

Lista de Exercícios 1

1) Considerando a existência de notas (cédulas) nos valores R\$ 100, R\$ 50, R\$ 20, R\$ 10, R\$ 5, R\$ 2 e R\$ 1, escreva um programa que capture um valor inteiro em reais (R\$) e determine o menor número de notas para se obter o montante fornecido. A função deve ter o seguinte protótipo:

```
int total_de_notas(int valor);
```

2) Considere a função que calcula o n -ésimo número harmônico $H(n)$ conforme definida a seguir:

$$H(n) = 1 + 1/2 + 1/3 + 1/4 + \dots + 1/n \quad (n \geq 1)$$

Apresente uma definição recursiva para $H(n)$ e implemente o cálculo desse valor como uma função RECURSIVA em C. A função deve ter o seguinte protótipo:

```
double harmonico(int n);
```

3) Escreva uma função RECURSIVA que recebe como parâmetro um valor inteiro e imprime a representação binária deste número. A função deve ter o seguinte protótipo:

```
void imprime_binario(int n);
```

4) O MDC de dois números inteiros é o maior número inteiro que divide ambos sem deixar resto. Este valor pode ser determinado da seguinte forma: calcula-se o resto da divisão do maior número pelo menor. Se o resto for igual a zero o MDC é o menor dos números. Se não for, o MDC será o mesmo que o MDC do menor número e o resto. Escreva uma função RECURSIVA para o cálculo do MDC. A função deve ter o seguinte protótipo:

```
int mdc(int a, int b);
```

5) Em uma progressão aritmética (PA) de razão k , considerando quaisquer elementos consecutivos x_i e x_{i+1} , temos que $x_{i+1} = x_i + k$. Escreva uma função RECURSIVA que recebe como parâmetros o inteiro x , correspondendo ao primeiro elemento da PA, e o inteiro k , representando a razão da PA, e retorna a soma dos k primeiros elementos da PA. O protótipo da função é:

```
int soma_pa(int x, int k);
```

6) Implemente uma função que recebe como parâmetro um vetor de números reais vet de tamanho n e retorna o índice do maior valor armazenado nesse vetor. Considere que não há valores repetidos nesse vetor. Essa função deve obedecer ao protótipo:

```
int max_vet (int n, float *vet);
```

7) Implemente uma função que receba como parâmetros um vetor de números inteiros v e seu comprimento n , e verifique se os elementos do vetor correspondem aos termos de uma PA (progressão aritmética), isto é, se existe um número inteiro k tal que v_i seja igual a $v_0 + i * k$. Caso os elementos do vetor correspondam aos termos de uma PA, a função deve retornar o valor calculado para k . Por exemplo, se for passado para essa função o vetor $\{2,10,18,26,34\}$, ela deve retornar o valor 8. Caso o vetor não corresponda a uma PA, a função deve retornar 0. Assuma que o vetor sempre terá pelo menos três elementos. O cabeçalho da função é dado por:

```
int teste_PA(int n, int *v);
```

8) Escreva uma função em C que determine a posição em um vetor do elemento que corresponde ao “menor valor único”. Essa função deve receber como parâmetros um vetor de números reais (double) e o inteiro n que indica o tamanho do vetor, e deve retornar o índice do elemento que tem o menor valor que não se repete no vetor.

```
int menor_valor_unico(float *valores, int n);
```

9) Escreva uma função em C que identifique se um vetor é um subconjunto ordenado de outro vetor, isto é, se todos os elementos deste vetor são também elementos do outro vetor e aparecem na mesma ordem relativa em ambos. Por exemplo, o vetor $\{34, 12, 7, 41, 52, 45\}$ é subconjunto ordenado do vetor $\{9, 34, 12, 7, 41, 52, 9, 27, 45, 13\}$. Essa função deve receber como parâmetros o vetor sub e seu tamanho n , e o vetor vet e seu tamanho m e retornar 1, se sub for subconjunto ordenado de vet , ou 0 caso contrário.

```
int subconjunto_ordenado(int *sub, int n, int *vet, int m);
```

10) Escreva uma função que verifica se um vetor de números inteiros está em ordem crescente, decrescente ou fora de ordem. A função recebe como parâmetros o vetor vet e o inteiro n , que representa o tamanho do vetor, e deve retornar 1, se o vetor está em ordem crescente, -1, se está em ordem decrescente, ou 0, se não está em ordem. Considere que o vetor não tem valores repetidos. O protótipo da função é:

```
int ordenado(int *vet, int n);
```

11) Implemente uma função que calcula as raízes reais de uma equação do segundo grau, do tipo $ax^2 + bx + c = 0$. A função deve retornar o número de raízes reais da equação, obedecendo ao seguinte protótipo:

```
void raizes(float a, float b, float c, float * x1, float * x2);
```

Onde os inteiros a , b e c representam os coeficientes da equação, e $x1$ e $x2$ são ponteiros para as variáveis onde devem ser guardadas as raízes reais da equação. Quando houver apenas uma raiz, $x1$ deve guardar esse valor. Quando houver duas raízes, $x1$ deve guardar

a a raiz de menor valor e x_2 a de maior valor. Para o cálculo da raiz quadrada, utilize a função `sqrt` definida na biblioteca `math.h` do C, que tem o protótipo `double sqrt (double n)`.

