

Lista de Exercícios

22) Ponteiros

a) Compile e execute o seguinte programa.

```
1.  int main (void) {
2.      int i; int *p;
3.      i = 1234; p = &i;
4.      printf ("*p = %d\n", *p);
5.      printf (" p = %ld\n", (long int) p);
6.      printf (" p = %p\n", (void *) p);
7.      printf ("&p = %p\n", (void *) &p);
8.      return EXIT_SUCCESS;
9.  }
```

b) Por que o código abaixo está errado?

```
void troca (int *i, int *j) {
    int *temp;
    *temp = *i; *i = *j; *j = *temp;
}
```

c) Um ponteiro pode ser usado para dizer a uma função onde ela deve depositar o resultado de seus cálculos. Escreva uma função hm que converta minutos em horas-e-minutos. A função recebe um inteiro mnts e os endereços de duas variáveis inteiras, digamos h e m, e atribui valores a essas variáveis de modo que m seja menor que 60 e que $60 \cdot h + m$ seja igual a mnts. Escreva também uma função main que use a função hm.

d) Escreva uma função mm que receba um vetor inteiro v[0..n-1] e os endereços de duas variáveis inteiras, digamos min e max, e deposite nessas variáveis o valor de um elemento mínimo e o valor de um elemento máximo do vetor. Escreva também uma função main que use a função mm.

e) O que faz a seguinte função?

```
void imprime (char *v, int n) {
    char *c;
    for (c = v; c < v + n; c++)
        printf ("%c", *c);
}
```

f) O programa abaixo produziu a seguinte resposta, que achei surpreendente:

x: 111

v[0]: 999

Os valores de x e v[0] não deveriam ser iguais?

```
void func1 (int x) {
    x = 9 * x;
}
```

```
void func2 (int v[]) {
    v[0] = 9 * v[0];
}
```

```
int main (void) {
    int x, v[2];
    x = 111;
```

```

func1 (x); printf ("x: %d\n", x);
v[0] = 111;
func2 (v); printf ("v[0]: %d\n", v[0]);
return EXIT_SUCCESS;
}

```

g) O seguinte fragmento de código pretende decidir se “abacate” vem antes ou depois de “uva” no dicionário. O que está errado?

```

1. char *a, *b;
2. a = "abacate"; b = "uva";
3. if (a < b)
4.     printf ("%s vem antes de %s\n", a, b);
5. else
6.     printf ("%s vem depois de %s\n", a, b);

```

h) Compile e execute o seguinte programa:

```

int main (void) {
    typedef struct {
        int dia, mes, ano;
    } data;
    printf ("sizeof (data) = %d\n",
        sizeof (data));
    return EXIT_SUCCESS;
}

```

i) Compile e execute o seguinte programa. (O [cast](#) (long int) é necessário para que &i possa ser impresso com [especificação de formato](#) %ld. É mais comum imprimir endereços em [notação hexadecimal](#), usando a especificação de formato %p, que exige o cast (void *).)

```

int main (void) {
    int i = 1234;
    printf (" i = %d\n", i);
    printf ("&i = %ld\n", (long int) &i);
    printf ("&i = %p\n", (void *) &i);
    return EXIT_SUCCESS;
}

```