

Estruturas de Dados

Exercícios – Capítulos 3 e 4

PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO



Questão 1

Considerando a definição da série harmônica abaixo:

$$H_n = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \dots + \frac{1}{n}$$

- (a) Escreva uma função que calcule o valor de H_n dado o número de termos da série desejado. Essa função recebe como parâmetro o número inteiro positivo n , e retorna o H_n correspondente, seguindo o protótipo:

```
float harmonico (int n);
```

- (b) Escreva um programa C completo que capture o número de termos da série harmônica, fornecido pelo usuário via teclado, e imprima o H_n correspondente. Este programa deve fazer uso da função do item anterior.

Questão 1

```
/* programa para calcular a série Harmônica */
#include <stdio.h>

/* Função auxiliar */
float harmonico (int n)
{
    float s = 0.0;
    int i;
    for (i=1; i <=n; i++)
        {s = s + 1.0/i;}
    return s;
}
```

Questão 1

```
/* Função principal */
int main (void)
{
    int n;
    float t;
    /* mostra mensagem para o usuário */
    printf("Digite o número do termo desejado: ");
    /* captura valor entrado via teclado */
    scanf("%d",&n);
    /* chama a função */
    t = harmonico(n);
    /* exibe resultado */
    printf("Valor do n-ésimo termo harmonico: %f\n", t);
    return 0;
}
```

Questão 1

- Erro de precisão:

```
/* programa para calcular a série Harmônica */
#include <stdio.h>
/* Função auxiliar */
float harmonico (int n)
{ float s = 0.0;
  int i;
  for (i=1; i <=n; i++)
    {s = s + 1/i;} /* a expressão em vermelho retorna sempre 0 */
  return s;      /* para i>1 pois será realizada sobre inteiros */
}
```

Questão 2

a) Implemente uma função que retorne uma aproximação do valor de π , de acordo com a Fórmula de Leibniz:

$$\pi \approx 4 * \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \dots \right)$$

Ou seja:

$$\pi \approx 4 * \sum_{i=0}^{n-1} \frac{(-1)^i}{2 * i + 1}$$

Esta função deve obedecer ao protótipo abaixo, onde n indica o número de termos da série que devem ser usados para o cálculo do valor de π retornado.

```
float pi (int n);
```

b) Implemente um programa completo que capture via teclado o número de termos a ser utilizado na avaliação do valor π e, usando a função do item acima, calcule e imprima o valor de π calculado. Se o número de termos fornecido for menor do que um, o programa deve exibir uma mensagem de erro.

Questão 2

```
/* Programa para cálculo de PI pela fórmula de Leibnitz */
#include <stdio.h>
/* Função auxiliar */
float pi (int n)
{ float s = 0.0;    /* somatorio */
  int num = -1;    /* numerador do somatorio */
  int i;          /* contador de termos */
  for (i=0; i<n; i++)
  {
    num = (-1)*num;
    s = s + (float)num/(float)(2*i+1);
  }
  return 4*s;
}
```

Questão 2

```
/* Função principal */
int main (void)
{
    int n;
    float t;
    /* mostra mensagem para o usuário */
    printf("Digite o número do termo desejado: ");
    /* captura valor entrado via teclado */
    scanf("%d",&n);
    if (n<1)
        {printf("Erro: numero de termos menor do que 1");
        return 1;};
    /* chama a função e exibe resultado */
    printf("Valor de pi com n termos: %f\n", pi(n));
    return 0;
}
```

Questão 3

Considere uma função `movimento` para calcular a posição (s) e velocidade (v) de uma partícula em um determinado instante t , dado sua aceleração a , posição s_0 e velocidade v_0 , de acordo com as fórmulas:

$$s = s_0 + v_0 t + \frac{at^2}{2}$$

$$v = v_0 + at$$

O programa abaixo ilustra o uso dessa função, imprimindo os valores 55.5 e 25.0, que representam a posição e velocidade calculadas para os valores utilizados pelo programa.

```
int main (void) {
    float s0 = 3.0, v0 = 10.0, a = 5.0, t = 3.0;
    float s, v;
    movimento(s0, v0, a, t, &s, &v);
    printf("%f %f\n", s, v);
    return 0;
}
```

Implemente a função `movimento` de tal forma que o exemplo acima funcione adequadamente. A solução deve funcionar para quaisquer valores de t , a , s_0 e v_0 , não podendo ser particularizada para o programa acima.

Questão 3

```
/* Programa para calcular a posição e velocidade */
#include <stdio.h>
/* Função movimento */
void movimento (float s0, float v0, float a, float t, float* s, float* v)
{
    *s = s0 + v0 * t + (a * t * t)/2.0;
    *v = v0 + a * t;
}
/* Função principal */
int main (void) {
    float s0 = 3.0, v0 = 10.0, a = 5.0, t = 3.0;
    float s, v;
    movimento(s0, v0, a, t, &s, &v);
    printf("%f %f\n", s, v);
    return 0;
}
```

Questão 3

- Uso incorreto de apontador:

```
/* Função movimento */  
void movimento (float s0, float v0, float a, float t, float* s, float* v)  
{  
    &s = s0 + v0 * t + (a * t * t)/2.0;  
    &v = v0 + a * t;  
}
```

Questão 4

Considere uma disciplina que adota o seguinte critério de aprovação: os alunos fazem duas provas (P1 e P2) iniciais, obtendo média dada por: $M = (P1+P2)/2$. Se a média M for maior ou igual a 5.0, e se nenhuma das duas notas for inferior a 3.0, o aluno passa direto (com média final, MF, igual a M). Caso contrário, o aluno faz uma terceira prova (P3) e a média final é calculada considerando-se essa terceira nota.

- Se $P3 \geq 3.0$: $MF = (P3 + \max(P1,P2)) / 2$
- Se $P3 < 3.0$: $MF = (M + P3) / 2$

O aluno é considerado aprovado se sua média final for maior ou igual a 5.0.

Escreva um programa completo que leia inicialmente as duas notas de um aluno, fornecidas pelo usuário via o teclado. Se as notas não forem suficientes para o aluno passar direto, o programa deve capturar a nota da terceira prova, também fornecida via o teclado. Como saída, o programa deve imprimir a média final do aluno, seguida da mensagem “Aprovado” ou “Reprovado”, conforme o critério descrito acima. Note que se as duas primeiras notas forem suficientes para o aluno passar, o programa não deve ler a terceira.

```

/* Programa para calcular a média final */
#include <stdio.h>
int main ( )
{
    float P1,P2,P3,M,MF;
    printf("Entre P1 e P2: ");
    scanf(" %f %f ",&P1,&P2);
    M=(P1+P2)/2;
    if (P1>=3 && P2>=3 && M>=5.0)
        { MF=M; }
    else {
        printf("Entre P3: ");
        scanf(" %f ",&P3);
        if (P3>=3)
            { float P=(P1>P2)?P1:P2;
              MF=(P3+P)/2;
            }
        else { MF=(M+P3)/2; }
    }
    if (MF>=5)
        printf("%.1f Aprovado\n",MF);
    else printf("%.1f Reprovado\n",MF);
    return 0;
}

```