12) Escreva uma função em C que determina a média e a situação de um aluno em uma disciplina. A função recebe como parâmetros as três notas de um aluno (p_1 , p_2 , e p_3), seu número de faltas (`faltas`), o número total de aulas da disciplina (`aulas`) e o ponteiro para uma variável (`media`), conforme o seguinte protótipo:

```
char situacao(float p1, float p2, float p3, int faltas,
             int aulas, float *media);
```

Na variável indicada pelo ponteiro `media`, a função deve armazenar a média do aluno, calculada como a média aritmética das três provas. Além disso, a função deve retornar um caractere indicando a situação do aluno no curso, de acordo com o critério definido na tabela abaixo:

Número de Faltas	Média	Situação	Retorno
Menor ou igual a 25% do total de aulas	Maior ou igual a 6,0	Aprovado	A
	Menor que 6,0	Reprovado	R
Maior que 25% do total de aulas	Qualquer	Reprovado por faltas	F

13) Desde março deste ano, os 33 táxis da cidade de Rio de Janeiro tiveram que instalar a aferir taxímetros para o cálculo do valor das corridas. De acordo com a tabela em vigor, a tarifa inicial (ou bandeirada) custa R\$4,95. Além disso, para cada quilômetro percorrido são cobrados mais R\$2,50, na bandeira 1, ou R\$3,00, na bandeira 2. Escreva uma função em C que calcula os valores da corrida de táxi em Rio das Ostras. A função recebe como parâmetros o valor real `dist`, correspondendo à distância percorrida pelo táxi (em quilômetros), e os ponteiros `b1` e `b2`, indicando os endereços onde devem ser armazenados, respectivamente, os valores calculados para a corrida na bandeira 1 e na bandeira 2.

```
void calcula_corrida(float dist, float *b1, float *b2);
```

14) No estudo da física, o deslocamento instantâneo s e a velocidade instantânea v de corpo podem ser calculados a partir do deslocamento inicial s_0 , da velocidade inicial v_0 , da aceleração a e do intervalo de tempo t utilizando-se as seguintes fórmulas:

$$s = s_0 + v_0 * t + \frac{(a * t^2)}{2} \quad \text{e} \quad v = v_0 + a * t$$

Escreva uma função em C para o cálculo do movimento de um corpo que recebe como parâmetros os ponteiro para as variáveis reais s (que já contém o valor do deslocamento inicial) e v (que já contém o valor da velocidade inicial), e os valores reais a , representando a aceleração do corpo, e t , representando o intervalo de tempo. Essa função deve atualizar as variáveis apontadas por s e v , respectivamente, com o valor do deslocamento instantâneo e o valor da velocidade instantânea calculados ao final do intervalo de tempo fornecido em t , de acordo com as fórmulas apresentadas.

```
void movimento(float *s, float *v, float a, float t);
```

15) A média aritmética, A , a média harmônica, H , e a média desarmônica D , de um conjunto de n valores são dadas pelas seguintes fórmulas:

$$A_n = \frac{1}{n} \times \sum_{i=1}^n v_i, \quad H_n = \frac{n}{\sum_{i=1}^n \left(\frac{1}{v_i}\right)} \quad \text{e} \quad D_n = H\left(A_n, \frac{H_n^2}{A_n}\right)$$

Implemente uma função que calcule a média aritmética e a média harmônica de um conjunto de valores armazenados em um vetor. Esta função deve obedecer ao protótipo abaixo, recebendo como parâmetros o número de elementos do vetor, o endereço inicial do vetor e os endereços das variáveis onde os valores das médias, isto é, os valores de A , H e D , devem ser armazenados.

```
void medias(int n, float* v, float* pA, float* pH, float* pD);
```

16) Utilizando-se a série de Taylor, o valor da função $\text{sen}(x)$ pode ser aproximado por uma função polinomial definida pela seguinte soma infinita de n termos.

$$\text{sen } x = \sum_{n=0}^{\infty} \frac{-1^n}{2n+1!} x^{2n+1}$$

Ou seja, tomando os 4 primeiros termos da série (para n variando de 0 até 3), um valor aproximado da função $\text{sen}(x)$ poderia ser calculado pela expressão:

$$\text{sen } x \approx x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!}$$

Por exemplo, calculando o valor da expressão, podemos obter o valor estimado para o seno de um ângulo de 60° ($x = 0,5236$ radianos) como:

$$\text{sen } 0,5236 \approx 0,5236 - \frac{0,5236^3}{3!} + \frac{0,5236^5}{5!} - \frac{0,5236^7}{7!} \approx 0,49999999$$

Escreva uma função em C que calcule o valor dos n primeiros termos da série de Taylor que aproxima o valor da função *seno* para um dado valor de x . Esta função recebe como parâmetros o inteiro n , correspondente ao número de termos, e um número real correspondente ao valor de x (em radianos), e deve alocar dinamicamente um vetor de tamanho n para, em cada elemento i , armazenar o valor referente ao i -ésimo termo da série. A função deve retornar o ponteiro para o novo vetor, conforme o protótipo:

```
float* taylor_sen(int n, float x);
```

17) 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);
```

18) 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);
```

19) Considere que você armazenou os valores observados em um experimento de física em um vetor de números reais (do tipo float). Para melhorar a precisão dos seus dados, você pode eliminar o maior e o menor valor dentre todas as medições. Por exemplo, para o vetor (1.2, 1.2, 0.9, 1.5, 1.4, 1.5), o terceiro elemento deveria ser retirado por ser o de menor valor, e também o quarto ou o sexto elementos (a escolha é indiferente) por serem o de maior valor. Escreva uma função em C que receba como parâmetros um vetor de n números reais, correspondendo às medições de um experimento, e retorne um novo vetor (alocado

dinamicamente) sem os elementos do vetor original que têm o maior e o menor valores (admita que n é sempre maior que 3).

```
float* vet2(int n, float *vet1);
```

20) 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:

- (a) calcular a média ponderada de cada participante;
- (b) 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);
- (c) armazenar no novo vetor as inscrições correspondentes (em qualquer ordem);
- (d) armazenar o tamanho do novo vetor na variável tam ; e
- (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);
```