

Lista 6 de Exercícios

Exercícios com alocação dinâmica

1. Escreva uma função em C que receba como parâmetros um vetor (vet) de n números reais e um ponteiro para inteiro (tam) e retorne o ponteiro para um novo vetor, alocado dinamicamente, contendo somente os elementos do vetor original que são maiores que a média de todos os valores. Por exemplo, o vetor {5.5, 3.2, 2.5, 9.4, 7.5, 4.1, 5.4, 8.0, 6.9}, daria origem ao vetor {9.4, 7.5, 8.0, 6.9}, de tamanho 4. No ponteiro para inteiro deverá ser armazenado o tamanho do novo vetor. Se não for possível alocar o novo vetor, a função deve retornar NULL.

float *acima_da_media(int n, float *vet, int *tam);

- 1) Calcular a média de vet
- 2) verificar quantos elementos de vet são maiores do que a média
- 3) alocar dinamicamente um vetresposta conforme o tamanho verificado no item 2
- 4) preencher o vetresposta com os elementos do item 2
- 5) atualizar a variável ponteiro tam com o tamanho do novo vetor

2. Escreva uma função em C que receba como parâmetro uma cadeia de caracteres, contendo apenas letras entre 'A' e 'Z' ou 'a' e 'z' e espaços em branco, e retorne uma nova cadeia -- alocada dinamicamente --, contendo uma cópia da cadeia original, mas sem espaços em branco no início ou no final, isto é, sem qualquer caractere ' ' (espaço) localizado antes da primeira letra ou depois da última letra da cadeia. Por simplificação, considere que a cadeia de entrada possui no mínimo uma letra (entre 'A' e 'Z' ou 'a' e 'z'). Por exemplo, se a cadeia “ nota do aluno ” (com três espaços em branco no início e dois no final) for passada para a função, a cadeia retornada deve ser “nota do aluno”, sem nenhum espaço no início ou no final. Se não for possível alocar dinamicamente a nova cadeia, a função deve retornar NULL. O protótipo da função é:

char *compactar(char *s);

3. O cadastro de uma turma de n alunos é representado por um vetor de inteiros, contendo a matrícula de cada aluno, e um vetor de reais, contendo a nota final de cada aluno, de tal modo que para uma mesma posição do vetor (mesmo índice), a matrícula e a nota se referem a um mesmo estudante. Considere que um aluno está aprovado se tem nota final maior ou igual a 5,0. Escreva uma função que recebe como parâmetros o inteiro n, o ponteiro mat para o vetor de matrículas, o ponteiro notas para o vetor de notas e um ponteiro para a variável inteira tam. A função deve contar o número de alunos aprovados, criar um novo vetor de inteiros alocado dinamicamente com o

tamanho exato para conter apenas a matrícula desses alunos, e armazenar no novo vetor as matrículas correspondentes. Finalmente, a função deverá armazenar o tamanho do novo vetor na variável tam e retornar o ponteiro para seu primeiro elemento. O protótipo da função é:

```
int *aprovados(int n, int *mat, float *notas, int *tam);
```

4. Todo ano um concurso de programação premia os participantes que obtêm a maior média ponderada em uma bateria de dois testes. Escreva uma função que recebe sete parâmetros: o inteiro n indicando a quantidade de participantes do concurso, o ponteiro inscr para o vetor de inteiros que contém as inscrições desses participantes, o ponteiro t1 para o vetor de reais que contém a nota de cada participante no primeiro teste, o inteiro p1 que indica o peso dessa nota na média ponderada, o ponteiro t2 para o vetor de reais que contém a nota de cada participante no segundo teste, o inteiro p2 que indica o peso dessa nota na média ponderada, e um ponteiro para a variável inteira tam. A função deve:

- calcular a média ponderada de cada participante;
- criar um novo vetor de inteiros alocado dinamicamente com o tamanho exato para conter apenas as inscrições dos participantes que obtiveram a maior média (pode haver empate);
- armazenar no novo vetor as inscrições correspondentes (em qualquer ordem);
- armazenar o tamanho do novo vetor na variável tam; e
- retornar o ponteiro para o novo vetor.

Considere que para uma mesma posição do vetor (mesmo índice), a inscrição e as notas se referem a um mesmo participante. O protótipo da função é:

```
int *premiados(int n, int *inscr, float *t1, int p1, float *t2, int p2, int *tam);
```

5. Júlio César, o imperador romano, usava a Cifra de César para enviar ordens secretas aos seus generais. Este código consiste em trocar cada letra de uma mensagem pela terceira letra seguinte. Ou seja, a letra 'a' é substituída por 'd', 'b' é substituída por 'e', 'c' é substituída por 'f', e assim por diante. As exceções são as três últimas letras do alfabeto, 'x', 'y' e 'z', que são substituídas, respectivamente, por 'a', 'b' e 'c'. Por exemplo, a palavra "zebra" se transformaria em "cheud". Escreva uma função que codifique uma mensagem de acordo com a Cifra de César, aplicando a codificação apenas às letras da mensagem, maiúsculas ou minúsculas, e preservando os demais caracteres. Essa função deve receber como parâmetro uma cadeia de caracteres com a mensagem a ser codificada e retornar o ponteiro para uma nova cadeia de caracteres, alocada dinamicamente, com a mensagem codificada. A cadeia de caracteres original não pode ser alterada. A função deve ter o seguinte protótipo:

```
char* cifra_cesar(char* msg);
```

6. Faça um programa que informe quais são os números que se repetem em um vetor. O programa deve seguir os seguintes passos:

a) Leia um inteiro do usuário e aloque um vetor de inteiros dinamicamente com este tamanho. Depois, preencha todo o vetor de inteiros com valores entre 0 e 9.

b) Imprima na tela os valores que se repetem no vetor.

- Exemplo de execução do programa:

> Entradas:

```
tam = 9  
vet = |1|5|8|4|4|5|0|8|8|
```

> Saída:

```
5, 8, 4
```

- Obs.: Faça tudo na main.

7. Modifique o programa anterior da seguinte forma:

a) Passo igual ao da questão 1.

b) Ao invés de calcular e imprimir na main as repetições, crie uma função 'repeticao' para fazer esse trabalho. A função deve seguir o protótipo

```
void repeticao(int *vet, int tam);
```

Onde 'vet' e 'tam' foram lidos no passo a).

- Exemplo de execução: idêntico ao anterior.

- Obs.: A impressão é feita dentro da função 'repeticao'.

8. Modifique novamente o seu programa da seguinte forma:

a) Passo igual aos das questões 1 e 2.

b) A sua função `repeticao` não mais imprimirá o resultado, mas sim a `main`. Para isso, será necessário que a sua função `'repeticao'` siga o seguinte protótipo:

```
int *repeticao(int *vet, int tam, int *n);
```

A sua função `'repeticao'` agora alocará um novo vetor de inteiros dinamicamente que deverá ser preenchido com os valores que se repetem. O tamanho desse vetor deve ser gravado na variável `n`, que referencia uma variável na `main` (crie uma variável na `main` para isso). A função `'repeticao'` retornará para a `main` este novo vetor resposta.

c) Imprima na `main` o vetor retornado pela função `'repeticao'`.

- Exemplo de execução do programa:

> Entradas:

```
tam = 9  
vet = |1|5|8|4|4|5|0|8|8|
```

> Valores contidos nas variáveis após a execução da função `'repeticao'` (não devem ser exibidos):

```
n = 3  
resposta = |5|8|4|
```

> Saída:

```
5, 8, 4
```