

Atividade De Estrutura de Dados

Estudo dirigido: Lista de Ponteiros 1

Aluno: William Cardoso Barbosa

Curso: Ciência da Computação

Conteúdo/Atividade: Lista de Ponteiros 1

1-O que é um ponteiro? E para que serve?

R= Ponteiros são variáveis que apontam para outras a partir do armazenamento do endereço de memória da variável que está sendo referenciada. O ponteiro é usado quando queremos acessar determinada variável em diferenciadas partes do programa.

2-Qual das instruções abaixo é correta para declarar um ponteiro para inteiro?

- a) `*int pti;`
- b) `*pti;`
- c) `&i;`
- d) `int_pti pti;`
- e) `int *pti;`**

3-Seja o seguinte trecho de programa:

```
int i=3, j=5;  
int *p, *q;  
p = &i; → recebe o endereço de memória de i  
q = &j; → recebe o endereço de memória de j
```

Qual é o valor das seguintes expressões?

- a) $p == \&i \rightarrow 1$ (verdadeiro)
- b) $*p - *q \rightarrow -2$
- c) $*\&p \rightarrow$ endereço de memória de p.
- d) $3 - *p / (*q) + 7 \rightarrow 3 (- 3 / (5)) + 7 \rightarrow 10$

4- Seja a seguinte sequência de instruções:

```
int i=10, j=20;  
int *pti, *ptj;  
pti = &i;  
ptj = &j;
```

Qual expressão não é válida?

- a) $i = pti == ptj$;
- b) $i = pti - ptj$;
- c) $pti += ptj$;
- d) $pti++$;
- e) $i = pti || ptj$;

5- Seja a seguinte sequência de instruções em programa C:

```
int *pti;  
int i=10;  
pti = &i;
```

Qual afirmativa é falsa?

- a) pti armazena o endereço de i

- b) *pti é igual a 10
- c) Ao se executar *pti=20; i passará a ter o valor 20
- d) Ao se alterar o valor de i, *pti será modificado
- e) pti é igual a 10

6- Qual serão as saídas do seguinte programa?

```
#include <conio.h>
#include <stdio.h>
void main(){
float vet[5] = {1.1,2.2,3.3,4.4,5.5};
float *f;
int i;
f = vet;
printf("contador/valor/valor/endereco/endereco");
for(i = 0 ; i <= 4 ; i++){
    printf("\n i = %d",i);
    printf(" vet[%d] = %.1f",i, vet[i]);
    printf(" *(f + %d) = %.1f",i, *(f+i));
    printf(" &vet[%d] = %X",i, &vet[i]);
    printf(" (f + %d) = %X",i, f+i);
}
}

/*

saida

i = 0 vet[0] = 1.1 *(f + 0) = 1.1 &vet[0] = 61FF04 (f + 0) = 61FF04
i = 1 vet[1] = 2.2 *(f + 1) = 2.2 &vet[1] = 61FF08 (f + 1) = 61FF08
i = 2 vet[2] = 3.3 *(f + 2) = 3.3 &vet[2] = 61FF0C (f + 2) = 61FF0C
i = 3 vet[3] = 4.4 *(f + 3) = 4.4 &vet[3] = 61FF10 (f + 3) = 61FF10
i = 4 vet[4] = 5.5 *(f + 4) = 5.5 &vet[4] = 61FF14 (f + 4) = 61FF14

*/
```

7- O que fazem os seguintes programas?

a)

```

#include <conio.h>
#include <stdio.h>
void main(){
    int vet[] = {4,9,13};
    int i;
    for(i=0;i<3;i++){
        printf("%d ",*(vet+i)); // Na pega o endereço de memória da primeira posição do array "vet" e acrescenta + 1 posição a cada iteração no loop. Logo em seguida é usada a simbologia de ponteiro para pegar o valor que o endereço referencia.
    }
}

/*
imprime os 3 valores do vetor "vet" na mesma linha com espaço.

saida: 4 9 13

*/

```

b)

```

#include <conio.h>
#include <stdio.h>
void main(){
    int vet[] = {4,9,13};
    int i;
    for(i=0;i<3;i++){
        printf("%X ",vet+i);
    }
}

/*
imprime o endereço de memória, na mesma linha, de cada elemento do vetor, começando do primeiro.

*/

```

c)

```

#include <conio.h>
#include <stdio.h>
void main(){
    int vet[] = {4,9,13};
    int i;
    for(i=0;i<3;i++){
        printf("%X ",vet+i);
    }
}

```

```

    }
}

/*
imprime o endereço de memória, na mesma linha, de cada elemento do vetor, começando do primeiro.
*/

```

8- O que fazem os seguintes programas?

a)

```

#include <conio.h>
#include <stdio.h>
void main() {
int vet[] = {4,9,12};
int i,*ptr;
ptr = vet;
for(i = 0 ; i < 3 ; i++) {
    printf("%d ",*ptr++);
}
}

/*
para cada loop incrementa uma posição na memória e pega o valor correspondente.
*/

```

b)

```

#include <conio.h>
#include <stdio.h>
void main(){
int vet[] = {4,9,12};
int i,*ptr;
ptr = vet;
for(i = 0 ; i < 3 ; i++) {
    printf("%d ",(*ptr)++);
}
}

/*
por causa da ordem de precedência dos parenteses, *(prt) sempre irá pegar o primeiro elemento do vetor, pois quando fazemos referência a um vetor sem citar a posição do elemento el

```

e sempre retornará o primeiro elemento, dessa forma, o ++ incrementará +1 no primeiro elemento 3 vezes.

saida: 4 5 6

/*

9- Seja vet um vetor de 4 elementos: TIPO vet[4]. Supor que depois da declaração, vet esteja

armazenado no 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?

vet + 1 = 4093. vet + 2 = 4094. vet + 3 = 4095

b) vet for declarado como int?

vet + 1 = 4094. vet + 2 = 4096. vet + 3 = 4098

c) vet for declarado como float?

vet + 1 = 4096. vet + 2 = 4100. vet + 3 = 4104

d) vet for declarado como double?

vet + 1 = 4100. vet + 2 = 4108. vet + 3 = 4116

10- Qual será a saída deste programa supondo que i ocupa o endereço 4094 na memória?

```
main() {
int i=5, *p;
p = &i;
printf("%x %d %d %d %d \n", p, *p+2, &p ,3p, **&p+4);
}

/*
4094 7 40xx 40xx
```

* /

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

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

12. Supor a declaração: int mat[4], *p, x; Quais expressões são válidas? Justifique:

a) p = mat + 1;

b) p = mat++;

c) p = ++mat;

d) x = (*mat)++;

b e c estão tentando incrementar em um endereço de memória.