

```

#
# OBS.: RETIRAR OS COMENTARIOS "" ""
# PARA TESTAR CADA EXEMPLO INDIVIDUALMENTE.
#

""
# ----- EXEMPLO101

print "EXEMPLO101 - PRIMEIRO EXEMPLO EM PYTHON";
raw_input ( "\nPRESSIONAR <Enter> PARA TERMINAR." );

# fim do programa
""
""
# ----- EXEMPLO102
import os
os.system ( "cls" );      # ( ou "clear" ) para limpar a tela
                        # (dependente do sistema operacional)
print "EXEMPLO102 - PRIMEIRO EXEMPLO EM PYTHON";
os.system ( "pause" );   # ( ou getch ( ) ) para esperar
                        # (dependente do sistema operacional)

# fim do programa
""
""
# ----- EXEMPLO103

print "EXEMPLO103 - PRIMEIRO EXEMPLO EM PYTHON";
print                # para mudar de linha
print "MATRICULA: _____ ALUNO : _____ ";
print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );        # para esperar
# fim do programa
""
""
# ----- EXEMPLO104

print "EXEMPLO104 - PRIMEIRO EXEMPLO EM PYTHON";
print '\n';           # para mudar de linha (="\n")
print "MATRICULA: _____ ALUNO : _____ ";
print                # para mudar de linha
print "PRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );        # para esperar
# fim do programa
""
""
# ----- EXEMPLO105

print "EXEMPLO105 - PRIMEIRO EXEMPLO EM PYTHON";
print "\nMATRICULA: _____ ALUNO : _____ ";
print "\nEXEMPLOS DE VALORES : ";
print "\nCARACTERE : " , 'A';                # letra ou simbolo
print "\nINTEIRO    : " , 10;                 # valor sem parte fracionaria
print "\nREAL       : " , 3.1415;              # valor com parte fracionaria
# para esperar
raw_input ( "\nPRESSIONAR <Enter> PARA TERMINAR." );
# fim do programa
""

```

```

"""
# ----- EXEMPLO106
from math import pi

PI = pi          # nome para substituir valor

print "EXEMPLO106 - PRIMEIRO EXEMPLO EM PYTHON";
print "\nMATRICULA: _____ ALUNO : _____ ";
print "\nEXEMPLOS DE VALORES : ";
print "\nCARACTERE : " , 'A';          # letra ou simbolo
print "\nINTEIRO      : " , 10;         # valor sem parte fracionaria
print "\nREAL         : " , PI;         # emprego de macro

# para esperar
raw_input ("\nPRESSIONAR <Enter> PARA TERMINAR.");
# fim do programa
"""

# ----- EXEMPLO107
import math

print "EXEMPLO107 - PRIMEIRO EXEMPLO EM PYTHON";
print "\nMATRICULA: _____ ALUNO : _____ ";
print "\nEXEMPLOS DE VALORES : ";
print "\nCARACTERE : " , 'A';          # letra ou simbolo
print "\nINTEIRO      : " , 10;         # valor sem parte fracionaria
print "\nREAL         : " , math.pi;    # constante real

# para esperar
raw_input ("\nPRESSIONAR <Enter> PARA TERMINAR.");
# fim do programa
"""

# ----- EXEMPLO108

PI = 3.14;        # definicao de constante
                  # com nome e tipo (melhor)
X = 10.01;        # definicao de variavel real
                  # com atribuicao de valor inicial

print "EXEMPLO108 - PRIMEIRO EXEMPLO EM PYTHON";
print "\nMATRICULA: _____ ALUNO : _____ ";
print "\nEXEMPLOS DE VALORES : ";
print "\nCARACTERE : " , 'A';          # letra ou simbolo
print "\nINTEIRO      : " , 10;         # valor sem parte fracionaria
print "\nREAL         : " , PI;         # constante real
print "\nREAL         : " , X;         # variavel real

# para esperar
raw_input ("\nPRESSIONAR <Enter> PARA TERMINAR.");
# fim do programa
"""

```

```

"""
# ----- EXEMPLO109

PI = 3.14;          # definicao de constante
                    # com nome e tipo (melhor)
                    # definicao de variavel real
X = 10.01;

                    # definicao de variavel inteira
I = 10;

print "EXEMPLO109 - PRIMEIRO EXEMPLO EM PYTHON";
print "\nMATRICULA: _____ ALUNO : _____ ";
print "\nEXEMPLOS DE VALORES : ";
print "\nINTEIRO : " , I ;
print "\nREAL   : " , X ;
print "\nREAL   : " , PI;

# para esperar
raw_input ("\nPRESSIONAR <Enter> PARA TERMINAR.");
# fim do programa
"""

# ----- EXEMPLO110

PI = 3.14;          # definicao de constante
                    # com nome e tipo (melhor)
                    # definicao de variavel real
X = 10.01;

                    # definicao de variavel inteira
I = 10;

                    # definicao de variavel caractere
N = '\n';          # mudar de linha

print "EXEMPLO110 - PRIMEIRO EXEMPLO EM PYTHON";
print "\nMATRICULA: _____ ALUNO : _____ ";
print N, "EXEMPLOS DE VALORES : ";
print N, "INTEIRO : " , I ;
print N, "REAL   : " , X ;
print N, "REAL   : " , PI;

# para esperar
raw_input ("\nPRESSIONAR <Enter> PARA TERMINAR.");
# fim do programa
"""

```

```

#
# OBS.: RETIRAR OS COMENTARIOS "" ""
# PARA TESTAR CADA EXEMPLO INDIVIDUALMENTE.
#

""
# ----- EXEMPLO201

# PROGRAMA PARA LER E IMPRIMIR UM VALOR INTEIRO
# VARIABEL:
X = 0;

print "EXEMPLO201 - LER E IMPRIMIR UM VALOR INTEIRO";
print "\nFORNECER UM VALOR INTEIRO QUALQUER: ";
X = input ( "X = " );
print "\nO VALOR DIGITADO FOI: ", X;
print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );      # para esperar
# fim do programa
""

""
# ----- EXEMPLO202

# PROGRAMA PARA LER E IMPRIMIR UM VALOR REAL
# VARIABEL:
X = 0.0;

print "EXEMPLO202 - LER E IMPRIMIR UM VALOR REAL";
print "\nFORNECER UM VALOR REAL QUALQUER: ";
input X;
print "\nO VALOR DIGITADO FOI: ", X;
print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );      # para esperar
# fim do programa
""

""
# ----- EXEMPLO203

# PROGRAMA PARA LER E IMPRIMIR UM CARACTERE
# VARIABEL:
X = 'A';

print "EXEMPLO203 - LER E IMPRIMIR UM CARACTERE";
print "\nFORNECER UM CARACTERE QUALQUER: ";
X = input ( "X = " );      # X = 'A'
print "\nO VALOR DIGITADO FOI: ", X;
print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );      # para esperar
# fim do programa
""

