



Professores: José Américo (jose.americo@ifsp.edu.br)
Samuel Martins (samuel.martins@ifsp.edu.br)

Lab 07 – Jogando Cartas

1) Descrição

Dado um baralho com n cartas numeradas de 1 à n , no qual todas as cartas estão ordenadas, com a carta 1 no topo do baralho e a carta n no fundo.

A seguinte operação é realizada, enquanto houver pelo menos duas cartas no baralho:

- Retire a carta que está no **topo** do baralho, exiba-a e coloque a **próxima carta** no fundo do baralho;

Sua tarefa é escrever um programa que encontra a **sequência de cartas descartadas** e a **carta remanescente** para um dado baralho com n cartas.

É obrigatório o uso da estrutura de dados **Fila** para a resolução dos problema.

2) Entrada

Cada linha de entrada (exceto a última) contém um número $n \leq 50$. A última linha contém 0 e deve terminar o programa.

3) Saída

Para cada entrada deve se produzir duas linhas de saída. A primeira linha representa a **sequência de cartas descartadas** e a segunda linha informa a **última carta remanescente no baralho**.

Obs: Nenhuma linha terá espaços à esquerda ou à direita. Veja o exemplo para o formato esperado.

4) Exemplos

Entrada	Saída
7	Cartas jogadas: 1, 3, 5, 7, 4, 2
19	Carta restante: 6
10	Cartas jogadas: 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 4, 8, 12, 16, 2, 10, 18, 14
6	Carta restante: 6
0	Cartas jogadas: 1, 3, 5, 7, 9, 2, 6, 10, 8
	Carta restante: 4
	Cartas jogadas: 1, 3, 5, 2, 6
	Carta restante: 4

5) Exemplos

- Para **compilar** seu código no terminal:
☐ `gcc lab.c -o lab`
- **-o** significa output. Ele é responsável por gerar o binário do seu programa para execução. É OBRIGATÓRIO que o arquivo tenha a função **main**;
- Logo, o que você está dizendo é: *compile o código **lab.c** com o compilador **gcc**, gerando o executável (saída) **lab***;
- Para **executar** seu programa:
☐ `./lab`
- Você pode baixar os arquivos de casos de teste do run.codes e executá-los manualmente:
☐ `./lab < 01.in`
- A diretiva `<` redireciona o conteúdo do arquivo `01.in` para o terminal, cujas entradas/dados serão lidos pelo `scanf`;
- Você pode ainda redirecionar a saída impressa no terminal para um arquivo:
☐ `./lab < 01.in > 01.res`
- Por fim, você poder comparar sua resposta com o gabarito (resultado do caso de teste), fazendo
☐ `diff 01.res 01.out`
☐ onde `01.out` é a saída esperada para a entrada `01.in`

6) Observações Gerais

- A nota é dada pelo **número de casos de teste acertados**;
- É obrigatório usar **Fila**, caso contrário, a nota será **ZERO**.
- É obrigatório desalocar a lista corretamente. Caso contrário, pontos serão descontados.
- Códigos com **erros de compilação e execução**, tais como Segmentation Fault, **serão considerados errados**;
- Utilize ***return 0***; na main de seu programa;
- Qualquer tentativa de fraude, plagio e afins, corresponderá em **nota ZERO** para os envolvidos;