

## Universidade da Beira Interior Departamento de Informática

1. Considere as seguintes estruturas de dados relativas ao sistema de informação utilizado por uma empresa de jogos *online*, para registar a atividade dos seus utilizadores. A empresa disponibiliza continuamente um conjunto de jogos, sendo todos eles apenas para 2 jogadores e de resultado “vitória/derrota”.

```
typedef struct JOGADOR{
    int codigo; //critério de ordenação na lista
    char nome[80];
    struct JOGADOR *nseg;
    Participacao *jogos;
}Jogador;

typedef struct PARTICIPACAO{
    Jogo *jogo;
    struct PARTICIPACAO *nseg;
}Participacao;

typedef struct JOGO{
    long int data; //(em milissegundos, desde 1/1/1970, 00:00, chave de ordenação)
    Jogador *jogador1, *jogador2; (apontadores para os respetivos jogadores)
    int resultado; //(1 ou 2 consoante da vitória ser do 1º ou 2º jogador)
    struct JOGO *nseg[10]; //lista de salto
    int nivel; //nível na lista de salto
}Jogo;
```

**1a) (3 valores)** Mediante as estruturas apresentadas, esquematize a organização da informação resultante do armazenamento dos seguintes itens:

- Jogador 3, José Andrade;
- Jogador 2, André Almeida;
- Jogador 4, Andreia Soares;
- Jogador 1, Paula Correia;
  
- Jogo data 12345678, Jogador 1 e Jogador 3, Resultado 1
- Jogo data 12567821, Jogador 2 e Jogador 3, Resultado 2
- Jogo data 23456789, Jogador 2 e Jogador 3, Resultado 1
- Jogo data 32345678, Jogador 1 e Jogador 3, Resultado 2
- Jogo data 52345678, Jogador 3 e Jogador 1, Resultado 1

**1b) (3 valores)** Implemente uma função que imprima o nome do campeão, isto é, do jogador que venceu mais jogos desde sempre.

**Protótipo:** void mostraCampeao(Jogador \*J);

**1c) (3 valores)** Codifique uma função que conte o total de jogos registados no sistema após uma determinada data.

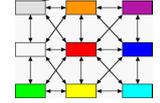
**Protótipo:** int totalJogos(Jogo \*L, long int data);

**1d) (3 valores)** Implemente uma função que imprima o nome dos jogadores que perderam todos os jogos em que participaram.

**Protótipo:** void listaPerdedores(Jogador \*J);

**1e) (3 valores)** Implemente uma função que remove um jogador do sistema, bem como todos os jogos em que ele participou. A função deve retornar um apontador para a lista dos jogadores (atualizada). No caso da lista de jogos, o apontador atualizado deve ser retornado através do parâmetro por referência.

**Protótipo:** Jogador\* removeJogador(Jogador \*J, int codigoJogador, Jogo \*\*L);



2) A empresa “Confusao” trata os assuntos pendentes de uma forma original, dando prioridade aos mais recentes. Assim, implementou uma estrutura do tipo “Pilha”, que serve para registrar os assuntos à espera de ser tratados.

```
typedef struct STACK{
    int hora, minuto, segundo; //momento de entrada na pilha
    char nomeCliente[80];
    int codigoFuncionario;
    struct STACK *nseg;
} Stack;
```

2a) (3 valores) Implemente uma função que retira da estrutura os elementos há mais que “n” segundos à espera de ser atendidos, e os transfere para uma nova pilha, chamada “Atrasados”. Os elementos deverão ficar em “Atrasados” na mesma ordem que estavam na pilha original. A função deverá retornar um apontador atualizado para a cabeça da pilha “Atrasados”, enquanto a pilha original deverá ser atualizada através do parâmetro por referência.

```
Protótipo: Stack* transfereAtrasados(Stack **P, int n);
```

Considere que tem à sua disposição as seguintes funções:

```
Stack * push(Stack *F, Stack *nv);
//insere um novo elemento numa fila
```

```
Stack * pop(Stack **F);
//retira um elemento da fila
```

```
int totalSegundos(int hora1, int minuto1, int segundo1, int hora2, int minuto2, int segundo2); //devolve o total de segundos
entre “hora1:minuto1:segundo1” e “hora2:minuto2:segundo2”.
```

3. (2 valores) Considere a seguinte estrutura de dados, do tipo “Pilha”, e as respectivas funções push() e pop(). Suponha que a Pilha “P” contém os seguintes elementos: P→’a’→’b’→’i’→’o’→’e’.

```
typedef struct STACK{
    char ID;
    struct STACK *nseg;
}Stack;
```

Considere que Stack \*V=NULL, \*C=NULL;

Qual o conteúdo das variáveis “P”, “V” e “C” após a execução da seguinte instrução?

```
P=funcaoEstranha(F, &V, &C);
```

```
Stack * funcaoEstranha(Stack *F, Stack **V, Stack **C){
    Stack * aux;
    while (F!=NULL){
        aux=pop(&F);
        if ((aux->ID=='a')||((aux->ID=='e')||((aux->ID=='i'))))
            V=push(C, aux);
        else
            C=push(V, aux);
    }
    return(F);
}
```