

# Micromouse

*Micromouse* é um evento onde pequenos robôs competem para resolver labirintos. Começando em uma posição inicial, os robôs devem encontrar o objetivo, voltar à posição inicial e retornar ao objetivo. Quanto mais rápido melhor!

Os robôs não possuem conhecimento da estrutura do labirinto: as paredes só são encontradas quando o robô está suficientemente perto para senti-las ou quando ocorre colisão.

## Funcionamento

O robô começa em uma posição aleatória dentro do labirinto. Em cada rodada do jogo, o robô pode decidir qual ação realizará. Uma ação é realizada imprimindo um ou dois caracteres na saída padrão. As possíveis ações são:

1. `l` → rotacionar 90° para a esquerda (sentido anti-horário).
2. `r` → rotacionar 90° para a direita (sentido horário).
3. `m` → andar para frente.
4. `M` → correr para frente.
5. `s` → ativar sensor de paredes próximas.
6. `d` → ativar sensor de proximidade do objetivo.

Após realizar qualquer ação, o robô deve ler da entrada padrão o resultado obtido. Esse resultado é sempre um número inteiro, mas seu significado muda de acordo com a ação previamente executada.

**IMPORTANTE:** O jogador sempre deve realizar a leitura da entrada padrão após qualquer movimento. Caso essa leitura não seja feita, a correção do exercício não será executada corretamente e a pontuação não será atribuída ao jogador.

## Retornos das Ações

Quando o robô se move (`m`), o valor de retorno pode ser 0 (o robô conseguiu se mover), 1 (o robô bateu em uma parede) ou 4 (o robô encontrou o objetivo).

Quando o robô corre (`M`), o valor de retorno pode ser 1 (bateu em uma parede sem se mover), 2 (se moveu uma vez e depois bateu em uma parede) ou 3 (se moveu duas vezes).

A corrida (`M`) penaliza o robô caso ele colida com uma parede antes do movimento finalizar. Essa penalidade é equivalente à 3 movimentos bem-sucedidos.

Quando o robô ativa o sensor de paredes próximas (`s`), o valor de retorno é um número inteiro de 4 bits representando para quais direções o robô pode se mover. Por exemplo:

```
int retorno = 13; // 0b1101
int frente = (retorno << 0) & 1; // 1 (parede)
int direita = (retorno << 1) & 1; // 0 (livre)
int tras = (retorno << 2) & 1; // 1 (parede)
int esquerda = (retorno << 3) & 1; // 1 (parede)
```

Quando o robô ativa o sensor de distância (`d`), o valor de retorno é a distância da posição atual até o objetivo.

O uso de qualquer um dos dois sensores penaliza o robô em o equivalente à 10 unidades de tempo.

Quando robô rotaciona (`l` ou `r`), o valor de retorno é insignificante. **Contudo, é importante que o robô leia o valor de retorno.** Se o robô não ler o valor de retorno, a solução não será corrigida corretamente.

## Pontuação

A pontuação final é calculada pela quantidade de ações tomadas para finalizar o caminho *inicial* → *objetivo* → *inicial* → *objetivo* somadas com as penalidades aplicadas por colisões ou uso de sensores.

\  
rightlineAuthor: Bruno Ribas, Guilherme Puida e Thalisson Alves