

**VETOR, MATRIZ, STRING, PONTEIRO E FUNÇÕES**

**1ª Questão**

Explique o que acontece na execução de cada instrução que estão seguidas de comentários (//):

```
int *a, *b, c = 10, d = 4;  
a = &c; //a recebe o endereço da variável c.  
b = &d; //b recebe o endereço da variável d.  
*b = 8; //o endereço apontado por b recebe o valor 8 //d = 8  
*a = *b; //o endereço apontado por a recebe o conteúdo do endereço apontado por b // c = d  
*a = 1; //o endereço apontado por a recebe o valor 1  
b = a; //b aponta para o endereço contido em a  
*b = 0; //o endereço apontado por b recebe o valor 0 //c = 0
```

**2ª Questão**

Faça o que se pede:

a) Complete o programa abaixo mostrando os valores dos elementos do vetor usando as seguintes forma de acesso:

1. Por meio de indexadores ar[i];
2. Por meio de ponteiros \*(p+i);

b) Altere os valores do vetor acrescentando 10 unidades em cada elemento usando o acesso por meio de ponteiros \*(p+i);

c) Mostre os endereços de cada elemento do vetor usando o especificador %p;

```
#include <stdio.h>  
#include <stdlib.h>  
int main(int argc, char *argv[])  
{  
    int ar[10]={20,30,40,50,60};  
    int *p;  
    p=&ar[0];  
    int i;  
    for(i= 0; i<5; i++)  
        printf("%d, ", ar[i]);  
    printf("\n\n");  
  
    for(i= 0; i<5; i++)  
        printf("%d, ", *(p+i));  
    printf("\n\n");  
  
    for(i= 0; i<5; i++)  
    {  
        *(p+i) += 10 ;  
        printf("%p, ", p+i);  
    }  
    printf("\n\n");  
    system("PAUSE");  
    return 0;  
}
```

**3ª Questão**

Dada a função abaixo, identifique:

- a) O tipo que a função retorna; //Retorna o tipo float
- b) Os parâmetros da função; //Possui 2 parâmetros do tipo float

```
#include <stdio.h>
#include <stdlib.h>
```

```
float CalcularMedia(float x, float y);
```

```
int main(int argc, char *argv[]) {
float a=5,b=15,resultado;
resultado=CalcularMedia(a,b);
printf("Media=%6.2f\n",resultado);
system("PAUSE");
return 0;
}
```

```
float CalcularMedia(float a, float b) {
float media;
media =(a+b)/2;
return(media);
}
```

#### 4ª Questão

Observe o programa abaixo, e responda:

- a) Onde ocorre nas duas funções a chamada de parâmetros por valor e por referência ?

Na função Funcao\_Swap recebe parâmetros por referência e na função Multiplica recebe parâmetros por valor.

- b) O que caracteriza cada uma das formas de chamada ?

Na chamada de parâmetros por valor uma copia dos valores dos parâmetros é passada para a função e a alteração desses valores não altera o valor da variável passada como parâmetro. Na chamada de parâmetros por referência o endereço da variável é passado para a função e a alteração do valor desse parâmetro implica na alteração do valor da variável passada como parâmetro.

- c) A partir do conceito de localidade e escopo de variáveis que tipo de chamada seria usada para que uma função conseguisse acessar e modificar valores de variáveis com escopo diferentes ?

Chamada por referência.

```
#include <stdio.h>
#include <stdlib.h>
```

```
void Funcao_Swap(int *x, int *y);
int Multiplica(int x, int y);
```

```
int main(int argc, char *argv[])
{
int var1=20, var2=50;
Funcao_Swap(&var1, &var2);
printf("O valor de var1=%d e de var2=%d\n\n", var1, var2);
printf("O Produto var1 x var2=%d\n\n", Multiplica(var1, var2));
system("PAUSE");
return 0;
```

```
}
```

```
void Funcao_Swap(int *a, int *b) {  
    int temp;  
    temp=*a;  
    *a=*b;  
    *b=temp;  
}
```

```
int Multiplica(int a, int b) {  
    return a * b;  
}
```

### 5ª Questão

Observe a função descrita abaixo:

```
int questao(char primeiro[],char segundo, char terceiro)  
{  
    int contador, variavel=0;  
    for(contador = 0; primeiro[contador] != '\0'; contador++)  
    {  
        if(primeiro[contador] >= segundo && primeiro[contador] <= terceiro)  
        {  
            variavel++;  
        }  
    }  
    return(variavel);  
}
```

Determine qual sua utilidade.

