

INF1007 - PROGRAMAÇÃO II

LISTA DE EXERCÍCIOS 6

1. Escreva um programa que implemente o jogo “Qual é o filme?”. Nesse o jogo, o jogador 1 deve digitar o nome de um filme e em seguida 5 pistas descritivas sobre o filme. Depois, o jogador 2 tem 5 chances para tentar adivinhar o nome do filme. Inicialmente somente a primeira pista é mostrada para o jogador 2, mas sempre que ele errar o nome do filme, uma nova pista é exibida na tela e o jogador pode tentar adivinhar o nome filme novamente.

A pontuação do jogador 2 depende do número de pistas que ele utilizou para descobrir o nome do filme:

1 pista: +100

3 pistas: +40

5 pistas: +10

2 pistas: +60

4 pistas: +20

Na implementação do jogo, você deve utilizar um vetor de ponteiros para cadeias de caracteres para armazenar as pistas sobre o filme, onde cada cadeia de caractere do vetor deve ser alocada dinamicamente com o tamanho exato para armazenar a pista fornecida pelo jogador 1.

O seu programa deve implementar pelo menos as seguintes funções:

- `lerPistas`: a função recebe como parâmetro um vetor de ponteiros para cadeias de caracteres. A função deve pedir para o jogador digitar as 5 pistas e armazenar as pistas dentro do vetor de ponteiros, onde cada elemento do vetor deve ser alocado dinamicamente com espaço do tamanho exato para armazenar as pistas;
- `mostrarPistas`: a função recebe como parâmetro um vetor de ponteiros para cadeias de caracteres e um inteiro indicando o número de pistas que devem ser exibidas. A função deve exibir na tela o número especificado de pistas;
- `comparaFilme`: a função recebe como parâmetro duas cadeias de caracteres e deve retornar 1 caso o nome dos filmes seja igual e 0 caso sejam diferentes;

Além das funções sugeridas, você é livre para criar mais função para facilitar a implementação do programa.

Implemente também a função principal do programa para permitir que 2 jogadores possam jogar o jogo “Qual é o filme?”. Ao final do jogo, o programa deve exibir a pontuação dos jogadores.

Para impedir que o jogador 2 possa ler o nome do filme e as pistas digitadas pelo jogador 1, você pode utilizar o comando:

```
system("CLS");
```

Este comando apaga todo o conteúdo existente na tela. Para utilizar ele você deve incluir a biblioteca `stdlib.h`.

O exemplo abaixo mostra como seria uma execução do jogo:

```
Jogador 1 - Digite o nome do filme: Matrix
```

```
Jogador 1 - Digite a pista 1: Humanidade dominada por maquinas
```

```
Jogador 1 - Digite a pista 2: Siga o coelho branco
```

```
Jogador 1 - Digite a pista 3: Oraculo
```

```
Jogador 1 - Digite a pista 4: Keanu Reeves
```

```
Jogador 1 - Digite a pista 5: Neo
```

```
Pista 1: Humanidade dominada por maquinas
```

```
Jogador 2 - Qual o nome do filme? Exterminador do Futuro
```

```
Você errou!
```

```
Pista 1: Humanidade dominada por maquinas
```

```
Pista 2: Siga o coelho branco
```

```
Jogador 2 - Qual o nome do filme? Matrix
```

```
Você acertou! +60 pontos!
```

Opcionalmente, você pode implementar o jogo completo, incluindo turnos, pontuação de cada jogador e permitindo que os jogadores possam continuar jogando até que um deles decida terminar o jogo.

2. Crie um programa para gerenciar as inscrições para o próximo vestibular da PUC-Rio. Para se inscrever no vestibular, os candidatos fornecem as seguintes informações:

Candidato
Nome
Curso Desejado
Endereço
Data Nascimento

Data
Dia
Mês
Ano

Onde:

- **Nome:** é um ponteiro para uma cadeia de caracteres alocada dinamicamente contendo o tamanho exato para o nome do candidato;
- **Curso Desejado:** é um ponteiro para uma cadeia de caracteres alocada dinamicamente contendo o tamanho exato para o curso desejado pelo candidato;
- **Endereço:** é um ponteiro para uma cadeia de caracteres alocada dinamicamente contendo o tamanho exato para o endereço do candidato;
- **Data Nascimento:** é uma estrutura do tipo Data que contém dia, mês e ano;

Escreva um programa que defina as estrutura acima e implemente as seguintes funções:

- `inicializa`: a função recebe como parâmetro um vetor de ponteiros de estruturas do tipo `Candidato` e também o tamanho máximo de candidatos. A função deve atribuir `NULL` para todos os elementos do vetor.
- `criaCandidato`: a função recebe como parâmetro três cadeia de caracteres representando o nome, curso desejado e endereço do candidato. A função recebe também uma estrutura do tipo `Data` contendo a data de nascimento do candidato. A função deve alocar dinamicamente uma estrutura do tipo `Candidato` e armazenar nela as informações do candidato recebidas pela função. Lembre-se de alocar dinamicamente as cadeias de caracteres da estrutura com o tamanho exato necessário para armazenar as informações.
- `novoCandidato`: a função recebe como parâmetro um vetor de ponteiros de estruturas do tipo `Candidato` e também o tamanho máximo de candidatos. A função deve ler as informações do candidato (nome, curso, endereço e data de nascimento) e armazena-las no vetor de candidatos em um registro vazio (igual a `NULL`). Obrigatoriamente você deve utilizar a função `criaCandidato` para criar um novo candidato. Se não existirem mais espaços livres no vetor de candidatos, a função deve exibir uma mensagem de erro.

- `exibeInscritos`: a função recebe como parâmetro um vetor de ponteiros de estruturas do tipo `Candidato` e também o tamanho máximo de candidatos. A função deve exibir na tela todas as informações sobre todos os candidatos inscritos no vestibular.

Além das funções sugeridas, você é livre para criar mais função para facilitar a implementação do programa.

Após implementar as funções, escreva a função principal do programa que permita que o usuário cadastre um número indeterminado de candidatos (até que o usuário indique que ele deseja parar de cadastrar candidatos). Em seguida, o programa deve exibir a lista de todos os candidatos inscritos para o vestibular.

Lembre-se de liberar toda a memória alocada dinamicamente quando ela não for mais ser utilizada pelo programa.