



Algoritmos e Estrutura de Dados I

Profa. Simone Aires – Prof. Saulo

Lista de Exercício 02 - Ponteiros

1. Seja o seguinte trecho de programa:

```
int i=3, j=5;  
  
int *p, *q;  
  
p= &i;  
  
q= &j;
```

Qual é o valor das expressões?

- a) `p == &i`
- b) `*p -*q`
- d) `3*-*p/(*q-2)`

2. Qual é a instrução que deve ser adicionada ao programa abaixo para que ele funcione corretamente?

```
void main()  
{  
    int i, *ptri;  
    *ptri = 0;  
}
```

3. Assumindo que **o endereço de vox foi atribuído a um ponteiro variável invox**, quais das seguintes expressões são verdadeiras?

- a) `vox == &invoy`
- b) `vox == *invoy`
- c) `invoy == *vox`
- d) `invoy == vox`
- e) `&vox == &invoy`



Algoritmos e Estrutura de Dados I

Profa. Simone Aires – Prof. Saulo

4. Na expressão float *fptr; o que é do tipo float?

- (a) A variável fptr
- (b) O endereço de fptr
- (c) A variável apontada por fptr
- (d) Nenhuma das anteriores

5. Assumindo que pulo[] é um vetor do tipo int, quais das seguintes expressões referenciam o valor do terceiro elemento do vetor?

- (a) *(pulo + 2);
- (b) *(pulo + 4);
- (c) pulo + 4;
- (d) pulo + 2;

6. Considerando a declaração abaixo, quais das seguintes expressões são válidas? Justifique.

int mat[4], *p, x;

- (a) p = mat + 1;
- (b) p = mat++;
- (c) p = ++mat;
- (d) x = (*mat)++;

7. O que fazem os seguintes programas em C?

```
int main() {  
    int vet[] = {4, 9, 13};  
    int i;  
    for(i=0; i<3; i++) {  
        printf("%d ", *(vet+i));  
    }  
}
```

```
int main() {  
    int vet[] = {4, 9, 13};  
    int i;  
    for(i=0; i<3; i++) {  
        printf("%X ", vet+i);  
    }  
}
```



Algoritmos e Estrutura de Dados I

Profa. Simone Aires – Prof. Saulo

8. Seja x um vetor de 4 elementos, declarado da forma TIPO $x[4]$. Suponha que depois da declaração, x esteja armazenado no endereço de memória 4092 (ou seja, o endereço de $x[0]$). Suponha também que na máquina seja 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. Quais serão os valores de $x+1$, $x+2$ e $x+3$ se:

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

9. Seja o código abaixo:

```
#include <stdio.h>

int main(void) {

    int mat[3] = {5, 9, 11};

    int *ptr;

    ptr = mat;

    printf(" %d", *ptr + 1);

    printf(" %d", *(ptr + 1));

    return 0; 1

}
```

O que será impresso? E para o código abaixo?

```
ptr = mat;

printf(" %d", *(ptr));

printf(" %d", *(ptr + 2));
```



Algoritmos e Estrutura de Dados I

Profa. Simone Aires – Prof. Saulo

10. Levando em consideração o trecho a seguir:

```
int *p;
```

```
int i;
```

```
int k;
```

```
i = 42;
```

```
k = i;
```

```
p = &i;
```

Qual das alternativas irá alterar o valor de i para 120?

a) k = 120;

b) *k = 120;

c) p = 120;

d) *p = 120;

e) Mais de uma resposta irá mudar o valor de i para 120.

11. Dado o programa abaixo, diga o valor das variáveis após cada atribuição. Tome um valor aleatório como endereço de vetor.

```
int main()
```

```
{
```

```
    char vetor[3] = {'A','B','C'};
```

```
    char *j = &vetor[0];
```

```
    char aux;
```

```
    aux = vetor[0];
```

```
    aux = *(vetor + 2);
```

```
    aux = *(j + 1);
```

```
    aux = *j;
```

```
    j = vetor + 1;
```

```
    aux = *j;
```

```
    aux = *(j + 1);
```

```
    j = vetor;
```

```
    aux = *++j;
```

```
    aux = ++*j;
```

```
    aux = *j++;
```

```
    aux = *j;
```

```
    return 0;
```

```
}
```



12. Critique o código abaixo.

```
void troca (int *i, int *j)
{
    int *temp;
    *temp = *i;
    *i = *j;
    *j = *temp;
}
int main ()
{
    int a, b;
    a = 0;
    b = 1;
    troca(&a, &b);
    return 0;
}
```

13. Faça um programa que leia 2 valores inteiros e chame uma função que receba estas 2 variáveis e troque o seu conteúdo, ou seja, esta função é chamada passando duas variáveis A e B por exemplo e, após a execução da função, A conterá o valor de B e B terá o valor de A. **OBS:** resolvido no slide de aula... é ó testar!

14. Escreva uma função que calcule o perímetro de um retângulo ($P=2b+2h$). A função deverá retornar o valor do perímetro calculado. Após o cálculo do perímetro altere o valor das variáveis base e altura para: $b=2*b$ e $h=2*h$. No programa principal imprima o valor do perímetro calculado e os novos valores da base e da altura. **OBS:** Utilize passagem de parâmetros por referência.

Protótipo: float perímetro(float *a, float *b)

15. Crie uma função que receba um ponteiro para um array de float com 10 elementos. Imprima o endereço de cada posição desse array.

16. Escreva uma função `double* max(double* a, int size)` que retorne um ponteiro para o maior valor contido em um array de elementos do tipo double, se o array estiver vazio, retorne NULL.

17. Fazer um procedimento que recebe como parâmetro uma matriz com 4 notas de 20 alunos e um vetor de 20 posições. Esta função deve encontrar a média de cada aluno e armazenar em sua respectiva posição no vetor. Ao final (no main()) deverá imprimir a média de cada aluno contida no vetor. Protótipo:

`void calcula_media(float *matriz, float *vetor).` //E para uma matriz de tamanho n ??