```

```

""
# ----- EXEMPLO204

# PROGRAMA PARA LER E IMPRIMIR, NO MAXIMO, 10 CARACTERES
# VARIÁVEL:
X = "";

print "EXEMPLO204 - LER E IMPRIMIR, NO MAXIMO, 09 CARACTERES";
print "\nDIGITE, NO MAXIMO, 09 CARACTERES QUAISQUER: ";
X = input ( "X = " );      # X = "ABC"
print "\nFOI DIGITADO: ", X;
print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );             # para esperar
# fim do programa
""

# ----- EXEMPLO205

# PROGRAMA PARA LER E SOMAR DOIS VALORES INTEIROS
# VARIÁVEIS:
X = Y = Z = 0;

print "EXEMPLO205 - LER E SOMAR DOIS VALORES INTEIROS";
print "\nFORNECER UM VALOR INTEIRO QUALQUER: ";
X = input ( " X = " );
print "\nFORNECER OUTRO VALOR INTEIRO QUALQUER: ";
Y = input ( " Y = " );
Z = X + Y;
print "\nA SOMA DOS DOIS = ", Z;
print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );             # para esperar
# fim do programa
""

# ----- EXEMPLO206

# PROGRAMA PARA LER E SUBTRAIR DOIS VALORES REAIS
# VARIÁVEIS:
X = Y = Z = 0.0;

print "EXEMPLO206 - LER E SUBTRAIR DOIS VALORES REAIS";
print "\nFORNECER UM VALOR REAL QUALQUER: ";
X = input ( " X = " );
print "\nFORNECER OUTRO VALOR REAL QUALQUER: ";
Y = input ( " Y = " );
Z = X - Y;
print "\nA DIFERENÇA ENTRE OS DOIS = ", Z;
print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );             # para esperar
# fim do programa
""

```

```

""
# ----- EXEMPLO207

# PROGRAMA PARA OPERAR VALORES LOGICOS
# VARIAVEIS:
X = Y = Z = False;

print "EXEMPLO207 - OPERAR VALORES LOGICOS";
X = True;
Y = False;
Z = X or Y;          # X ou Y
print "\nA DISJUNCAO ENTRE VERDADEIRO E FALSO = ", Z;
Z = X and Y;         # X e Y
print "\nA CONJUNCAO ENTRE VERDADEIRO E FALSO = ", Z;
print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );       # para esperar
# fim do programa
""

# ----- EXEMPLO208

# PROGRAMA PARA CALCULAR A VELOCIDADE DE UM VEICULO
# VARIAVEIS:
D = 0.0;             # Distancia
T = 0.0;             # Tempo
V = 0.0;             # Velocidade

print "EXEMPLO208 - CALCULAR A VELOCIDADE DE UM VEICULO";
print "\nFORNECER UMA DISTANCIA QUALQUER EM METROS: ";
D = input ( "D = " );
print "\nFORNECER O TEMPO PARA PERCORRE-LA EM SEGUNDOS: ";
T = input ( "T = " );
V = D / T;
print "\nV = D / T = ", V, " m/s ";
print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );       # para esperar
# fim do programa
""

# ----- EXEMPLO209

# PROGRAMA PARA COMPARAR CARACTERES COM UMA SENHA
# CONSTANTE:
SENHA = "XXXX";
# VARIABEL:
S = "";

print "EXEMPLO209 - COMPARAR CARACTERES COM UMA SENHA";
print "\nFORNECER UMA CADEIA DE CARACTERES QUALQUER: ";
S = input ( "S = " ); # S = "XXXX"
print "\nA COMPARACAO COM A SENHA = ", (1 if (S==SENHA) else 0);
print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );       # para esperar
# fim do programa
""

```

```

"""
# ----- EXEMPLO210a

# PROGRAMA PARA CALCULAR O ARCO TRIGONOMETRICO DE UM SENO
from math import *

# CONSTANTE:
PI = 3.14;
# VARIAVEIS:
ARCO = 0.0;
COSSENO = 0.0;
SENO = 0.0;
TANGENTE = 0.0;

print "EXEMPLO210 - CALCULAR O ARCO TRIGONOMETRICO DE UM SENO";
print "\nFORNECER O VALOR DO SENO: ";
SENO = input ( "SENO = " );
COSSENO = sqrt( 1.0 - pow(SENO,2));
TANGENTE = SENO / COSSENO;
ARCO = atan(TANGENTE);
print "\nO ARCO TRIGONOMETRICO EM GRAUS = ", (ARCO*180.0/PI);
print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );          # para esperar
# fim do programa
"""

# ----- EXEMPLO210b

# PROGRAMA PARA CALCULAR O ARCO TRIGONOMETRICO DE UM SENO
import math

# CONSTANTE:
PI = 3.14;
# VARIAVEIS:
ARCO = 0.0;
COSSENO = 0.0;
SENO = 0.0;
TANGENTE = 0.0;

print "EXEMPLO210 - CALCULAR O ARCO TRIGONOMETRICO DE UM SENO";
print "\nFORNECER O VALOR DO SENO: ";
SENO = input ( "SENO = " );
COSSENO = math.sqrt( 1.0 - math.pow(SENO,2));
TANGENTE = SENO / COSSENO;
ARCO = math.atan(TANGENTE);
print "\nO ARCO TRIGONOMETRICO EM GRAUS = ", (ARCO*180.0/PI);
print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );          # para esperar
# fim do programa
"""

```

```

#
# OBS.: RETIRAR OS COMENTARIOS "" ""
# PARA TESTAR CADA EXEMPLO INDIVIDUALMENTE.
#

""
# ----- EXEMPLO301

# PROGRAMA PARA LER UM VALOR INTEIRO E VERIFICAR SE E' ZERO
# VARIABEL:
X = 0;

print "EXEMPLO301 - LER E TESTAR UM VALOR INTEIRO";
print "\nFORNECER UM VALOR INTEIRO QUALQUER: ";
X = input ( "X = " );
if ( X == 0 ):
    print "\nO VALOR DIGITADO FOI ZERO";
else:
    print "\nO VALOR DIGITADO NAO FOI ZERO";
print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );      # para esperar
# fim do programa
""

# ----- EXEMPLO302

# PROGRAMA PARA LER UM REAL E TESTAR SE DIFERENTE DE ZERO
# VARIABEL:
X = 0.0;

print "EXEMPLO302 - LER E TESTAR UM VALOR REAL";
print "\nFORNECER UM VALOR REAL DIFERENTE DE ZERO: ";
X = input ( "X = " );
if ( X != 0.0 ):
    print "\nO VALOR DIGITADO FOI DIFERENTE DE ZERO";
print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );      # para esperar
# fim do programa
""

```



```

""
# ----- EXEMPLO303

# PROGRAMA PARA LER CARACTERE E VERIFICAR SE E' UM ALGARISMO
# VARIÁVEL:
X = 'A';

print "EXEMPLO303 - LER E TESTAR UM CARACTERE";
print "\nFORNECER UM ALGARISMO QUALQUER: ";
X = input ( "X = " );
if ( X >= '0' and X <= '9' ):

    print "\nO VALOR DIGITADO FOI UM ALGARISMO";
    print "\nO ALGARISMO DIGITADO FOI: " , X ;
    # if ALGARISMO
print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );          # para esperar
# fim do programa
""
""
# ----- EXEMPLO304

# PROGRAMA PARA LER CARACTERE E TESTAR SE NAO E' ALGARISMO
# VARIÁVEL:
X = 'A';

print "EXEMPLO304 - LER E TESTAR CARACTERE";
print "\nFORNECER UM CARACTERE QUALQUER: ";
X = input ( "X = " );
if ( not ( X >= '0' and X <= '9' ) ):

    print "\nNAO FOI DIGITADO UM ALGARISMO";
    print "\nFOI DIGITADO O CARACTERE: " , X;
    # if NAO ALGARISMO
print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );          # para esperar
# fim do programa
""
""
# ----- EXEMPLO305

# PROGRAMA PARA LER E TESTAR A IGUALDADE DE DOIS INTEIROS
# VARIÁVEIS:
X = 0; Y = 0;

print "EXEMPLO305 - LER E TESTAR DOIS VALORES INTEIROS";
print "\nFORNECER UM VALOR INTEIRO QUALQUER: ";
X = input ( "X = " );
print "\nFORNECER OUTRO VALOR INTEIRO QUALQUER: ";
Y = input ( "Y = " );
if ( X == Y ):
    print "\nDOIS VALORES IGUAIS";
else:
    print "\n", X, " DIFERENTE DE ", Y;

print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );          # para esperar
# fim do programa
""

