

Ficha 5

1. Sejam i é uma variável inteira e p e q apontadores para inteiros. Indique quais das seguintes atribuições são legais.
(a) $p = i$; (b) $*p = i$; (c) $p = \&q$; (d) $p = \&i$;
(e) $\&i = p$; (f) $p = q$; (g) $p = *q$; (h) $*p = q$;
(i) $*p = *q$;
2. Sejam i uma variável `int` e p é um apontador para i . Indique quais das seguintes instruções de leitura e escrita são corretas.
(a) `printf("%d", i);`
(b) `scanf("%d", i);`
(c) `scanf("%d", &i);`
(d) `printf("%d", p);`
(e) `printf("%d", *p);`
(f) `scanf("%d", p);`
(g) `printf("%p", p);`
(h) `printf("%p", i);`
(i) `printf("%p", *p);`
3. Escreva uma função
`void decompor(int total_seg, int *horas, int *mins, int *secs);`
que decompõe um total inteiro de segundos `total_seg` em horas, minutos (0-59) e segundos (0-59); os resultados devem ser atribuídos ao conteúdo dos apontadores `horas`, `mins` e `secs`. Pode assumir que o total de segundos é maior que zero.
4. Escreva uma função
`void max_min(int vec[], int size, int *pmax, int *pmin);`
que determina o valor máximo e mínimo de um array; os resultados devem ser atribuídos ao conteúdo dos apontadores `pmax` e `pmin`. Pode assumir que `size` é sempre maior que zero.
5. Escreva uma função `void reduzir(int *pnun, int *pdenom)` que reduz uma fração de numerador e denominador `*pnun` e `*pdenom` à forma simplificada. O numerador e denominador devem ser modificados por meio dos apontadores `pnun` e `pdenom`. Pode assumir que o denominador é sempre diferente de zero.
Sugestão: para simplificar uma fração basta dividir o numerador e o denominador pelo seu máximo divisor comum.