

Essa lista de exercícios tem como objetivo principal desenvolver algoritmos a partir dos conteúdos abordados em sala de aula. Todos os exercícios também devem ser implementados em linguagem C. Nos programas que pedem para implementar apenas funções desenvolva também o programa principal (main) para testá-los. Não utilize variáveis globais.

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 92572 na memória?
- ```
main() {
int i=5, *p;
p = &i;
printf(“%x%d%d %d %d”, p,*p+2,**&p,3**p,**&p+4);
}
```
3. Se  $i$  e  $j$  são variáveis inteiras e  $p$  e  $q$  ponteiros para  $\text{int}$ , 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. Desenvolva um programa para testar as funções a seguir:

- Uma função que receba dois números  $a$  e  $b$ , em seguida, faça a troca destes dois números. Dica:  $a$  e  $b$  devem ser passados por referência.
- Uma função que receba dois números  $a$  e  $b$ , em seguida, decremente o primeiro e incremente o segundo. Dica:  $a$  e  $b$  devem ser passados por referência.
- Uma função que receba o raio, perímetro e área de um círculo. Em seguida, calcule e retorne nas variáveis perímetro e área o respectivo perímetro e área do círculo. Dica: perímetro e área devem ser passados por referência.

- Uma função que receba o lado, perímetro e área de um quadrado. Em seguida, calcule e retorne nas variáveis perímetro e área o respectivo perímetro e área do quadrado. Dica: perímetro e área devem ser passados por referência.
  - Uma função que receba os valores  $a$ ,  $b$  e  $c$  passados por valor, receba também dois valores  $x1$  e  $x2$  passados por referência. Em seguida, calcule e retorne as duas raízes da equação do segundo grau nas variáveis  $x1$  e  $x2$ .
5. Desenvolva um programa que receba coordenadas cartesianas digitadas pelo usuário e imprima seus valores convertidos para coordenadas polares. O usuário poderá digitar quantas coordenadas ele desejar. Utilize uma função que converte de coordenada cartesiana  $(x, y)$  para coordenada polar  $(r, s)$ .
  6. Desenvolva um programa que primeiro utilize uma função para gerar um vetor com tamanho definido pelo usuário e valores aleatórios dentro de um intervalo  $[min, max]$  determinado pelo usuário. Em seguida, uma outra função deve receber o vetor gerado, o tamanho do vetor e dois números *maior* e *menor*. Retorne nas variáveis *maior* e *menor* o menor e maior elemento do vetor. Dica1: *menor* e *maior* devem ser passados por referência. Dica2: use apenas uma estrutura de repetição para achar o menor e maior elemento.
  7. Desenvolva um programa que utilize uma função para inverter os elementos de um vetor (com valores gerados usando a função do exercício anterior) sem utilizar um vetor auxiliar. Apenas o vetor fornecido à função deverá ser utilizado.
  8. Desenvolva um programa para testar uma função com protótipo *void somabit(int b1, int b2, int \*vaium, int \*soma)*; A função recebe três bits (inteiros 0 ou 1)  $b1$ ,  $b2$  e  $*vaium$ . A função retorna um bit soma que armazena o resultado da soma dos três primeiros e o bit "vai-um" em  $*vaium$ .
  9. Desenvolva um programa que leia a quantidade total de segundos e converta para Horas, Minutos e Segundos. Imprima o resultado da conversão no formato HH:MM:SS. Para isso, utilize a função com protótipo *void converteHora(int total\_segundos, int \*hora, int \*min, int \*seg)*.
  10. Desenvolva um programa que utilize os parâmetros argv e argc. Logo, o programa principal (*main*) deverá receber o dia, mês e ano. Utilizando funções, o programa deverá verificar se os valores fornecidos formam uma data válida. Se for válida, o programa principal imprime a data como ilustrado a seguir:  
**Entrada:** 01/11/2011  
**Saída:** 01 de novembro de 2011
  11. Desenvolva um programa que utilize os parâmetros argv e argc para informar à função principal a dimensão  $M \times N$  de uma matriz e os valores *min* e *max*. Em seguida, o programa principal utiliza uma função para gerar a matriz  $M \times N$  com valores aleatórios dentro de um intervalo  $[min, max]$ . Uma outra função deve receber a matriz gerada e retornar o maior elemento da matriz, o menor elemento da matriz e o valor médio das entradas da matriz.