

Estruturas de Dados e Algoritmos - Primeira Prova 2018.2 - Eng. de Computação

Nome: _____

Data: 05/10/2018

obs: todos os códigos devem ser escritos em C.

Questão 1.) (2,0) Informe a saída fornecida pelo código e mostre o que ficará armazenado nas variáveis. Use a representação conceitual de ponteiro, com uma seta para indicar para qual região de memória o ponteiro aponta.

```

1  #include <stdio.h>
2
3  int main() {
4      char nome[20] = "IFPB-CG";
5      int dias[5] = {5,3,1,6,7};
6
7      char *p = nome+5;
8      nome[4] = '\0';
9
10     for(int i = 0; nome[i] != '\0'; i++) {
11         printf("%c", nome[i]);
12     }
13     printf("\n");
14     for(int i = 0; p[i] != '\0'; i++) {
15         printf("%c", p[i]);
16     }
17     printf("\n");
18
19     int *q;
20     for(int i = 0; i < 5; i++) {
21         if(dias[i] == 1) {
22             q = &dias[i];
23         }
24     }
25
26     printf("%d\n", q[2]);
27     return 0;
28 }
```

Questão 2.) (2,5) A função denominada *media_depois* tem como objetivo aplicar um tipo de filtro da média aos elementos de um vetor de valores reais. Esse filtro consiste em atualizar cada elemento do vetor, como sendo a média aritmética entre o valor atual do elemento e o valor do próximo elemento, caso exista próximo elemento. Por exemplo, se o vetor possuir inicialmente os valores {1.0, 0.0, 3.0, 5.0}, ele terá os valores {0.5, 1.5, 4.0, 5.0} após a aplicação do filtro.

De modo a verificar se o valor médio dos valores do array muda com a aplicação do filtro, a função deve retornar o valor médio dos valores antes da aplicação do filtro (usando o **return**) e deve colocar o valor médio dos elementos do array após a aplicação do filtro em uma variável que foi passada como parâmetro por referência.

Deseja-se que a função funcione para arrays de qualquer tamanho. O programa a seguir está escrito de forma correta para o objetivo proposto? Em caso negativo, aponte as linhas de código que possuem erros e sugira correções a serem aplicadas ao código para que ele se torne correto.

```

1  #include <stdio.h>
2
3  double media_frente(double vet[], int *m) {
4
5      double ma = 0, md = 0;
6
7      for(int i = 0; i < 30; i++) {
8          ma += vet[i];
9          vet[i] = (vet[i] + vet[i+1])/2.0;
10         md += vet[i];
11     }
12     ma = ma/30.0;
13     m = md/30.0;
14     return ma;
15 }
16
17 int main() {
18     double v[20];
19     double mediad, mediaa;
20
21     for(int i = 0; i < 20; i++) {
22         scanf("%lf",&v[i]);
23     }
24
25     mediaa = media_frente(v, mediad);
26     printf("Media antes: %lf\n",mediaa);
27     printf("Media depois: %lf\n",mediad);
28     return 0;
29 }

```

Questão 3.) (2,5) Crie uma função **RECURSIVA** que calcule o valor do número irracional π (a função deve-se chamar *getPi(int n)*) e deve receber como parâmetro o último termo (n) a ser considerado no cálculo do π ($0 \leq n \leq 1000$). Para implementar a função que calcula o valor de π deve-se utilizar a fórmula de Leibniz para π , que considera os termos de 0 a n , mostrada a seguir:

$$\pi(n) = \frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} \dots + (-1)^n \frac{4}{2n+1}$$

Note que esse somatório teoricamente é infinito, pois em teoria o n poderia ser infinito, mas na sua função você deve considerar apenas os termos de 0 a n , sendo que n é um valor finito passado como parâmetro para a função. Lembre-se que a resposta é um valor real (**double**).

Para calcular a potenciação, pode-se utilizar a função *pow()* de *math.h* (ex: *pow(a,b)* calcula a^b).

Após escrever a função para obter o valor de π , escreva na função *main()* um programa que lê da entrada um valor inteiro n e coloque na saída o valor de π calculado usando a função *getPi*, considerando os termos de 0 a n .

Questão 4.) (3,0) Você foi contratado para desenvolver um revolucionário esquema de criptografia de mensagens, que utiliza frases aparentemente comuns para enviar mensagens escondidas. Esse método consiste em combinar as últimas letras de um conjunto de palavras, para obter a palavra desejada. Por exemplo, considere a seguinte frase:

“E ED vai mais longe eita saudades py”

Ao combinar as últimas letras de cada palavra se obtém a mensagem:

“EDiseasy”

O seu programa deve ler da entrada uma string de no máximo 100 caracteres, contendo um conjunto de palavras separadas por espaços em branco. É garantido que a string dada como entrada contém ao menos uma palavra e é formada apenas por letras e espaços em branco. Deve-se colocar na saída a mensagem criptografada que foi obtida por meio do método descrito.

Atente para os seguintes detalhes:

- Uma palavra é uma sequência de letras;
- As palavras sempre são separadas por pelo menos um espaço em branco;
- Pode haver espaços em branco no início da string, mas é garantido que após a última palavra não haverá mais espaços em branco.

obs: use `%[^\n]s` para ler a string com espaço em branco. Deve-se resolver esse problema sem utilizar qualquer função pronta para manipulação de string.

Exemplo de entrada 1

Vou fechar as prova geral

Exemplo de saída 1

ursal

Exemplo de entrada 2

Ainda bem que fechei

Exemplo de saída 2

amei