```

```

""
# ----- EXEMPLO306

# PROGRAMA PARA LER E TESTAR DOIS VALORES REAIS
# VARIÁVEIS:
X = 0.0; Y = 0.0;

print "EXEMPLO306 - LER E TESTAR DOIS VALORES REAIS";

print "\nFORNECER UM VALOR REAL QUALQUER: ";
X = input ( "X = " );
print "\nFORNECER OUTRO VALOR REAL QUALQUER: ";
Y = input ( "Y = " );
if ( not (X == Y) ):
    print "\n", X, " DIFERENTE DE ", Y;
else:
    print "VALORES IGUAIS";
# if VALORES DIFERENTES
print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );          # para esperar
# fim do programa
""

""
# ----- EXEMPLO307

# PROGRAMA PARA TRATAR ALTERNATIVAS COM VALORES LÓGICOS
# VARIÁVEIS:
X = 0; Y = 0;
Z = False;

print "EXEMPLO307 - TRATAR VALORES LÓGICOS";
print "\nFORNECER UM VALOR INTEIRO QUALQUER: ";
X = input ( "X = " );
print "\nFORNECER OUTRO VALOR INTEIRO QUALQUER: ";
Y = input ( "Y = " );
Z = (X == Y);
if ( Z ):
    print "VALORES IGUAIS";
else:
    print "VALORES DIFERENTES";
print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );          # para esperar
# fim do programa
""

```

```

""
# ----- EXEMPLO308

# PROGRAMA PARA LER E TESTAR UMA LETRA
# VARIÁVEL:
X = '0';

print "EXEMPLO308 - LER E TESTAR UMA LETRA";
print "\nFORNECER UMA LETRA QUALQUER: ";
X = input ( "X = " );
if ( X >= 'A' and X <= 'Z' ):
    print "FOI DIGITADA UMA LETRA MAIUSCULA";
else:
    if( X >= 'a' and X <= 'z' ):
        print "FOI DIGITADA UMA LETRA MINUSCULA";
    else:
        print "NAO FOI DIGITADA UMA LETRA";
print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );          # para esperar
# fim do programa
""

# ----- EXEMPLO309

# PROGRAMA PARA COMPARAR CARACTERES < , = , >
# CONSTANTES:
MAIOR = '>',
IGUAL = '==',
MENOR = '<';
# VARIÁVEL:
X = '0';

print "EXEMPLO309 - COMPARAR CARACTERES < , = , >";
print "\nFORNECER UM DOS CARACTERES CITADOS: ";
X = input ( "X = " );   # X = ('>,)
if ( X == MAIOR ):
    print "FOI DIGITADO O SINAL DE MAIOR";
elif ( X == IGUAL ):
    print "FOI DIGITADO O SINAL DE IGUAL";
elif ( X == MENOR ):
    print "FOI DIGITADO O SINAL DE MENOR";
else:
    print "FOI DIGITADO UM OUTRO CARACTERE QUALQUER";
# COMPARACAO DE X COM < , = , >
print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );          # para esperar
# fim do programa
""

```

```

""
# ----- EXEMPLO310

# PROGRAMA PARA IDENTIFICAR CARACTERES
# VARIÁVEL
X = '0';

print "EXEMPLO310 - IDENTIFICAR CARACTERES";
print "\nFORNECER UM CARACTERE QUALQUER: ";
X = raw_input ( "X = " );
if ( X == 'A' or X == 'E' or X == 'I' or X == 'O' or X == 'U' ):
    print "FOI DIGITADO UMA VOGAL";
elif ( X == '0' or X == '1' or X == '2' or X == '3' or X == '4' or
        X == '5' or X == '6' or X == '7' or X == '8' or X == '9' ):
    print "FOI DIGITADO UM ALGARISMO";
    print "\nO NÚMERO CORRESPONDENTE = ", (X-48);
else:
    default: print "FOI DIGITADO UM OUTRO CARACTERE QUALQUER";
# IDENTIFICAÇÃO DE UM CARACTERE
print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );          # para esperar
# fim do programa
""

```

```

#
# OBS.: RETIRAR OS COMENTARIOS "" ""
#PARA TESTAR CADA EXEMPLO INDIVIDUALMENTE.
#

""
# ----- EXEMPLO401

# PROGRAMA PARA LER E IMPRIMIR 03 VALORES INTEIROS
# VARIAVEIS :
X = 0;
CONTADOR = 0;

print "EXEMPLO401 - LER E IMPRIMIR 03 VALORES INTEIROS";
print                                # mudar de linha
CONTADOR = 1;
while ( CONTADOR <= 3 ):      # REPETIR
    print                        # mudar de linha
    print CONTADOR , " FORNECER UM VALOR INTEIRO : ";
    X = input ( "X = " );
    print "\nO VALOR DIGITADO FOI : " , X , "\n";
    CONTADOR = CONTADOR + 1;
                                # ENQUANTO CONTADOR <= 3

print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );                # para esperar
# fim do programa
""

# ----- EXEMPLO402

# PROGRAMA PARA LER E IMPRIMIR (N) VALORES INTEIROS
# VARIAVEIS :
X = N = CONTADOR = 0;

print "EXEMPLO402 - LER E IMPRIMIR (N) VALORES INTEIROS\n";
print "\nFORNECER O NUMERO DE VEZES (N) : ";
N = input ( "N = " );
CONTADOR = 1;
while ( CONTADOR <= N ):
    print "\n" , CONTADOR;
    print " FORNECER UM VALOR INTEIRO QUALQUER : ";
    X = input ( "X = " );
    print "\nO VALOR DIGITADO FOI : " , X , "\n";
    CONTADOR = CONTADOR + 1;
# ENQUANTO CONTADOR <= N

print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );                # para esperar
# fim do programa
""

```

```

""
# ----- EXEMPLO403

# PROGRAMA PARA LER E IMPRIMIR (N) VALORES INTEIROS
# VARIÁVEIS :
X = N = 0;

print "EXEMPLO403 - LER E IMPRIMIR (N) VALORES INTEIROS\n";
print "\nFORNECER O NUMERO DE VEZES (N) : ";
N = input ( "N = " );
while ( N > 0 ):          # REPETIR
    print "\n", N;
    print " FORNECER UM VALOR INTEIRO QUALQUER : ";
    X = input ( "X = " );
    print "\nO VALOR DIGITADO FOI : ", X , "\n";
    N = N - 1;
    # ENQUANTO N > 0
print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );          # para esperar
# fim do programa
""

