

LISTA SOBRE APONTADORES E VETORES

1 - Assumindo que **pulo1[]** é um vetor do tipo int, quais das seguintes expressões referenciam o valor do terceiro elemento do vetor ?

- (a) *(pulo1 +2) (b) *(pulo1 + 4) (c) pulo1 + 4 (d)pulo1 + 2

2- O que faz o programa seguinte quando executado?

```
main( )
{ int vet[ ] = {4, 5, 6};
  int j;
  for (j= 0; j < 3; j++)
    printf("%d", *(vet + j));
}
```

3- E este?

```
main( )
{ int vet[ ] = {4, 5, 6};
  int j;
  for (j= 0; j < 3; j++)
    printf("%d", (vet + j));
}
```

4- (a) O que faz o programa seguinte quando executado?

```
main( )
{   int vet[ ] = {4, 5, 6};
    int j, *ptr;
    ptr= vet;
    for (j= 0; j < 3; j++)
        printf("%d", *ptr++);
}
```

(b) O último comando poderia ser substituído por `printf("%d", *vet++);`

6. Seja **vet** um vetor de 4 elementos: **TIPO vet[4]**. Supor que depois da declaração, **vet** armazena o endereço de memória 4092 (ou seja, o endereço de `vet[0]`). Supor também que na máquina usada uma variável do tipo char ocupa 1 byte, do tipo int ocupa 2 bytes, do tipo float ocupa 4 bytes e do tipo double ocupa 8 bytes.

Qual o valor de `vet +1`, `vet +2` e `vet +3` se:

- (a) **vet** for declarado como *char* ?
(b) **vet** for declarado como *int* ?
(c) **vet** for declarado como *float* ?
(d) **vet** for declarado como *double* ?

7. Fazer uma função que leia e imprima uma frase e depois a imprima novamente ao contrário. Sua função deve ser genérica, ou seja, capaz de imprimir diferentes frases de diferentes tamanhos, no entanto, o tamanho da frase não deve ser fornecido. (Dica: use apontadores)

8. Faça um programa para ler um vetor de nomes e ordená-lo utilizando o algoritmo bubblesort. Utilize um vetor de apontadores para cada nome. Em vez de mudar os nomes, mude somente os apontadores.

9. O que faz a função abaixo?

```
int strfunc (char *st)
{
    char *pst = st;
    while (*pst)
        ++pst;
    return (pst - st);
}
```

10. Faça uma função strcmp que compara duas strings como implementado pela linguagem C com os seguintes protótipos:

```
int strcmp_1 (char *str1, char *str2)
int strcmp_2 (char str1[ ], char [ ] str2 [ ])
```

11. Faça uma função com o seguinte protótipo int isPalim (char *str) que retorna 1 se str for um palíndromo, e 0 caso contrário. Use dois apontadores um que percorre a string no sentido normal e outro no sentido inverso.

12. Faça um programa que receba como parâmetros (argumentos em linha de entrada) uma palavra e uma frase e veja se a palavra (ou subfrase) está contida na frase dada.

Exemplo de uso de argumentos de linha e comando

```
c> prog "Ana Banana" "Quem vai com Ana Banana!"
```

A resposta deve ser sim.

13. Faça um programa que receba como parâmetros (argumentos em linha de entrada) um inteiro x e três nomes de arquivos. Se x = 0 o terceiro arquivo deverá conter uma cópia do primeiro arquivo. Se x=1 um o terceiro arquivo deverá conter uma cópia do segundo arquivo. Se x=2 o terceiro arquivo receberá a concatenação dos dois arquivos fornecidos.

14. Repita os exercícios 7 e 8 usando alocação dinâmica.