ""
  x = x-1
  while x > 0:
    if grid[y][x] not in string.digits:
      break
    out = out + grid[y][x]
    x = x-1
  return int(out)

for y in range(len(grid)):
  for x in range(len(grid[y])):

    if grid[y][x] != "$":
      continue

    stack = []
    flag = []

    while True:

      if grid[y][x] == "@":
        break
      elif grid[y][x] == "#":
        break

      elif grid[y][x] == "$":
        y = y+1

      elif grid[y][x] == "(":
        flag = [stack.pop()] + flag
        x = x - readright(y, x)
      elif grid[y][x] == ")":
        flag = flag + [stack.pop()]
        x = x + readleft(y, x)
      elif grid[y][x] == "-":
        flag = flag[1:]
        y = y - readdown(y, x)
      elif grid[y][x] == "+":
        flag = flag[:-1]
        y = y + readup(y, x)
```

```
    elif grid[y][x] == "%":
      flag = flag[::-1]
      y = y+1

    elif grid[y][x] == "[":
      stack.append(grid[y][x+1])
      x = x+2
    elif grid[y][x] == "]":
      stack.append(grid[y][x-1])
      x = x-2
    elif grid[y][x] == "*":
      stack.append(grid[y-1][x])
      y = y-2
    elif grid[y][x] == ".":
      stack.append(grid[y+1][x])
      y = y+2

    elif grid[y][x] == "<":
      x = x - readright(y, x)
    elif grid[y][x] == ">":
      x = x + readleft(y, x)
    elif grid[y][x] == "^":
      y = y - readdown(y, x)
    elif grid[y][x] == "v":
      y = y + readup(y, x)

    else:
      continue

  flag = "".join(flag)
  if "{FLG:" in flag:
    print(flag)
```

The found flag is:

{FLG:H4ckItUpH4ckItInL33tM3B3g1n}