```

```

""
# ----- EXEMPLO404

# PROGRAMA PARA LER E IMPRIMIR 03 VALORES INTEIROS
# VARIÁVEIS :
X = CONTADOR = 0;

print "EXEMPLO404 - LER E IMPRIMIR 03 VALORES INTEIROS\n";
for CONTADOR in xrange ( 1, 4, 1 ):
    print "\n", CONTADOR , " FORNECER UM VALOR INTEIRO : ";
    X = input ( "X = " );
    print "\nO VALOR DIGITADO FOI : ", X , "\n";
    # PARA CONTADOR EM [1:3]

print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );          # para esperar
# fim do programa
""

```

```

""
# ----- EXEMPLO405

# PROGRAMA PARA LER E IMPRIMIR (N) VALORES INTEIROS
# VARIÁVEIS :
X = N = CONTADOR = 0;

print "EXEMPLO405 - LER E IMPRIMIR (N) VALORES INTEIROS\n";
print "\nFORNECER O NUMERO DE VEZES (N) : ";
N = input ( "N = " );
for CONTADOR in xrange ( 1, N+1, 1 ):
    print "\n", CONTADOR, " FORNECER UM VALOR INTEIRO : ";
    X = input ( "X = " );
    print "\nO VALOR DIGITADO FOI : ", X, "\n";
# PARA CONTADOR EM [1:N]

print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );          # para esperar
# fim do programa
""

```

```

""
# ----- EXEMPLO406

# PROGRAMA PARA LER E IMPRIMIR 03 VALORES INTEIROS
# VARIÁVEIS :
X = CONTADOR = 0;

print "EXEMPLO406 - LER E IMPRIMIR 03 VALORES INTEIROS\n";
CONTADOR = 1;
condition = True;
while ( condition ):      # REPETIR
    print "\n", CONTADOR, " DIGITE UM VALOR INTEIRO : ";
    X = input ( "X = " );
    print "\nO VALOR DIGITADO FOI : ", X, "\n";
    CONTADOR = CONTADOR + 1;
    condition = ( CONTADOR <= 3 );
# ATE' ( CONTADOR > 3 )

print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );          # para esperar
# fim do programa
""

```

```

""

```

```
# ----- EXEMPLO407

# PROGRAMA PARA LER E IMPRIMIR (N) VALORES INTEIROS
# VARIÁVEIS :
X = CONTADOR = 0;

print "EXEMPLO407 - LER E IMPRIMIR (N) VALORES INTEIROS\n";
print "\nFORNECER O NUMERO DE VEZES (N) : ";
CONTADOR = input ( "CONTADOR = " );
condition = True;
while ( condition ):      # REPETIR
    print "\n", CONTADOR, " FORNECER UM VALOR INTEIRO : ";
    X = input ( "X = " );
    print "\nO VALOR DIGITADO FOI : ", X, "\n";
    CONTADOR = CONTADOR + 1;
    condition = ( CONTADOR > 0 );
                    # ATE' ( CONTADOR <= 3 )

print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );      # para esperar
# fim do programa
""

""
# ----- EXEMPLO408

# PROGRAMA PARA LER E IMPRIMIR INTEIROS DIFERENTES DE ZERO
# VARIÁVEL :
X = 0;

print "EXEMPLO408 - LER E IMPRIMIR INTEIROS NAO NULOS\n";
print "\nFORNECER UM VALOR INTEIRO (0 = PARAR) : ";
X = input ( "X = " );
while ( X != 0 ):      # REPETIR
    print "\nO VALOR DIGITADO FOI : ", X, "\n";
    print "\nDIGITE UM VALOR INTEIRO QUALQUER : ";
    X = input ( "X = " );
                    # ENQUANTO X DIFERENTE DE ZERO

print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );      # para esperar
# fim do programa
""
```



```

""
# ----- EXEMPLO409

# PROGRAMA PARA LER UM INTEIRO DIFERENTE DE ZERO
# VARIÁVEL :
X = 0;

print "EXEMPLO409 - PARA LER UM INTEIRO NAO NULO\n";
print "\nFORNECER UM VALOR DIFERENTE DE ZERO : ";
X = input ( "X = " );
while ( X == 0 ):          # REPETIR
    print "\nFORNECER UM VALOR DIFERENTE DE ZERO : ";
    X = input ( "X = " );
    # ENQUANTO X IGUAL A ZERO
print "\nDIGITADO UM NUMERO DIFERENTE DE ZERO\n";

print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );          # para esperar
# fim do programa
""

```

```

""
# ----- EXEMPLO410

# PROGRAMA PARA LER UM INTEIRO DIFERENTE DE ZERO
# VARIÁVEL :
X = 0;

print "EXEMPLO410 - LER UM INTEIRO NAO NULO\n";
condition = True;
while ( condition ):      # REPETIR
    print "\nFORNECER UM VALOR DIFERENTE DE ZERO : ";
    X = input ( "X = " );
    condition = ( X == 0 ); # ATE' X DIFERENTE DE ZERO
print "\nDIGITADO UM NUMERO DIFERENTE DE ZERO\n";

print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );          # para esperar
# fim do programa
""

```

```

#
# OBS.: RETIRAR OS COMENTARIOS "" ""
#   PARA TESTAR CADA EXEMPLO INDIVIDUALMENTE.
#

""
# ----- EXEMPLO501

def P1( ):

    print;
    print "\nCHAMADO O PROCEDIMENTO P1 SEM PARAMETROS";
    print;
    return;
# fim procedimento P1( )

# PROGRAMA PARA MOSTRAR PASSAGENS DE PARAMETROS

print "EXEMPLO501 - CHAMADA A UM PROCEDIMENTO";
P1 ( );          # chamada ao procedimento
print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );   # para esperar
# fim do programa
""

""
# ----- EXEMPLO502

global X; # VARIABEL GLOBAL

def P1 ( ):
    global X;
    print;
    print "CHAMADO O PROCEDIMENTO P1 ", X , " VEZES";
    print;
# fim procedimento P1( )

# PROGRAMA PARA MOSTRAR PASSAGENS DE PARAMETROS

X = 0;
print "EXEMPLO502 - CHAMADA COM VARIABEL GLOBAL\n";
for X in range ( 5 ):
    P1 ( );      # chamar 5 vezes
print;

print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );   # para esperar
# fim do programa
""

