

Programação – Exame da Época Normal

3 de julho de 2023 – Duração: 120 minutos

LEI, LEI-PL, LEI-CE

1. Considere as seguintes definições:

```
typedef struct {int h, m;} tempo;
```

```
typedef struct Voo voo;  
struct Voo{  
    char codigo[6];           // código do voo  
    char destino[4];          // código do destino  
    int emTerra;              // sim = 1; não = 0;  
    tempo t;                  // hora de partida  
};
```

Para cada dia, o serviço de tráfego aéreo de um aeroporto, regista, num ficheiro binário, os dados dos voos que descolam desse local. O ficheiro é constituído, exclusivamente, por estruturas do tipo *voo*, uma para cada voo. O campo *emTerra* indica se o voo ainda está no aeroporto ou já partiu. Nesta altura todos os voos têm este campo com valor 1, ou seja, todos os voos estão registados como ainda estando em terra. No lado direito, pode-se ver um exemplo de um ficheiro com sete voos. As estruturas não estão ordenadas por nenhum critério em particular.

Escreva uma função em C que efetue as seguintes operações:

- Altere, no ficheiro binário, o campo *emTerra* (de 1 para 0), para todos os voos cuja hora de partida já passou.
- Crie um *array* dinâmico, de estruturas do tipo *voo*, para onde deve copiar todos os voos que tenham como destino um determinado aeroporto. Considerando o ficheiro exemplificado, e assumindo que o aeroporto destino tem código DUB, deveria ser criado um *array* dinâmico com 2 estruturas (códigos de voo RA342 e RA343).

A função recebe, como parâmetros, o nome do ficheiro a atualizar, uma estrutura do tipo *tempo* contendo a hora atual, o código do destino e o endereço de uma variável inteira, onde deve colocar a dimensão do *array* dinâmico. A função devolve, como resultado, o endereço inicial do *array*. Em caso de erro, a função devolve NULL.

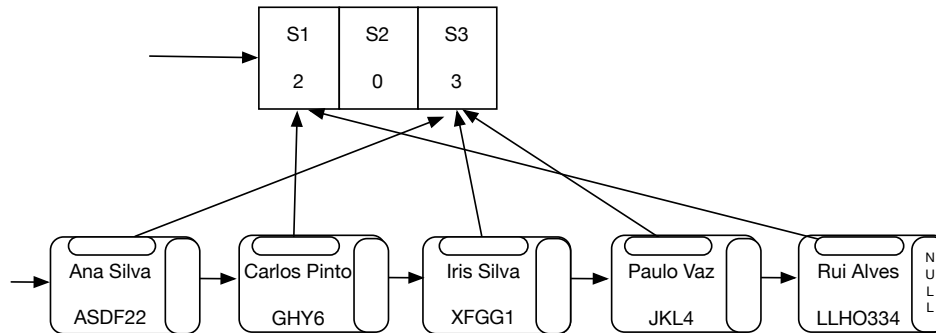
ST123
FNC
1
07 25
AP434
FAO
1
18 35
AP435
FNC
1
09 15
AP535
OPO
1
11 45
RA342
DUB
1
12 15
EJ262
BCN
1
10 25
RA343
DUB
1
11 15

[Cotação: 30%]

Considere as seguintes definições:

```
struct sala{
    char id[10];
    int total;
};

typedef struct pessoa no, *pno;
struct pessoa{
    char nome[100], num[20];    // Nome e número do aluno
    struct sala* p;    // Ptr. para a sala onde o aluno fará o teste
    pno prox;
};
```



A informação dos alunos que vão realizar um teste está armazenada numa lista ligada simples, constituída por nós do tipo *no*. Esta lista está organizada alfabeticamente pelo nome do aluno. Existe ainda um array dinâmico, constituído por estruturas do tipo *struct sala*, contendo informação sobre as salas onde vai ser realizada a prova. Cada estrutura do array contém a identificação da sala e um contador indicando quantos alunos farão a prova nesse local. O array não está ordenado de acordo com nenhum critério. Cada nó da lista ligada (i.e., cada aluno) tem um campo *p*, referenciando a sala a que está associado. No exemplo da figura pode verificar-se que existem 3 salas e 5 alunos, estando o aluno Carlos Pinto associado à sala S1.

2. Escreva uma função em C que imprima na consola o nome e número dos alunos que vão realizar a prova na sala com mais pessoas. Caso exista mais do que uma sala com um número máximo de alunos, a função não escreve nada. A função recebe, como parâmetros, o endereço inicial da lista de alunos, o endereço inicial e a dimensão do array de salas.

[Cotação: 20%]

3. Escreva uma função em C que adicione um aluno à estrutura dinâmica e o associe a uma das salas. A função tem o seguinte protótipo:

```
pno adicionaAl(pno lista, struct sala *s, int tot, char *nome, char *num);
```

A função recebe o início da lista ligada, o endereço inicial e a dimensão do array de salas e os dados do novo aluno. Deve adicionar o aluno à lista ligada (mantendo a ordem alfabética) e associá-lo à sala que tenha menos alunos nessa altura. Caso existam várias salas com um número mínimo de alunos, a função deve escolher uma delas. A função devolve um ponteiro para o início da lista depois da atualização. Ao responder a esta pergunta, pode assumir que o array dinâmico tem, no mínimo, 1 sala.

[Cotação: 40%]

Programação – Exame da Época Normal

3 de julho de 2023 – Duração: 120 minutos

LEI, LEI-PL, LEI-CE

Nome: _____ Número: _____

4. Funções Recursivas

Qual é o output na consola quando é executado o seguinte programa em C?

```
#include <stdio.h>

#define TAM 5

int fRec(int *a, int n){
    if(n<0 || n>=TAM)
        return n-1;
    else if(a[n] <= 0)
        return a[n]-n;
    else {
        printf("%d\t", a[n]);
        return fRec(a, n+a[n]);
    }
}

int main(){
    int tab[TAM] = {1, 2, 3, -4, -5};

    printf("%d\n", fRec(tab,0));
    return 0;
}
```

Resposta:

[Cotação: 10%]