

# Exercícios - LP

1. Suponha que `v` é um vetor declarado assim:

```
int v[]={1,2,3,4,5};
```

- a) Coloque V ou F:

```
v++; F (não pode haver atribuição a ponteiro constante)
v+1; V (pode-se fazer operação sem atribuição em matrizes)
p=v; V (sendo p um ponteiro, pode receber endereço do mesmo tipo)
v=p; F idem primeira resposta
```

- b) O que retorna?

```
*v+5; 6
*(v+5); "sujeira": endereço fora do array
```

2. Considere a seguinte matriz:

```
char nome[50]={ "prova"};
```

Qual a saída dos comandos abaixo?

```
printf("\n%c", *nome); p
printf("\n%s", nome); prova
printf("\n%c", *(nome+1)); r
printf("\n%s", nome+1); rova
```

3. Assumindo que `M1[]` é um vetor do tipo `int`, quais das seguintes expressões referenciam o valor do terceiro elemento de `M1`?

a) `*(M1 + 2)`      b) `*(M1 + 4)`      c) `M1 + 4`      d) `M1 + 2`

4. O que pode acontecer se for atribuído algum valor a um ponteiro que não tenha sido inicializado. Ex:

```
float *p;
*p = 2000;
Não funciona, já que ele não aponta para nenhum endereço conhecido pelo programa.
```

5. Qual a saída do programa abaixo? Faça o teste de mesa para descobrir:

```
# include <stdio.h>
int main()
{
    int t, i, M[3][4];
    for (t=0; t<3; ++t)
        for (i=0; i<4; ++i)
            M[t][i] = (t*4)+i+1;

    for (t=0; t<3; ++t)
    {
        for (i=0; i<4; ++i)
```

```

        printf ("%3d ", M[t][i]);
    printf ("\n");
}
return(0);
}

```

➔ Executar e conferir

6. Implemente uma função chamada subschar que substitua um caracter por outro dentro de uma string de até 10 caracteres. Usar notação ponteiro;  
 Exemplo de entrada: arara r v  
 Saída: avava

```

#include <stdio.h>

void subschar(char *texto, char char1, char char2)
{
    while (*texto != '\0')
    {
        if (*texto == char1)
            *texto=char2;
        texto++;
    }
}

int main()
{
    char c1[2], c2[2], texto[10];
    printf("Informe um texto: ");
    scanf("%s", texto);
    printf("Informe o caracter a ser excluido: ");
    scanf("%s", c1);
    printf("Informe o novo caracter: ");
    scanf("%s", c2);
    subschar(texto, *c1, *c2);
    printf("Nova string: %s", texto);
    return 0;
}

```