

Prova 2

Algoritmos e Estruturas de Dados I

Professor: Pedro O.S. Vaz de Melo

Nome: _____
escrevendo o meu nome eu juro que seguirei o código de honra

Código de Honra para este exame:

- Não darei ajuda a outros colegas durante os exames, nem lhes pedirei ajuda;
- não copiarei nem deixarei que um colega copie de mim;
- não usarei no exame elementos de consulta não autorizados.

Informações importantes:

- Em questões que pede um **programa**, este deve ser completo, com bibliotecas (incluindo, quando necessário, a biblioteca `prova2.h`), função `main`, etc. Se deve ser feita uma **função**, somente a função é suficiente. Se deve ser feito um **procedimento**, somente o procedimento é suficiente.
- A interpretação das questões da prova faz parte do critério de avaliação. Caso tenha dúvida sobre a sua interpretação de uma determinada questão, escreva as suas suposições na resolução da mesma.
- As funções implementadas no módulo `prova2.h` podem ser usadas em **qualquer** exercício da prova. Além disso, se você usar uma função do módulo `prova2.h`, considere que ela está implementada de forma correta.

Referências:

Função/Operador	Descrição	Biblioteca	Exemplo
<code>float exp(float x)</code>	retorna e^x	<code>math.h</code>	<code>exp(1)</code> retorna $e^1 = 2.71828$

1. (3 points) Defina um novo tipo de dados chamado `Video` para representar os metadados de um vídeo de um sistema online de *streaming* de vídeos (ex: Youtube). O seu tipo de dados deve ter os seguintes campos: `id` (inteiro), que é o identificador do vídeo; `likes` (inteiro), que é o número de *likes* que o vídeo recebeu; `ndislikes` (inteiro), que é o número de *dislikes* que o vídeo recebeu; e `views` (vetor de 10 inteiros), que representa a quantidade de visualizações que o vídeo recebeu nos últimos 10 dias. No caso desse último campo, `views[0]` representa a quantidade de visualizações no dia corrente, `views[1]` a quantidade de visualizações no dia anterior, etc.

2. (5 points) Escreva uma função de nome `preencheVideo` que recebe um parâmetro do tipo `Video` como parâmetro **por referência** e um inteiro `id` representando o seu identificador e preenche os seus campos. O número de *likes*, *dislikes* deve ser um número aleatório gerado pela função `rand()`. O vetor `views` deve ser preenchido da seguinte maneira. Sendo i o índice do vetor, o número de visualizações do índice i é dado pela parte inteira do resultado de $rand() \times e^{-i}$.

3. (5 points) Escreva uma função que recebe um `Video` como parâmetro e retorna 1 se o vídeo está *trending* (viralizou) e 0 caso contrário. Um vídeo é considerado *trending* se o número de visualizações **sempre** cresceu nos últimos 10 dias, ou seja, se o número de visualizações do vídeo no dia corrente é maior que no dia anterior, que é maior que há dois dias, que é maior que há três dias, e assim por diante. A sua função deve ter o seguinte protótipo:

```
int ehTrending(Video v) ;
```

4. (4 points) Escreva um procedimento de nome `troca` que recebe um vetor de vídeos e dois índices inteiros `i` e `j` e troque os vídeos de índice `i` e `j` de posição no vetor. O seu procedimento deve ter o seguinte protótipo:

```
void troca(Video v[], int i, int j) ;
```

5. (7 points) Escreva um procedimento para identificar vídeos *ruins* em um vetor de vídeos e os mover para o fim do vetor. Um vídeo ruim é um vídeo em que a razão *likes/dislikes* é menor que 1. O seu procedimento recebe como parâmetro o vetor de vídeos e o seu tamanho *n* por referência. Ao final, você deve atualizar o valor de *n* para que ele considere apenas os vídeos que não são *ruins*. O seu procedimento deve ter o seguinte protótipo:

```
void afastaRuins(Video v[], int *n) ;
```

6. (6 points)

Complete o programa abaixo. Primeiro, este programa deve criar e preencher aleatoriamente um vetor de 30 vídeos. Depois, deve imprimir quais desses vídeos são *ruins*. Depois, dos que não são ruins, deve imprimir quais são *trending*.

```
#define NUM_VIDEOS 30
```

```
void imprimeVideo(Video v) {
    int i;
    printf("\n%d %d %d views: ", v.id, v.likes, v.dislikes);
    for(i=0; i<10; i++)
        printf(" %d", v.views[i]);
}
```

```
void main() {
    Video videos[NUM_VIDEOS];
    int i;
    //preenche videos aleatoriamente

    for(_____) {
        _____;
    }
    int n = NUM_VIDEOS;

    afastaRuins(_____);
    printf("\nvideos ruins:");

    for(_____)
        imprimeVideo(videos[i]);
    printf("\nvideos trending:");

    for(_____) {
        if(_____)
            imprimeVideo(videos[i]);
    }
}
```