

Pontifícia Universidade Católica de Minas Gerais  
Instituto de Ciências Exatas e Informática – ICEI  
Algoritmos e Estrutura de Dados I

AED1 - Preparação 01

Tema: Representações de dados e fundamentos de programação

Exercícios:

Usar o modelo de lista de métodos para desenvolver e testar programas em linguagem C para atender aos enunciados que se seguem.

```
/*
Funcao principal.
@return codigo de encerramento
@param argc - quantidade de parametros na linha de comandos
@param argv - arranjo com o grupo de parametros na linha de comandos
*/
int main ( int argc, char* argv [ ] )
{
    // definir dado
    int opcao = 0

    // identificar
    printf ( "%s\n", "Exemplo0100 - Programa = v0.0" );
    printf ( "%s\n", "Autor: _____" );
    printf ( "\n" );           // mudar de linha

    // acoes

    // repetir
    do
    {
        // para mostrar opcoes
        printf ( "\n%s\n", "Opcoes:" );
        printf ( "\n%s" , "0 - Terminar" );
        printf ( "\n%s" , "1 - Method_01" );
        printf ( "\n%s" , "2 - Method_02" );
        printf ( "\n%s" , "3 - Method_03" );
        printf ( "\n%s" , "4 - Method_04" );
        printf ( "\n%s" , "5 - Method_05" );
        printf ( "\n" );

        // ler a opcao do teclado
        printf ( "\n%s", "Opcao = " );
        scanf ( "%d", &opcao );
        getchar( );           // para limpar a entrada de dados

        // para mostrar a opcao lida
        printf ( "\n%s%d", "Opcao = ", opcao );
    }
```

```

// escolher acao dependente da opcao
switch ( opcao )
{
    case 0: /* nao fazer nada */ break;
    case 1: method_01 ( ); break;
    case 2: method_02 ( ); break;
    case 3: method_03 ( ); break;
    case 4: method_04 ( ); break;
    case 5: method_05 ( ); break;
    default: // comportamento padrao
        printf ( "\nERRO: Opcao invalida.\n" );
        break;
} // end switch
}
while ( opcao != 0 );

// encerrar
printf ( "\n\nApertar ENTER para terminar." );
getchar(); // aguardar por ENTER
return ( 0 ); // voltar ao SO (sem erros)
} // end main ( )

/*
----- documentacao complementar

----- notas / observacoes / comentarios

----- previsao de testes

a.) a
b.) 4.2
c.) 10
d.) 1
e.) abc def
f.) abc def

----- historico

Versao      Data      Modificacao
0.1         _/_/___  esboco

----- testes

Versao      Teste
0.1         01. ( OK )  identificacao de programa

*/

```

Exercícios:

DICAS GERAIS: Consultar os Anexos C 01 e C 02 ou na apostila o capítulo 05 para outros exemplos.  
Prever, testar e registrar todos os dados e os resultados obtidos.

01.) Definir e testar um procedimento (01) para:

- ler pares de valores inteiros positivos do teclado;
  - parar quando o par for igual a (0,0);
  - supor que o primeiro valor representará o raio de uma circunferência,
  - e o segundo valor representará em quantas partes dividi-la;
  - calcular o comprimento de uma dessas partes
- Usar constantes reais (**double**) em expressões que envolvam valores reais (**double**).

Exemplo: (5,2) -> calcular o comprimento da metade (2) da circunferência de raio igual a 5.

02.) Definir e testar um procedimento (02) para:

- ler a quantidade de testes a serem feitos;
  - para cada teste, ler três valores literais separadamente
  - (um caractere em cada) do teclado e
- dizer se estão em ordem crescente de seus códigos ASCII.  
Usar variável do tipo lógica (**bool**). A repetição deverá usar o comando (**while**).

Exemplo: { 5, ('a','e','c'), ('e','a','c'), ('a','c','e'), ('e','c','a'), ('a','e','a') }

03.) Definir e testar um procedimento (03) para:

- ler dois valores inteiros (a,b), limites para intervalo [a:b];
  - ler uma quantidade (n) de valores inteiros a serem testados;
  - ler outros tantos valores quantos os indicados pela quantidade, um por vez (x);
  - contar e mostrar quantos dentre esses valores lidos (x)
- possuem a seguinte característica:  
serem pares e sua metade e dobro estão contidos no intervalo.  
A repetição deverá usar o comando (**for**).  
Exemplo: [ 12: 72 ], e n = 5, com { 25, 27, 30, 35, 36 }

04.) Definir e testar um procedimento (04) para:

- ler uma sequência de caracteres do teclado;
  - contar e mostrar todos os símbolos alfanuméricos (letras, dígitos e espaço em branco)
- contidos na cadeia de caracteres por meio de uma função inteira.  
Usar o comando **fgets(s,80-1,stdin)** para a leitura.  
Usar repetição para testar cada caractere do último para o primeiro.  
Definir e usar funções lógicas para testar cada condição:  
letras maiúsculas, minúsculas, dígitos e espaço em branco.  
NÃO usar qualquer função predefinida (**ctype.h**).

Exemplo: sequência = (A1b2 + C3d4) \* E5f6

05.) Definir e testar um procedimento (05) para refazer o exercício (04)  
usando uma função recursiva em lugar da repetição: