

Introduction to programming and simulation

Computers accomplish complex tasks by following simple instructions contained in a program. To help you to understand how programs work, you will try to guide a robot mouse to a piece of electronic cheese using only these six simple instructions:

Movement instructions

step forward
turn left
turn right

Control instructions

:label
goto *label*
if obstacle goto *label*

For example, this simple program directs the robot mouse to walk forward in a straight line.

```
:straightline
step forward
goto straightline
```

:straightline is a label. Labels are remembered as placeholders. This command causes the robot to take a single step, and then stop. **goto** commands cause the program to continue from the specified label.

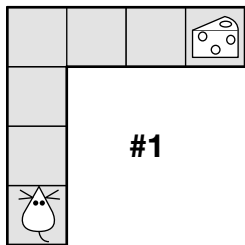
This program uses a conditional branch to avoid obstacles. Can it wander around a room?

```
:walk
if obstacle goto turn
step forward
goto walk
```

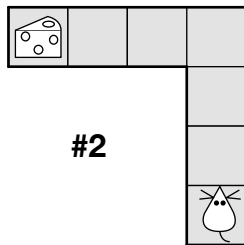
```
:turn
turn right
goto walk
```

Each of the labelled sections of this program is known as a sub-routine, or function, and accomplishes a particular task. Is it advisable to take a step without checking for obstacles?

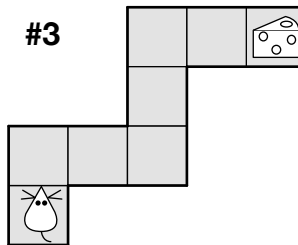
The best programs successfully accomplish tasks under a variety of conditions, and using the fewest instructions. Write one single program that will navigate the most possible mazes.



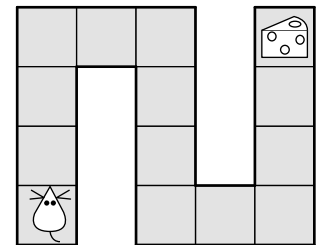
#1



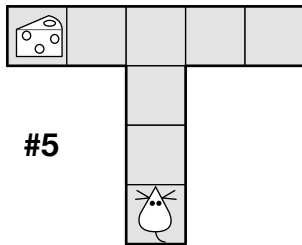
#2



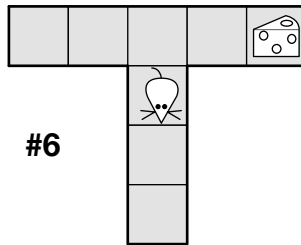
#3



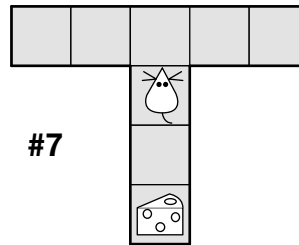
#4



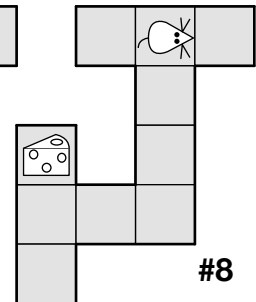
#5



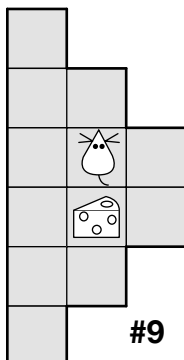
#6



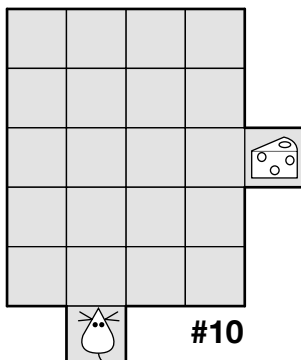
#7



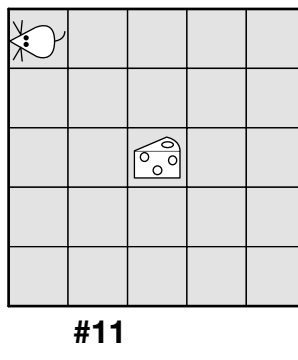
#8



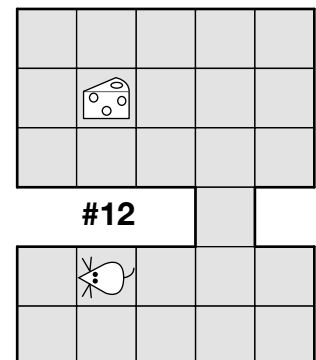
#9



#10



#11



#12

