

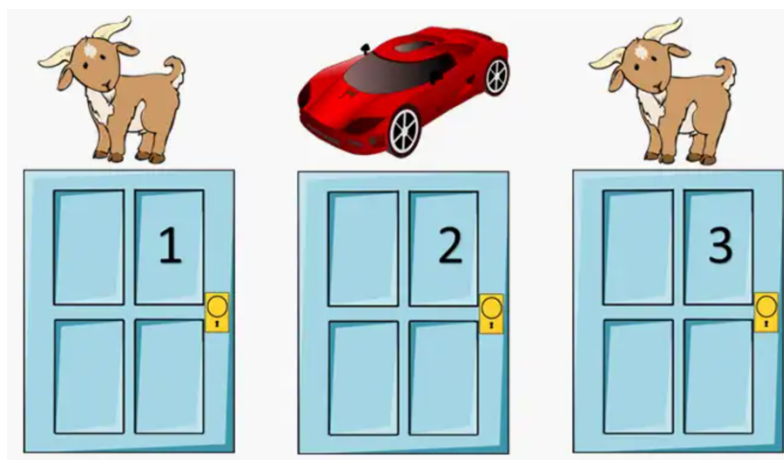


PUC Minas

O problema de Monty Hall, também conhecido por paradoxo de Monty Hall é um problema que surgiu a partir de um concurso televisivo dos Estados Unidos chamado *Let's Make a Deal*, exibido na década de 1970.



O jogo consistia no seguinte: Monty Hall, o apresentador, apresentava três portas aos concorrentes. Atrás de uma delas estava um prêmio (um carro) e, atrás das outras duas, dois bodes.



Na 1ª etapa o concorrente escolhe uma das três portas (que ainda não é aberta):

Na 2ª etapa, Monty abre uma das outras duas portas que o concorrente não escolheu, revelando que o carro não se encontra nessa porta e revelando um dos bodes;

Na 3ª etapa, Monty pergunta ao concorrente se quer decidir **permanecer com a porta que escolheu no início do jogo ou se ele pretende mudar para a outra porta que ainda está fechada**. Agora, com duas portas apenas para escolher – pois uma delas já se viu, na 2ª etapa, que não tinha o prêmio – e sabendo que o carro está atrás de uma das duas restantes, o concorrente tem que tomar a decisão.

No Brasil, os apresentadores de televisão Sérgio Malandro e Ratinho fizeram suas versões do quadro americano.



Construa um algoritmo em C para criar o jogo do Monty Hall. Você precisará utilizar a função **srand()** da biblioteca **<stdlib.h>** para inicializar o gerador de números pseudoaleatórios e a função **rand()** para gerar o número pseudoaleatório (sortear a porta premiada). Na Dica 2 você encontrará o código para realizar este sorteio.

Você também precisará utilizar a função **time()** para auxiliar na geração do número pseudoaleatório. A função **time()** retorna o número de segundos desde a época, geralmente 1 de janeiro de 1970. Quando usada dentro da função **srand()**, ela fornece uma maneira de inicializar o gerador de números pseudoaleatórios com uma semente diferente em cada execução do programa, baseada no tempo atual. Isso garante que a sequência de números gerada pela função **rand()** seja diferente em cada execução do programa, ajudando a evitar padrões repetitivos e tornando os números aparentemente mais aleatórios.

```
#include <stdlib.h>
#include <time.h> // Para auxiliar a função rand
```

Você precisará também de estruturas do tipo **do-while** e **if-else**. A solução final possui aproximadamente 60 linhas de código.

Dica 1: Você terá que criar as variáveis **portaPremiada**, **portaVazia** e **portaEscolhida**. Lembre-se que, caso o usuário opte por trocar de porta, você precisará atualizar a variável **portaEscolhida** para a porta que foi trocada.

Dica 2: Para sortear a **portaPremiada** (e também encontrar a **portaVazia** e trocar de porta) utilize:

```
// Definir a porta premiada

// Inicializa a semente para a função rand() usando o tempo atual
srand(time(NULL)); // Utilize uma única vez
```



PUC Minas

```
// Gera um número aleatório entre 1 e 3
int portaPremiada = rand() % 3 + 1;
// printf("\nA porta premiada é: %d", portaPremiada); //comente essa
linha para ficar divertido
```

Dica 3: Para ler a portaEscolhida (int), utilize:

```
scanf("%d", &portaEscolhida);
```

E para ler a opção de sim ou não: s/n (char), para trocar de porta, utilize:

```
char opcao;
scanf(" %c", &opcao);
```

O espaço em branco antes de %c na função scanf() é utilizado para ignorar quaisquer caracteres em branco (como espaços, tabulações, quebras de linha etc.) presentes no buffer de entrada antes do caractere que está sendo lido. Caso você tenha problemas com o buffer de entrada (stdin) durante a leitura de algum char, utilize a função `fflush(stdin)` ou `fflush(stdout)`; logo antes da função scanf().

Dica 4: Lembre-se de utilizar | | para testar condições com o operador **OR** (ou) ou && para o operador **AND** (e).

Simulação do jogo (sem trocar de porta: n):

```
=> Bem-vindo ao jogo Monty Hall!
=> Escolha uma porta (1, 2 ou 3): [portaEscolhida = 1]
(Sistema sorteia a porta premiada (1, 2 ou 3): [portaPremiada = 2]
(Abrir porta vazia que não pode ser a portaEscolhida nem a portaPremiada)
=> O apresentador abriu a porta 3 que está vazia. [portaVazia = 3]
=> Você deseja trocar para a outra porta? (s/n): n
=> Você perdeu. A porta premiada era a 2.
```

Simulação do jogo (trocando de porta: s):

```
=> Bem-vindo ao jogo Monty Hall!
=> Escolha uma porta (1, 2 ou 3): [portaEscolhida = 1]
(Sistema sorteia a porta premiada (1, 2 ou 3): [portaPremiada = 2]
(Abrir porta vazia que não pode ser a portaEscolhida nem a portaPremiada)
=> O apresentador abriu a porta 3 que está vazia. [portaVazia = 3]
=> Você deseja trocar para a outra porta? (s/n): s
(Trocar de porta que não pode ser a portaEscolhida originalmente nem a portaVazia.
=> Você trocou para a porta 2.
=> Você ganhou! Parabéns!
```