```

```

"""
# ----- EXEMPLO503

global X; # VARIAVEL GLOBAL

def P1 ( ):
    global X;
    X = X + 1;
    print "\nCHAMADO O PROCEDIMENTO P1 ", X , " VEZES";
    if( X < 5 ):
        P1 ( );          # chamar recursivamente
    print "\nRETORNANDO AO PROCEDIMENTO P1 PARA A CHAMADA ", X;
    X = X - 1;
    raw_input ( );       # para esperar
# fim procedimento P1( )

# PROGRAMA PARA MOSTRAR PASSAGENS DE PARAMETROS

print "EXEMPLO0503 - CHAMADA/RETORNO COM VARIAVEL GLOBAL\n";
X = 0;
P1 ( ); # OBSERVAR A RECURSIVIDADE !
print;

print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );          # para esperar
# fim do programa
"""

# ----- EXEMPLO504

def P1 ( X ):
    print "\nCHAMADO O PROCEDIMENTO P1 ", X , " VEZES";
    if ( X < 5 ):
        P1 ( X + 1 );    # chamar recursivamente com parametro
    print "\nRETORNANDO AO PROCEDIMENTO P1 PARA A CHAMADA ", X;
    raw_input ( );       # para esperar
# fim procedimento P1( )

# PROGRAMA PARA MOSTRAR PASSAGENS DE PARAMETROS

print "EXEMPLO0904 - CHAMADA/RETORNO COM PARAMETRO\n";
P1 ( 1 ); # OBSERVAR REPETICAO FINITA, SEM VARIAVEL GLOBAL !
print"\n";

print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );          # para esperar
# fim do programa
"""

```

```

""
# ----- EXEMPLO505

def P1 ( X ):
    print "\nCHAMADO O PROCEDIMENTO P1 ", X , " VEZES";
    if ( X > 1 ):
        P1 ( X - 1 );
    print "\nRETORNANDO AO PROCEDIMENTO P1 PARA A CHAMADA " , X;
# fim procedimento P1( )

# PROGRAMA PARA MOSTRAR PASSAGEM DE PARAMETRO POR VALOR

print "EXEMPLO0505 - CHAMADA/RETORNO COM PARAMETRO\n";
P1 ( 5 ); # OBSERVAR REPETICAO FINITA, SEM VARIAVEL GLOBAL !
print;

print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );          # para esperar
# fim do programa
""
""
# ----- EXEMPLO506

def P1 ( X ):
    print "\nCHAMADO O PROCEDIMENTO P1 COM X = " , X ;
    if ( X < 5 ):
        P2 ( X );
    print "\nRETORNANDO AO PROCEDIMENTO P1 PARA A CHAMADA " , X;
    raw_input ( );          # para esperar
# fim do procedimento P1( )

def P2 ( X ):
    print "\nCHAMADO O PROCEDIMENTO P2 COM X = " , X ;
    X = X+1;
    print "\nRETORNANDO AO PROCEDIMENTO P1 PARA A CHAMADA " , X;
    raw_input ( );          # para esperar
    P1 ( X );
# fim do procedimento P2( )

# PROGRAMA PARA MOSTRAR PASSAGENS DE PARAMETROS

print "EXEMPLO0506 - CHAMADA/RETORNO COM PARAMETRO\n";
P1 (1); # OBSERVAR RECURSIVIDADE INDIRETA !
print"\n";

print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );          # para esperar
# fim do programa
""

```

```

"""
# ----- EXEMPLO507

def P1 ( X ):
    X[0] = X[0] + 1;          # desempacotamento do valor
    print "\nCHAMADO O PROCEDIMENTO P1 ", X , " VEZES";
    if ( X[0] < 5 ):
        P1( X );
    print "\nRETORNANDO AO PROCEDIMENTO P1 PARA A CHAMADA " , X;
    X[0] = X[0] - 1;
    raw_input ( );           # para esperar
# fim procedimento P1( )

# PROGRAMA PARA MOSTRAR PASSAGEM DE PARAMETRO POR REFERENCIA
# VARIÁVEL LOCAL
print "EXEMPLO507 - CHAMADA/RETORNO COM REFERENCIA\n";
X = [0];                    # empacotamento do valor
P1 ( X );
print;

print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );              # para esperar
# fim do programa
"""

```

```

"""
# ----- EXEMPLO508

def P1 ( X ):
    X = X + 1;
    print "\nCHAMADO O PROCEDIMENTO P1 ", X , " VEZES";
    if ( X < 3 ):
        P1( X );
        P2( X );

    print "\n";
    print "\nRETORNANDO AO PROCEDIMENTO P1 PARA A CHAMADA " , X;
    raw_input ( );          # para esperar
# fim do procedimento P1( )

def P2 ( X ):
    print "\nCHAMADO O PROCEDIMENTO P2 ", X , " VEZES";
    if ( X > 1 ):
        P2( X - 1 );
    print "\nRETORNANDO AO PROCEDIMENTO P2 PARA A CHAMADA " , X;
    raw_input ( );          # para esperar
# fim procedimento P2( )

# PROGRAMA PARA MOSTRAR PASSAGENS DE PARAMETROS

print "EXEMPLO508 - MULTIPLAS CHAMADAS/RETORNOS\n";
P1 ( 0 );
print;

print "\nPRESSIONAR <Enter> PARA TERMINAR.";
raw_input ( );              # para esperar
# fim do programa
"""

```

““

- Digitar o conteudo abaixo em um arquivo com o nome LIB01.PY :

----- EXEMPLO509a

```
def LIMPAR_TELA ( ):
    import os;
    os.system ( "cls" );
```

```
def ESPERAR ( ):
    raw_input ( );
```

- Digitar o conteudo abaixo em outro arquivo com o nome LIB02.PY :

----- EXEMPLO509b

```
def ESCREVER_CARACTERES ( S ):
    print S;
```

- Digitar o conteudo abaixo em um terceiro arquivo :

““

----- EXEMPLO509

PROGRAMA PARA MOSTRAR O USO DE MODULOS

```
mport lib01
import lib02
```

```
lib01.LIMPAR_TELA ( )
```

```
lib02.ESCREVER_CARACTERES ( "EXEMPLO0509 - MODULOS\n" );
```

```
print "\nPRESSIONAR <Enter> PARA TERMINAR.";
```

```
lib01.ESPERAR ( )
```

““

- Digitar o conteudo abaixo em um arquivo com o nome LIB03.PY :

CONSTANTES GLOBAIS :

```
MINX = 1;    # valor minimo
MAXX = 3;    # valor maximo
```

PROCEDIMENTOS E FUNCOES :

```
def PARAR ( X ):          # condicao de parada igual a '.'
    return ( X <= 0 );
```

```
def P1 ( X ):
    print "\nCHAMADO O PROCEDIMENTO P1 ", X , " VEZES";
```

```
    if ( not PARAR ( X ) ):
        P1 ( X - 1 );
```

```
        P1 ( X - 1 );
```

```
    print "\nRETORNANDO AO PROCEDIMENTO P1 PARA A CHAMADA ", X;
```

```
    raw_input ( );          # para esperar
```

```
# fim do procedimento P1 ( )
```

Digitar o conteúdo abaixo em outro arquivo :

```
""
# ----- EXEMPLO510

# PROGRAMA PARA MOSTRAR O USO DE MODULOS
import lib01
import lib02
import lib03

lib01.LIMPAR_TELA ( )
lib02.ESCREVER_CARACTERES ( "EXEMPLO0510 - MODULOS\n" );

lib03.P1 ( lib03.MAXX );

print "\nPRESSIONAR <Enter> PARA TERMINAR.";
lib01.ESPERAR ( )
""
```

```

#
# OBS.: RETIRAR OS COMENTARIOS "" ""
# PARA TESTAR CADA EXEMPLO INDIVIDUALMENTE.
#

""
# ----- EXEMPLO601

def CONTAR ( X ):

    if ( X > 0 ):
        CONTAR ( X-1 );
        print "\n", X;

    # fim procedimento CONTAR ( )

# PROGRAMA PARA CONTAR DE 1 ATE' 5, RECURSIVAMENTE

print "EXEMPLO601 - CONTAR DE 1 A 5 RECURSIVAMENTE\n";
CONTAR ( 5 );
print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );          # para esperar
# fim do programa
""

""
# ----- EXEMPLO602

def CONTAR ( X ):

    if ( X > 0 ):
        print "\n", X;
        CONTAR ( X-1 );

    # fim procedimento CONTAR ( )

# PROGRAMA PARA CONTAR 5 10 ATE' 1, RECURSIVAMENTE

print "EXEMPLO602 - CONTAR DE 5 A 1 RECURSIVAMENTE\n";
CONTAR ( 5 );
print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );          # para esperar
# fim do programa
""

