

UNIFEI	Universidade Federal de Itajubá Instituto de Engenharia de Sistemas e Tecnologias da Informação - IESTI
	7º Laboratório de Estruturas de Dados Profa. Thatyana de Faria Piola Seraphim

O objetivo dessa atividade é simular uma partida de dominó entre 2 computadores usando ponteiros. As 3 variáveis `prijog1`, `prijog2` e `priMesa` serão definidas como global para todas as funções desse programa, sendo que representam o endereço de memória do primeiro elemento da lista do jogador1; o endereço de memória do primeiro elemento da lista do jogador2 e o endereço de memória do primeiro elemento da lista da mesa respectivamente.

```
typedef struct no{
    int face1;
    int face2;
    struct no* prox;
} noPeca;
noPeca* priJog1 = NULL;
noPeca* priJog2 = NULL;
noPeca* priMesa = NULL;
```

a) Faça uma função chamada **inicializa** que inicialmente vai imprimir todas as peças do jogo do dominó na tela do computador: (0,0); (0,1); (0,2); (0,3); (0,4); (0,5); (0,6); (1,1); (1,2)...(1,6); (2,2); (2,3)...(2,6);(3,3)...(3,6), ... (6,6)

b) Após o item (a) modifique a função **inicializa** para criar dinamicamente um `noPeca` e inserí-lo aleatoriamente no início de uma das listas (`priJog1` e `priJog2`). A peça (6,6) deve ser inserida no início da lista da mesa. Para evitar que um dos jogadores tenha mais peças que o outro, deve haver um contador para cada lista. Quando um dos contadores atingir 14 peças, somente a outra lista pode receber novas inserções. No código abaixo as variáveis `c1` e `c2` indicam a contagem da lista do jogador1 e do jogador2 respectivamente. A variável `sor` indica em qual lista será inserida (0 indica inserção no jogador1 e 1 indica inserção no jogador 2).

```
if(c1==14) {
    sor=1;
}else{
    if(c2==14){
        sor=0 ;
    }else{
        sor=rand() % 2;
    }
}
if((i==6) && (j==6)){
    //insere em mesa ...
}else{
    if(sor==0){
        //insere em jog1 ...
        c1++;
    }else{
        c2++;
        //insere em jog2 ...
    }
}
```

c) Modifique a função `inicializa` para retornar o se o

jogador 1 tem 14 peças ou retornar 1 se o jogador 2 tem 14 peças, pois um deles tem apenas 13 peças. Isso aconteceu porque a peça (6,6) foi inserida na lista da mesa.

d) Após o item (c) crie uma função chamada **imprime** que recebe um ponteiro de uma peça que representa o início de uma lista. Essa função deve imprimir, enquanto o ponteiro não for NULL, as faces 1 e 2 da peça. Para testar a função construa o programa abaixo.

```
int main(int argc, char * argv[]){
    inicializa();
    printf("jog1=");
    imprime(priJog1);
    printf("jog2=");
    imprime(priJog2);
    printf("mesa=");
    imprime(priMesa);
    return 0;
} //main
```

e) Após o item (d) crie uma função chamada **removejogador1** que recebe um número de face que deve ser removido na lista do jogador1. Esta função deve retornar o ponteiro da primeira peça que tenha `face1` ou `face2` com o mesmo número ou NULL se a face não existe na lista do jogador1.

f) Após o item (e) crie uma função chamada **removejogador2** que recebe um número de face que deve ser removido na lista do jogador2. Esta função deve retornar o ponteiro da primeira peça que tenha `face1` ou `face2` com o mesmo número ou NULL se a face não existe na lista do jogador2.

g) Após o item (f) crie uma função chamada **insereInicioMesa** que recebe como parâmetro o ponteiro de uma peça que será inserida no início da lista da mesa.

h) Após o item (g) crie uma função chamada **insereFimMesa** que recebe como parâmetro o ponteiro de uma peça que será inserida no final da lista da mesa.

i) Crie um programa **main** onde as inicializações das variáveis guardam as faces disponíveis na extremidade da lista da mesa: `ini=6` e `fim=6`. A variável `Jogador` recebe a inicialização do jogador que tem a maior quantidade de peças. Uma variável `peca` guarda a peça que será removida da lista de um dos jogadores e inserida na lista da mesa. O programa deve se manter enquanto nenhuma das listas do jogador 1 ou jogador 2 estiver vazia. Lembrando que após a jogada do jogador 1 deve passar a vez para que o jogador 2 possa jogar.