

---

## Laboratório #4 - lab4.c

Triplas

---

**Prof. Valério Rosset**

### 1 DEFINIÇÃO DO PROBLEMA

Para aplicar seus conhecimentos de Algoritmos e Estrutura de Dados, você decidiu desenvolver um jogo. O objetivo do jogo é que o usuário forme triplas de números que juntas somem 10. Seu jogo funcionará assim:

- Inicialmente, uma sequência de 10 números inteiros entre 0 e 9 é fornecida ao jogador. Por exemplo: **1 3 2 8 7 6 4 9 5 0**
- Em seguida, o jogador terá 5 jogadas. Em cada jogada, o jogador fornece um número para ser inserido na lista e a posição da lista na qual o número deverá ser inserido. Por exemplo: **5 2**
- Após a inserção, a lista é atualizada com o novo valor. Observe que os elementos da lista são deslocados para a **direita** para que a inserção ocorra na posição desejada: **1 3 5 2 8 7 6 4 9 5 3**
- Observe que agora existe uma tripla que soma 10: **1 3 5 2 8 7 6 4 9 5 3**
- Agora, todos os elementos que formam a tripla de soma 10 são removidos da lista: **1 8 7 6 4 9 5 3**
- O objetivo do jogador é remover todos os elementos da lista.

### TAREFA A

Desenvolva um algoritmo em linguagem C - **utilizando lista dinâmica duplamente encadeada** - que forneça uma lista com a sequência inicial do jogo. Após cada jogada,

seu algoritmo deverá imprimir a lista atualizada **após** as verificações de triplas e as remoções necessárias. Ao final da execução, seu algoritmo deve informar se o jogador ganhou ou perdeu o jogo. Note que o jogador ganha o jogo somente se esvaziar a lista.

Seu algoritmo deve ser capaz de realizar as seguintes operações:

- Inserção: Seu algoritmo deve permitir a inserção de números de acordo com a posição informada pelo jogador. Note que a posição é calculada a partir do primeiro elemento da lista. Esse primeiro elemento representa o primeiro número da sequência inicial.
- Remoção: Seu algoritmo deve permitir a remoção de elementos em qualquer posição.
- Análise: Após cada jogada, seu algoritmo deverá buscar por triplas de soma 10 e realizar as devidas remoções. Note que uma jogada não implica necessariamente em remoção.

## ENTRADA

A entrada contém um único conjunto de teste, que deve ser lido do dispositivo de entrada padrão (normalmente o teclado). O conjunto de teste é composto por 6 linhas.

1. A primeira linha contém uma sequência de 10 números iniciais.
2. Em cada uma das 5 linhas seguintes são dadas as informações da jogada. Cada linha contém um valor inteiro  $P$ ,  $1 \leq P \leq 9$ , que representa o número a ser inserido, seguido de um espaço e a posição desejada de inserção. Note que não é possível que o jogador insira o número 0 e que a primeira posição da lista é a posição 0.

Para as operações de entrada, considere a sintaxe:

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

```
scanf("%d %d",&variavel1, &variavel2);
```

## SAÍDA

Como saída padrão, seu programa deve imprimir a cada jogada a lista atual. Ao final das jogadas, seu programa deverá imprimir se o jogador ganhou ou perdeu.

As mensagens de vitória ou derrota são:

1. Ganhou: "ganhou".
2. Perdeu: "perdeu".

Para as operações de saída, considere a sintaxe:

```
printf("instrucao de saida\n");
```

```
printf("%d ", itemdalista);
```

```
printf("%d\n", ultimoitemdalista);
```

Entrada	Saída
1 3 2 8 7 6 4 9 5 3	1 3 2 8 7 6 4 9 5 3
5 2	1 8 7 6 4 9 5 3
2 7	1 8 7 6 4 9
1 2	7 6 4 9
1 0	1 7 6 4 9
2 0	6 4 9
	perdeu

Entrada	Saída
1 6 4 3 2 7 0 5 3 9	1 6 4 3 2 7 0 5 3 9
2 8	6 4 3 2 7
1 0	6 4 3
1 0	4
4 0	4 4
2 0	ganhou

### OBSERVAÇÕES

- É fundamental que o programa imprima apenas o valor de saída.
- Os exercícios devem ser postados no site <http://judge.sjc.unifesp.br>.
- Cada aluno deverá postar apenas um exercício dentro da data limite.
- A resolução submetida deve utilizar a estrutura de dados definida no enunciado do exercício.
- Os plágios serão penalizados. Como não será possível saber quem copiou, todos os casos envolvidos receberão punições iguais, portanto, não divulgue seu código.