```



```

""
# ----- EXEMPLO603

def PARES ( X ):

    if ( X > 0 ):
        if ( X % 2 == 0 ):
            PARES ( X-2 );
            print "\n", X;
        else:
            PARES ( X-1 );

    # fim procedimento PARES ( )

# PROGRAMA RECURSIVO PARA MOSTRAR PARES

print "EXEMPLO603 - MOSTRAR OS PARES <= 10\n";
PARES ( 10 );
print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );          # para esperar
# fim do programa
""

""
# ----- EXEMPLO604

def PARES ( X ):

    if ( X > 0 ):
        PARES ( X-1 );
        print X, " ", 2*X;

    # fim procedimento PARES ( )

# PROGRAMA RECURSIVO PARA MOSTRAR PARES

print "EXEMPLO604 - MOSTRAR OS 5 PRIMEIROS PARES\n";
PARES ( 5 );
print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );          # para esperar
# fim do programa
""

```

```

""
# ----- EXEMPLO605

def PARES ( X, S ):

    if ( X > 0 ):
        PARES ( X-1, S ); S[0] = S[0] + 2*X;
    else:
        S[0] = 0;
    # fim procedimento PARES ( )

# PROGRAMA RECURSIVO PARA SOMAR PARES
# DADO:
SOMA = [0];

print "EXEMPLO605 - SOMAR OS 5 PRIMEIROS PARES\n";
PARES ( 5, SOMA );
print "\nSOMA DOS 5 PRIMEIROS PARES = " + str (SOMA);
print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );          # para esperar
# fim do programa
""

```

```

""
# ----- EXEMPLO606

def PARES ( X ):

    S = 0;
    if ( X > 0 ):
        S = 2*X + PARES ( X-1 );
    else:
        S = 0;
    return ( S );
    # fim funcao PARES ( )

# PROGRAMA RECURSIVO PARA SOMAR PARES
# DADO :
SOMA = 0;

print "EXEMPLO606 - SOMAR OS 5 PRIMEIROS PARES\n";
SOMA = PARES ( 5 );
print "\nSOMA DOS 5 PRIMEIROS PARES = ", SOMA;
print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );          # para esperar
# fim do programa
""

```

```

"""
# ----- EXEMPLO607

def PARES ( X ):

    S = 0;

    if ( X > 0 ):
        if ( X % 2 == 0 ):
            S = 1 + PARES ( X-2 );
        else:
            S = PARES ( X-1 );
    else:
        S = 0;

    return S;
# fim funcao PARES ( )

# PROGRAMA RECURSIVO PARA CONTAR PARES

print "EXEMPLO607 - CONTAR OS PARES <= 10\n";
print "\nPARES <= 10 = ", PARES(10);
print "\nFORNECER <Enter> PARA TERMINAR";
raw_input ( );          # para esperar
# fim do programa
"""

"""
# ----- EXEMPLO608

def PROCURAR ( LETRA, S, POSICAO):

    R = False;

    if ( POSICAO < len(S) ):
        R = (S[POSICAO] == LETRA) or PROCURAR(LETRA,S,POSICAO+1);
    else:
        R = false;

    return ( R );
# fim funcao PROCURAR ( )

# PROGRAMA RECURSIVO PARA ACHAR A POSICAO DE UMA LETRA
L = '0';
S = "";

print "EXEMPLO608 - PROCURAR UMA LETRA EM UMA SENTENCA\n";
print "\nFORNECER UMA SENTENCA COM MENOS DE 30 CARACTERES : ";
S = input ( );          # "abc"
print "\nFORNECER UMA LETRA PARA SER PROCURADA : ";
L = raw_input ( );      # a
print "\nRESPOSTA = " + str ( PROCURAR ( L,S,0 ));
print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );          # para esperar
# fim do programa
"""

```

```

""
# ----- EXEMPLO609

def PROCURAR ( LETRA, S, POSICAO ):

    R = -1;    # nao encontrado

    if ( POSICAO < len(S) ):
        if ( S[POSICAO] == LETRA ):
            R = POSICAO + 1; # a primeira posicao e' 0 !
        else:
            R = PROCURAR ( LETRA, S, POSICAO+1 );
    else:
        R = -1; # nao encontrado

    return ( R );
# fim funcao PROCURAR ( )

# PROGRAMA RECURSIVO PARA PROCURAR UMA LETRA
L = '0';
S = "";

print "EXEMPLO609 - POSICAO DE UMA LETRA EM UMA SENTENCA\n";
print "\nFORNECER UMA SENTENCA COM MENOS DE 30 CARACTERES : ";
S = input ( );
print "\nFORNECER UMA LETRA PARA SER PROCURADA : ";
L = raw_input ( );
print "\nRESPOSTA = " + str ( PROCURAR ( L, S, 0 ));
print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );    # para esperar
# fim do programa
""

```

```

"""
# ----- EXEMPLO610

def PROCURAR ( LETRA, S, POSICAO ):

    if ( POSICAO < len(S) ):
        if ( S[POSICAO] == LETRA ):
            R = 1 + PROCURAR ( LETRA, S,POSICAO+1 );
        else:
            R = PROCURAR ( LETRA, S, POSICAO+1 );
    else:
        R = 0; # nao encontrado

    return ( R );
# fim funcao PROCURAR ( )

# PROGRAMA RECURSIVO PARA PROCURAR OCORRENCIAS DE UMA LETRA
L = '0';
S = "";

print "EXEMPLO610 - PROCURAR OCORRENCIAS DE UMA LETRA\n";
print "\nFORNECER UMA SENTENCA COM MENOS DE 30 CARACTERES : ";
S = input ( );
print "\nFORNECER UMA LETRA PARA SER PROCURADA : ";
L = raw_input ( );
print "\nRESPOSTA = " + str ( PROCURAR ( L, S, 0 ));
print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );          # para esperar
# fim do programa
"""

