

Pré-Lista de 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 seguintes expressões ?

a) $p == \&i;$ b) $*p - *q$ c) $**\&p$ d) $3* - *p / (*q) + 7$

2. 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, 3**p, **&p+4);
}
```

3. Se **i** e **j** são variáveis inteiras e **p** e **q** ponteiros para **int**, quais das seguintes expressões de atribuição são ilegais?

a) $p = \&i;$ b) $*q = \&j;$ c) $p = \&* \&i;$ d) $i = (*\&)j;$
e) $i = *\&j;$ f) $i = *\&* \&j;$ g) $q = *p;$ h) $i = (*p)++ + *q$

4. Qual serão as saídas do seguinte programa?

```
#include <stdio.h>
#include <conio.h>

int main() {
    int    valor;
    int    *p1;
    float  temp;
    float  *p2;
    char   aux;
    char   *nome = "Algoritmos";
    char   *p3;
    int     idade;
    int     vetor[3];
    int     *p4;
    int     *p5;
```

```

/* (a) */
valor = 10;
p1 = &valor;
*p1 = 20;
printf("(a) %d \n", valor);

/* (b) */
temp = 26.5;
p2 = &temp;
*p2 = 29.0;
printf("(b) %.1f \n", temp);

/* (c) */
p3 = &nome[0];
aux = *p3;
printf("(c) %c \n", aux);

/* (d) */
p3 = &nome[4];
aux = *p3;
printf("(d) %c \n", aux);

/* (e) */
p3 = nome;
printf("(e) %c \n", *p3);

/* (f) */
p3 = p3 + 4;
printf("(f) %c \n", *p3);

/* (g) */
p3--;
printf("(g) %c \n", *p3);

/* <h> */
vetor[0] = 31;
vetor[1] = 45;
vetor[2] = 27;
p4 = vetor;
idade = *p4;
printf("(h) %d \n", idade);

/* (i) */
p5 = p4 + 1;
idade = *p5;
printf("(i) %d \n", idade);

/* (j) */
p4 = p5 + 1;
idade = *p4;
printf("(j) %d \n", idade);

/* (l) */
p4 = p4 - 2;
idade = *p4;
printf("(l) %d \n", idade);

/* (m) */
p5 = &vetor[2] - 1;
printf("(m) %d \n", *p5);

/* (n) */
p5++;

```

```
printf("(n) %d \n", *p5);

return(0);
}
```

5. Qual é o resultado 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("\ni = %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);
    }
}
```

6. 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`

7. 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)++;`

8. O que fazem os seguintes programas?

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

<pre>printf("%d ",*(vet+i)); } }</pre>	<pre>printf("%X ",vet+i); } }</pre>	<pre>printf("%X ",vet+i); } }</pre>
--	-------------------------------------	-------------------------------------

9. O que faz o seguinte programa quando executado?

<pre>#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++); } }</pre>	<pre>#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)++); } }</pre>
(a)	(b)

10. 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:

- vet** for declarado como `char`?
- vet** for declarado como `int`?
- vet** for declarado como `float`?
- vet** for declarado como `double`?