



Exercícios

Ponteiros

- Se i é uma variável e p é um ponteiro que aponta para i , quais das seguintes expressões são apelidos para i ?
 - $*p$
 - $\&p$
 - $*\&p$
 - $\&*p$
 - $*i$
 - $\&i$
 - $*\&i$
 - $\&*i$
- Se i é uma variável do tipo *int* e p e q são ponteiros para *int*, quais das seguintes atribuições são corretas?
 - $p = i;$
 - $*p = \&i;$
 - $\&p = q;$
 - $p = \&q;$
 - $p = *\&q;$
 - $p = q;$
 - $p = *q;$
 - $*p = q;$
 - $*p = *q;$
- Explique a diferença entre:
 $p++;$ $(*p)++;$ $*(p++)$;
- Suponha que as declarações e atribuições simultâneas tenham sido realizadas nas variáveis listadas abaixo:

```
int v[] = 5, 15, 34, 54, 14, 2, 52, 72;  
int *p = &v[1], *q = &v[5];
```

 - Qual o valor de $*(p+3)$?
 - Qual o valor de $*(q-3)$?
 - Qual o valor de $q-p$?
 - A expressão $p < q$ tem valor verdadeiro ou falso?
 - A expressão $*p < *q$ tem valor verdadeiro ou falso?

5. Qual o conteúdo do vetor v após a execução do seguinte trecho de código?

```
1  #define N 10
2  int main( )
3  {
4      int v[N] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
5      int *p = &v[0], *q = &v[N - 1], temp;
6      while (p < q) {
7          temp = *p;
8          *p++ = *q;
9          *q-- = temp;
10     }
11     return 0;
12 }
```

6. Suponha que v é um vetor e p é um ponteiro. Considere que a atribuição $p = v$; foi realizada previamente. Quais das expressões abaixo não são permitidas? Das restantes, quais têm valor verdadeiro?

- (a) $p == v[0]$
- (b) $p == \&v[0]$
- (c) $*p == v[0]$
- (d) $p[0] == v[0]$

7. Escreva um programa que leia um vetor de inteiros e o imprima em ordem reversa. Pare quando encontrar o valor 0 (zero). Faça o programa de forma a usar um ponteiro, ao invés de um índice como um número inteiro, para controlar a posição corrente no vetor.

8. Qual a saída do programa descrito abaixo?

```
1  #include <stdio.h>
2  struct dois_valores {
3      int vi;
4      float vf;
5  };
6  int main(void)
7  {
8      struct dois_valores reg1 = {53, 7.112}, reg2, *p = &reg1;
9      reg2.vi = (*p).vf;
10     reg2.vf = (*p).vi;
11     printf("1: %d %f\n2: %d %f\n", reg1.vi, reg1.vf, reg2.vi, reg2.vf);
12     return 0;
13 }
```

9. Simule a execução do programa descrito abaixo.

```
1  #include <stdio.h>
2  struct pts {
3      char *c;
4      int *i;
5      float *f;
6  };
7  int main( )
8  {
9      char caractere;
10     int inteiro;
11     float real;
12     struct pts reg;
13     reg.c = &caractere;
14     reg.i = &inteiro;
15     reg.f = &real;
16     scanf("%c%d%f", reg.c, reg.i, reg.f);
17     printf("%c\n%d\n%f\n", caractere, inteiro, real);
18     return 0;
19 }
```

10. Simule a execução do programa descrito abaixo.

```
1  #include <stdio.h>
2  struct celula {
3      int valor;
4      struct celula *prox;
5  };
6  int main(void)
7  {
8      struct celula reg1, reg2, *p;
9      scanf("%d%d", &reg1.valor, &reg2.valor);
10     reg1.prox = &reg2;
11     reg2.prox = NULL;
12     for (p = &reg1; p != NULL; p = p->prox)
13         printf("%d ", p->valor);
14     printf("\n");
15     return 0;
16 }
```

11. Escreva um procedimento que receba um número inteiro n , com $1 \leq n \leq 100$, e um vetor v com $n > 0$ números inteiros e devolva um maior e um menor dos elementos desse vetor. Sua função deve ter a seguinte interface:

```
void min_max(int n, int v[ ], int *max, int *min)
```



12. Escreva um procedimento que receba um número inteiro n , com $1 \leq n \leq 100$, e um vetor v com $n > 0$ números inteiros e devolva um maior e um segundo maior elementos desse vetor. Sua função deve ter a seguinte interface:

```
void dois_maiores(int n, int v[ ], int *p_maior, int *s_maior)
```

13. Escreva um procedimento que receba dois números inteiros a e b e devolva a soma e o produto destes dois números. Sua função deve ter a seguinte interface:

```
void soma_prod(int a, int b, int *soma, int *prod)
```

14. Escreva uma função que receba um número inteiro n , com $1 \leq n \leq 100$, e um vetor v de n números inteiros e devolva o endereço do elemento de v onde reside um maior elemento de v . Sua função deve ter a seguinte interface:

```
int *maximo(int n, int v[ ])
```

15. Escreva, para cada uma das questões anteriores (11, 12, 13 e 14), um algoritmo que realize a chamada ao módulo criado e exiba seu resultado.