

Algoritmos e Estrutura de Dados II

Lista 3

Aluno: Leonardo Aguilar Murça

1. Qual a diferença entre: `mat[3]` e `*(mat + 3)`?

A expressão `(mat + 3)` tem exatamente o mesmo valor de `mat[3]`, uma vez que ambos referenciam a posição 3 de um vetor. Porém, a primeira expressão utiliza notação de vetores e a segunda de ponteiros, já que todo vetor é entendido como um ponteiro pelos compiladores.*

2. Admitindo a declaração: `int mat[8]`; quais das seguintes expressões se referem ao valor do terceiro elemento da matriz?

- a) `*(mat + 2)`;
- b) `*(mat + 3)`;
- c) `mat + 2`;
- d) `mat + 3`.

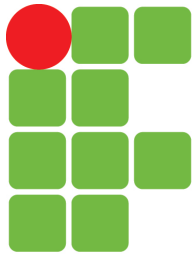
3. O que faz o programa seguinte?

```
#include <iostream>

using namespace std;

int main()
{
    int mat[]={4,5,6};
    for(int j=0; j<3; j++)
        cout << *(mat+j) << endl;
    return 0;
}
```

O programa acima declara e preenche um vetor de 3 posições. Logo após, ele imprime o valor de cada posição do vetor utilizando a notação de ponteiros `(mat+j)`.*



4. O que faz o programa seguinte?

```
#include <iostream>

using namespace std;

int main()
{
    int mat[]={4,5,6};
    for(int j=0; j<3; j++)
        cout << (mat+j) << endl;
    return 0;
}
```

O programa acima declara e preenche um vetor de 3 posições. Logo após, ele imprime o endereço de memória de cada posição do vetor. Por exemplo: **0x6dfef0**, **0x6dfef4**, **0x6dfef08**.

5. O que faz o programa seguinte?

```
#include <iostream>

using namespace std;

int main()
{
    int mat[]={4,5,6};
    int *p = mat;
    for(int j=0; j<3; j++)
        cout << *p++ << endl;
    return 0;
}
```

O programa acima declara e preenche um vetor de 3 posições e atribui à um ponteiro inteiro o valor da primeira posição desse mesmo vetor. Por fim, ele imprime o valor de cada posição de vetor.

6. Considerando a declaração:

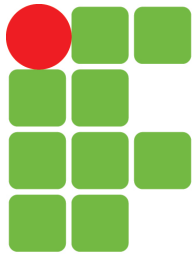
```
char *s = "Eu não vou sepultar Cesar";
```

o que imprimirão as seguintes instruções?

- a) `cout << s;` (**Saída:** Eu não vou sepultar César)
- b) `cout << &s[0];` (**Saída:** Eu não vou sepultar César)
- c) `cout<<(s+11);` (**Saída:** sepultar César) // Imprime a partir da posição 11 do vetor de caracteres.
- d) `cout << s[0];` (**Saída:** E) // Imprime o valor da primeira posição do vetor de caracteres.

7. Escreva a expressão `mat[i][j]` em notação ponteiro.

`mat(i * NC + j)` // Onde NC é o número de colunas.



8. Qual a diferença entre os seguintes protótipos de funções:

```
void func(char *p);  
void func(char p[]);
```

A única diferença entre os protótipos de funções acima é a notação de como é passado os parâmetros de um vetor. No primeiro exemplo é passado com notação de vetor e no segundo como notação de ponteiros.

9. Considerando a declaração:

```
char *items[5] = { "Abrir", "Fechar", "Salvar", "Imprimir", "Sair"};
```

Para poder escrever a instrução **p=items**; a variável p deve ser declarada como:

- a) char p;
- b) char *p;
- c) **char **p;**
- d) char ***p;
- e) char *p[];
- f) char **p[][];

10. O operador **new**:

- a) Cria uma variável de nome new;
- b) Retorna um ponteiro **void**;
- c) **Aloca memória para uma nova variável;**
- d) Informa a quantidade de memória livre.

11. O operador **delete**:

- a) Apaga um programa;
- b) **Devolve memória ao sistema operacional;**
- c) Diminui o tamanho do programa;
- d) Cria métodos de otimização.