```

```

#
# OBS.: RETIRAR OS COMENTARIOS "" ""
#      PARA TESTAR CADA EXEMPLO INDIVIDUALMENTE.
#

""
# ----- EXEMPLO701

# PROGRAMA PARA LER UMA TABELA DE ITENS
V = [0] * 10;
X = 0;

print "EXEMPLO701 - LER UM TABELA DE 10 ITENS\n";
      # REPETIR PARA CADA POSICAO
for X in range (10):
      # a primeira posicao e' zero !
      print "\nFORNECER O ", (X+1), "o. : ";
      V [ X ] = input ( );
      # FIM REPETIR
print "\nVETOR LIDO: \n";
      # REPETIR PARA CADA POSICAO
for X in range (10):

      print V [ X ], " ",

      # FIM REPETIR

print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );      # para esperar
# fim do programa
""

""
# ----- EXEMPLO702

# PROGRAMA PARA SOMAR UMA TABELA DE ITENS
V = [0] * 10;
X = 0;
SOMA = 0;

print "EXEMPLO702 - SOMAR UM TABELA DE 10 ITENS\n";
      # REPETIR PARA CADA POSICAO
for X in xrange ( 0, 10, 1):
      # a primeira posicao e' zero !
      print "\nFORNECER O ", (X+1), "o. : ";
      V [ X ] = input ( );
      # FIM REPETIR

SOMA = 0;

      # REPETIR PARA CADA POSICAO
for X in xrange ( 0, 10, 1):
      SOMA = SOMA + V [ X ];

print "\nSOMA = ", SOMA;

print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );      # para esperar
# fim do programa
""

```

```

""
# ----- EXEMPLO703

# PROGRAMA PARA CALCULAR A MEDIA DE UMA TABELA DE ITENS
V = [0] * 10;
X = 0;
SOMA = 0;
MEDIA= 0.0;

print "EXEMPLO703 - MEDIA DE UMA TABELA DE 10 ITENS\n";

                # REPETIR PARA CADA POSICAO
for X in xrange ( 0, len( V ), 1):

    print "\nFORNECER O ", (X+1), "o. : ";
    V [ X ] = input ( );
                # FIM REPETIR

SOMA = 0;
                # REPETIR PARA CADA POSICAO
for X in xrange ( 0, len( V ), 1):
    SOMA = SOMA + V[ X ];

MEDIA = SOMA / 10.0;
print "\nMEDIA = ", MEDIA;

print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );          # para esperar
# fim do programa
""

""
# ----- EXEMPLO704

# PROGRAMA PARA LER UMA PALAVRA
PALAVRA = "";
X = 0;
TAMANHO = 0;

print "EXEMPLO704 - LER UMA PALAVRA\n";
print "\nFORNECER UMA PALAVRA (NO MAXIMO 09 LETRAS) : ";
PALAVRA = input ( );          # "ABCDE"

# REPETIR PARA CADA POSICAO
TAMANHO = len ( PALAVRA );
print "TAMANHO=", TAMANHO;
print "\nLETRAS DA PALAVRA LIDA : ";
for X in xrange ( 0, TAMANHO, 1 ):
    print X, PALAVRA[ X ];

print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );          # para esperar ""

```

```

""
# ----- EXEMPLO705

# PROGRAMA PARA PROCURAR LETRA EM PALAVRA
LETRA = '0';
PALAVRA = [ ];
X = 0;
ACHAR = True;

print "EXEMPLO705 - PROCURAR LETRA EM UMA PALAVRA\n";

print "\nDIGITAR UMA PALAVRA (NO MAXIMO 09 LETRAS) : ";
PALAVRA = input ( );   # "ABCDE"

print "\nFORNECER A LETRA A SER PROCURADA : ";
LETRA = raw_input ( ); # C

ACHAR = False;
X = 0;
# REPETIR PARA CADA POSICAO
while ( X < len( PALAVRA ) and not ACHAR ):

    if ( PALAVRA[ X ] == LETRA ):
        ACHAR = True;
    else:
        X = X + 1;

                                # FIM REPETIR

if ( ACHAR ):
    print "LETRA ENCONTRADA";
else:
    print "LETRA NAO ENCONTRADA";

print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );           # para esperar
# fim do programa ""

```



```

"""
# ----- EXEMPLO706

# PROGRAMA PARA AVALIAR UM POLINOMIO
P = [ 0.0 ] * (10+1);
Y = 0;
N = 0;
X = 0.0;
PX = 0.0;

print "EXEMPLO706 - LER COEFICIENTES DE UM POLINOMIO\n";

print "\nFORNECER O GRAU DO POLINOMIO : ";
N = input ( );

# REPETIR PARA CADA POSICAO
for Y in xrange ( 0, N+1, 1 ):
    print "\nFORNECER O ", (Y+1), "o COEFICIENTE : ";
    P [ Y ] = input ( );
        # FIM REPETIR

print "\nFORNECER O PONTO DE AVALIACAO : ";
X = input ( );

PX = 0.0;
# REPETIR PARA CADA POSICAO
# DA ULTIMA ATE A PRIMEIRA
for Y in xrange ( 0, N+1, 1 ):
    PX = PX * X + P [ Y ];

print "\nP", X, " = ", PX;

print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );          # para esperar
# fim do programa
"""

```

```

""
# ----- EXEMPLO707

# PROGRAMA PARA AVALIAR O COMPRIMENTO DE UM VETOR
import math
import array

V = array.array ( 'f', ( 0 for _ in xrange(3) ) );
X = 0;
SOMA = 0.0;

print "EXEMPLO707 - COMPRIMENTO DE UM VETOR\n";

print "\nFORNECER O VALOR DE X :";
V[0] = input ( );
print "\nFORNECER O VALOR DE Y :";
V[1] = input ( );
print "\nFORNECER O VALOR DE Z :";
V[2] = input ( );

SOMA = 0.0;
# REPETIR PARA CADA POSICAO
for X in xrange ( 0, 3, 1 ):
    SOMA = SOMA + V [ X ] * V [ X ];

print "\nCOMPRIMENTO = ", math.sqrt(SOMA);

print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );          # para esperar
# fim do programa
""

```

```

""
# ----- EXEMPLO708

# PROGRAMA PARA LER UMA MATRIZ
M = [ [ 0 for i in range(2) ] for j in range(2) ];
X = 0;
Y = 0;

print "EXEMPLO708 - LER UMA MATRIZ INTEIRA 2x2\n";

                # REPETIR PARA CADA LINHA
for X in xrange ( 0, 2, 1 ):
                # REPETIR PARA CADA COLUNA
    for Y in xrange ( 0, 2, 1 ):

        print "\nFORNECER ELEMENTO ", (X+1), ", ", (Y+1), " : ";
        M[ X ][ Y ] = input ( );
                # FIM REPETIR
                # FIM REPETIR
    print;
                # REPETIR PARA CADA LINHA
for X in xrange ( 0, 2, 1 ):
                # REPETIR PARA CADA COLUNA
    for Y in xrange ( 0, 2, 1 ):

        print M[ X ][ Y ], " ",
        print;
                # FIM REPETIR

print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );      # para esperar
# fim do programa
""

```

```

""
# ----- EXEMPLO709

# PROGRAMA PARA MONTAR A TRANSPOSTA DE UMA MATRIZ
M = [ [ 0 for i in range(2) ] for j in range(2) ];    # MATRIZ ORIGINAL
MT = [ [ 0 for i in range(2) ] for j in range(2) ];    # MATRIZ TRANSPOSTA
X = 0;
Y = 0;

print "EXEMPLO709 - TRANSPOR UMA MATRIZ INTEIRA 2x2\n";

# REPETIR PARA CADA LINHA
for X in range ( 2 ):
    # REPETIR PARA CADA COLUNA
    for Y in range ( 2 ):

        print "\nFORNECER ELEMENTO ", (X+1), ", ", (Y+1), " : ";
        M [ X ][ Y ] = input ( );
        MT[ Y ][ X ] = M[ X ][ Y ];
        # FIM REPETIR
    # FIM REPETIR
print;
# REPETIR PARA CADA LINHA
for X in range ( 2 ):
    # REPETIR PARA CADA COLUNA
    for Y in range ( 2 ):
        print MT[ X ][ Y ], " ",
    prin ;
    # FIM REPETIR

print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );    # para esperar
# fim do programa
""

