

Nome: _____

Prova individual. Entregar os programas e respostas ao professor (a cópia e conferência dos programas copiados são responsabilidade do aluno). É permitida a consulta à apostila do professor, impressa ou em PDF. Qualquer acesso à rede ou qualquer programa aberto que não seja o ambiente de desenvolvimento invalidará a prova do aluno. No editor de programas em uso, apenas os programas referentes às soluções da prova poderão estar abertos. Não consulte programas prontos.

Questão 1:

[2 pontos]

Complete o programa 1 abaixo conforme as instruções.

Programa 1: Programa incompleto

```
void calculaMedia( PARÂMETROS ) {  
    CÓDIGO  
}  
  
int main() {  
  
    Alocar memória para 3 variáveis reais  
  
    Receber o valor de duas delas , via teclado  
  
    Usar a função acima para calcular a média das duas e  
    armazenar esta média na terceira variável  
  
    Imprimir o valor da média  
  
    Liberar memória  
  
}
```

(ponteiros-001)

Questão 2:

[3 pontos]

Faça um programa que tenha uma função que retorna **void** e que preenche um vetor com **20** números inteiros; outra função que também retorna **void** e que dobra os valores armazenados neste vetor; e uma última função que imprime este vetor. A função **main** deve alocar memória para este vetor, chamar a função que preenche o vetor, chamar a função que imprime o vetor, chamar a função que dobra os elementos do vetor, chamar novamente a função que imprime o vetor, liberar a memória alocada. Não use colchetes para indexar o vetor.

(vetorPassadoPorReferencia-001)

Questão 3:

[5 pontos]

Faça um programa que calcula o valor de a^n (não use funções da biblioteca **math.h**). Qual a complexidade assintótica $O(??)$ do programa? Insira um contador de instruções que são função de **n** para comprovar a complexidade deduzida analiticamente.

Agora, considere o programa a seguir:

Programa 2: Programa

```
#include <stdio.h>  
  
int main () {  
    long a, n;  
    printf("\na=");  
    scanf("%ld", &a);  
    printf("\nn=");  
    scanf("%ld", &n);  
    long z=1, t=a, k=n;  
    while ( k>0 ) {  
        if ( k%2 != 0 ) z*=t;  
        k = k/2;  
        if ( k != 0 ) t*=t;  
    }  
    printf("\nz=%ld\n", z);  
}
```

O que ele faz? Qual a complexidade assintótica $O(??)$? Insira um contador de instruções que são função de **n** para comprovar a complexidade deduzida analiticamente.

(complexidade-001)