Ela conta as letras de uma string se elas estiverem entre o intervalo determinado pelo parâmetro segundo e terceiro.

### 6ª Questão

Elaborar uma função que, recebendo como parâmetro uma string terminada por zero ('\0'), substitua todos os seus caracteres maiúsculos por minúsculos e todos os caracteres minúsculos por maiúsculos. Elaborar também um programa principal que leia uma string do teclado, chame a função descrita anteriormente e mostre o resultado obtido após a chamada.

**Sugestão:** Utilizar a seguinte regra que vale para a tabela ASCII: MAIUSCULA = MINUSCULA - ('a' - 'A') e MINUSCULA = MAIUSCULA + ('a' - 'A')

```
#include <stdio.h>
```

```
char* LowerToUpperAndUpperToLower(char* string);
```

```
int main(void)  
{  
    char enter[100];  
    scanf("%s", enter);  
    printf("%s\n", LowerToUpperAndUpperToLower(enter));  
  
    return 0;  
}
```

```

char* LowerToUpperAndUpperToLower(char* string)
{
    int i;
    for(i= 0; string[i]!='\0'; i++)
    {
        if(string[i] >= 'A' && string[i] <= 'Z')
        {
            string[i] += 'a' - 'A';
        }else if(string[i] >= 'a' && string[i] <= 'z')
            string[i] -= 'a' - 'A';
        }

    return string;
}

```

### 7ª Questão

Fazer uma função que recebe um mês e um ano como parâmetros e retorna o número de dias daquele mês daquele ano. Dica: um ano é bissexto quando é múltiplo de 4 e não múltiplo de 100, ou também quando é múltiplo de 400.

```

int diasNoMes(int mes, int ano)
{
    int diasNoMes[12] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
    int dias = diasNoMes[mes - 1];

    if( mes== 2 && (( ano%4== 0 && ano%100!= 0 ) || ano%400== 0))
        dias++;

    return dias;
}

```

### 8ª Questão

Escreva uma função CALCULA que:

- receba como parâmetros duas variáveis inteiras, X e Y;
- retorne em X a soma de X e Y;
- retorne em Y a subtração de X e Y.

Pergunta: a passagem dos parâmetros para a função deve ser por valor ou por referência? Justifique.

Deve ser por referência ou não seria possível retornar o resultado nos parâmetros.

```

void calcula(int* a, int* b)
{
    int soma = *a + *b;
    int subtracao = *a - *b;

    *a = soma;
    *b = subtracao;
}

```

### 9ª Questão

O programa abaixo declara uma matriz e uma string que são manipuladas por funções. Extenda o

programa de forma a incluir as seguintes funções:

- a) Crie uma função que receba três strings e que coloque na terceira o resultado da concatenação das outras duas. Altera a função principal (“main”) de forma que obtenha duas strings e mostre na tela as strings concatenadas.
- b) Crie uma função que obtenha um vetor que seja a soma das colunas;
- c) Crie uma função que obtenha a soma das linhas;
- d) Crie uma função que obtenha o maior valor de um vetor;

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

```
#define B 4
```

```
int SomaElementos(int xsize, int ysize, int array[][B]);
int ContaVogal(char string[]);
char* contatena(char* str1, char* str2, char* result);
```

```
int main(int argc, char *argv[])
{
    int sizex=B;
    int sizey=B;
    char nome[]="Jose Maria da Silva";
    int matriz[B][B] = { {1,2,3,4}, {5,6,7,8}, {9,10,11,12}, {13,14,15,16} };

    printf("A soma dos elementos da Matriz=%d\n", SomaElementos(sizex, sizey, matriz));
    printf("A quantidade de letras 'a' =%d\n", ContaVogal(&nome[0]));

    char str1[50], str2[50], result[100];
    scanf("%s", str1);
    scanf("%s", str2);
    printf("%s\n", contatena(str1, str2, result));

    system("PAUSE");
    return 0;
}
```

```
char* contatena(char* str1, char* str2, char* result)
{
    strcpy(result, str1);
    strcat(result, str2);

    return result;
}
```

```
int SomaElementos(int x, int y, int tabela[][B]){
    int i,j;
    int soma=0;
    for (i=0; i < x; i++) {
        for (j=0; j < y; j++) {
            printf("%d\n",tabela[i][j]);
            soma= soma + tabela[i][j];
        }
    }
    return soma;
}
```

```
int ContaVogal(char *frase){  
    int i,j;  
    int soma=0;  
    for (i=0; *(frase+i); i++) {  
        if (*(frase+i)=='a') {  
            soma = soma + 1;  
        }  
    }  
    return soma;  
}
```