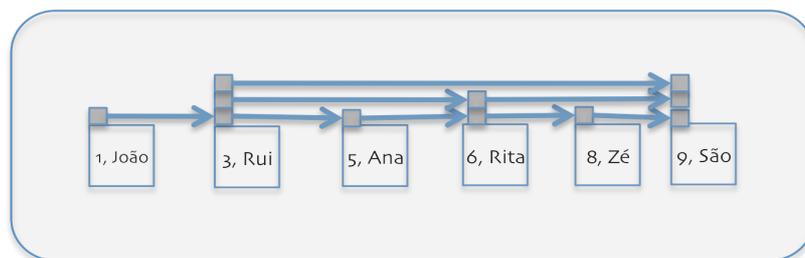


Universidade da Beira Interior
Departamento de Informática
2010/11

1. Considere a seguinte lista de salto com 3 níveis, cujo critério de ordenação é o atributo ID.
 - a. **(1.5 valores)** Exemplifique uma procura (através do atributo ID), em que a utilização do mecanismo de listas de salto se revelasse desvantajosa em relação à pesquisa sequencial.
 - b. **(1.5 valores)** Enumere as sequências de nós visitados, caso se procure: 1) pelo elemento com nome='Rita' e 2) pelo elemento com ID=8.



2. A seguinte estrutura de dados guarda os resultados relativos à eleição da direcção de um determinado clube recreativo. Considere que cada nó representa um eleitor (BI), o total de votos que este detém (de acordo com a sua antiguidade) e o BI do elemento em quem votou.

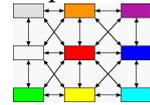
```
typedef struct ELEICAO{
    int BI;          //BI do sócio-votante
    int totVotos;   //total de votos que o votante detém
    int BIVoto;     //BI do elemento em que o sócio votou
    struct ELEICAO *nseg;
}Eleicao;
```

- a. **(4 valores)** Implemente uma função que verifique se as eleições foram fraudulentas, isto é, se existiram elementos que votaram mais que uma vez. A função deverá devolver 1 em caso afirmativo ou 0 em caso contrário.

```
int fraudulentas(Eleicao* L);
```

- b. **(3 valores)** Implemente uma função que conte o total acumulado de votos de todos os eleitores.

```
int totalVotos(Eleicao* L);
```



- c. **(3 valores)** Implemente uma função que devolva a percentagem de votos que um determinado elemento teve, em relação ao total calculado na alínea anterior.

```
float percentagem(Eleicao* L, int BI);
```

- d. **(4 valores)** Implemente uma função que devolva o BI do vencedor das eleições.

```
int vencedor(Eleicao* L);
```

3. **(3 valores)** Considere que os nós relativos à seguinte estrutura de dados podem ser manuseados através de uma fila ou de uma pilha, com as respectivas funções pushF() e popF() para filas ou pushP() e popP() para pilhas. Adicionalmente, imagine que tem disponível a função "Nodo* makeNode(char c)" que aloca espaço para um novo nó e o preenche com o caracter passado por parâmetro.

```
typedef struct NODO{
    char x;
    struct NODO *nseg;
}Nodo;
```

Transcreva o *output* do seguinte programa:

```
int main(){
    Nodo * f=NULL, *p=NULL, *r;
    f=pushF(f,makeNode('a'));
    f=pushF(f,makeNode('b'));
    p=pushP(p,makeNode('a'));
    p=pushP(p,makeNode('b'));
    p=pushP(p,makeNode('c'));
    f=pushF(f,makeNode('c'));
    r=popF(&f); free(r);
    r=popP(&p); free(r);
    f=pushF(f,makeNode('d'));
    p=pushP(p,makeNode('e'));
    while(f!=NULL){
        r=popF(&f);
        printf("%c",r->x);
        free(r);
    }
    printf("\n");
    while(r!=NULL){
        r=popP(&p);
        printf("%c",r->x);
        free(r);
    }
    return(0);
}
```