```

```

""
# ----- EXEMPLO710

# PROGRAMA PARA MOSTRAR A DIAGONAL DE UMA MATRIZ
M = [ [ ] ];          # MATRIZ ORIGINAL
X = 0;
Y = 0;
ORDEM = 0;

print "EXEMPLO710 - MOSTRAR A DIAGONAL DE UMA MATRIZ\n";

# DEFINIR O TAMANHO DA MATRIZ
print "FORNECER A ORDEM DA MATRIZ: ";
ORDEM = input ( );
M = [ [ 0 for X in range(ORDEM) ] for Y in range(ORDEM) ];

# REPETIR PARA CADA LINHA
for X in xrange ( 0, ORDEM, 1 ):
    # REPETIR PARA CADA COLUNA
    for Y in xrange ( 0, ORDEM, 1 ):
        print "\nFORNECER ELEMENTO ", (X+1), ", ", (Y+1), " : ";
        M [ X ][ Y ] = input ( );
    # FIM REPETIR
# FIM REPETIR
print;
# REPETIR PARA CADA LINHA
print "ELEMENTOS NA DIAGONAL:"
for X in xrange ( 0, ORDEM, 1 ):
    # REPETIR PARA CADA COLUNA
    for Y in xrange ( 0, ORDEM, 1 ):
        if ( X == Y ):
            #SE ESTIVER NA DIAGONAL
            print M[ X ][ Y ], " ",
        # FIM REPETIR
# FIM REPETIR
print;

print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );      # para esperar
""

```

```

#
# OBS.: RETIRAR OS COMENTARIOS "" ""
#PARA TESTAR CADA EXEMPLO INDIVIDUALMENTE.
#

"""
# PROGRAMA PARA CALCULAR A DISTANCIA ENTRE PONTOS
from math import *;

class PONTOS:
    X = Y = Z = 0;

P1 = PONTOS ( );
P2 = PONTOS ( );
P3 = PONTOS ( );
D = 0.0;

print "EXEMPLO801 - DISTANCIA ENTRE PONTOS\n";

print "\n ENTRE COM O PRIMEIRO PONTO : \n";
P1.X = input ( ); P1.Y = input ( ); P1.Z = input ( );

print "\n ENTRE COM O SEGUNDO PONTO : \n";
P2.X = input ( ); P2.Y = input ( ); P2.Z = input ( );

P3.X = P2.X - P1.X;
P3.Y = P2.Y - P1.Y;
P3.Z = P2.Z - P1.Z;

print "P3=[", P3.X, ",", P3.Y, ",", P3.Z, "];

D = sqrt( pow(P3.X, 2.0)+
          pow(P3.Y, 2.0)+
          pow(P3.Z, 2.0) );

print "\n DISTANCIA = ", D;

print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );      # para esperar
"""

```

```

""
# ----- EXEMPLO802

# PROGRAM PARA CALCULAR A DISTANCIA ENTRE PONTOS
from math import *;

class PONTOS:
    X = Y = Z = 0;

P1 = PONTOS ( );
P2 = PONTOS ( );

D = 0.0;

print "EXEMPLO802 - DISTANCIA ENTRE PONTOS\n";

print "\n ENTRE COM O PRIMEIRO PONTO : \n";
P1.X = input ( ); P1.Y = input ( ); P1.Z = input ( );

print "\n ENTRE COM O SEGUNDO PONTO : \n";
P2.X = input ( ); P2.Y = input ( ); P2.Z = input ( );

D = sqrt( pow(P2.X-P1.X, 2.0)+
          pow(P2.Y-P1.Y, 2.0)+
          pow(P2.Z-P1.Z, 2.0) );

print "DISTANCIA = ", D;

print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );      # para esperar
# fim do programa
""

```

```

""
# ----- EXEMPLO803

# PROGRAMA PARA CALCULAR A DISTANCIA ENTRE PONTOS
from math import *;

class PONTOS:
    V = [ ]; # lista sera' definida posteriormente

P1 = PONTOS ( );
P1.V = [ 0.0 ] * 3;
P2 = PONTOS ( );
P2.V = [ 0.0 ] * 3;
D = 0;

print "EXEMPLO803 - DISTANCIA ENTRE PONTOS\n";

print "\n ENTRE COM O PRIMEIRO PONTO : \n";
P1.V [0] = input ( ); P1.V[1] = input ( ); P1.V [2] = input ( );
print "[",P1.V[0],",",P1.V[1],",",P1.V[2], "];

print "\n ENTRE COM O SEGUNDO PONTO : \n";
P2.V [0] =input ( ); P2.V[1] = input ( ); P2.V [2] = input ( );
print P2.V;

D = sqrt( pow(P1.V[0]-P2.V[0], 2.0)+
          pow(P1.V[1]-P2.V[1], 2.0)+
          pow(P1.V[2]-P2.V[2], 2.0) );
print "\n DISTANCIA = ", D;

print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( ); # para esperar
# fim do programa
""

```



```

"""
# ----- EXEMPLO804

# PROGRAMA PARA CALCULAR A DISTANCIA ENTRE PONTOS
from math import *;

class PONTOS:
    X = Y = Z = 0;

V = [ PONTOS( ), PONTOS( ) ];
D = 0.0;

print "EXEMPLO804 - DISTANCIA ENTRE PONTOS\n";

print "\n ENTRE COM O PRIMEIRO PONTO : \n";
V[0].X = input ( ); V[0].Y = input ( ); V[0].Z = input ( );

print "\n ENTRE COM O SEGUNDO PONTO : \n";
V[1].X = input ( ); V[1].Y = input ( ); V[1].Z = input ( );

D = sqrt( pow(V[0].X-V[1].X, 2.0)+
          pow(V[0].Y-V[1].Y, 2.0)+
          pow(V[0].Z-V[1].Z, 2.0) );
print "\n DISTANCIA = ", D;

print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );          # para esperar
# fim do programa
"""

```

```

""
# ----- EXEMPLO805

# PROGRAMA PARA CALCULAR A DISTANCIA ENTRE PONTOS
from math import *;

class PONTO:
    X = Y = Z = 0;

class PONTOS:
    P1 = PONTO ( );
    P2 = PONTO ( );

P = PONTOS ( );
D = 0;

print "EXEMPLO805 - DISTANCIA ENTRE PONTOS\n";

print "\n ENTRE COM O PRIMEIRO PONTO : \n";
P.P1.X = input ( ); P.P1.Y = input ( ); P.P1.Z = input ( );

print "\n ENTRE COM O SEGUNDO PONTO : \n";
P.P2.X = input ( ); P.P2.Y = input ( ); P.P2.Z = input ( );

D = sqrt( pow(P.P2.X-P.P1.X, 2.0)+
          pow(P.P2.Y-P.P1.Y, 2.0)+
          pow(P.P2.Z-P.P1.Z, 2.0) );
print "\n DISTANCIA = ", D;

print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );          # para esperar
# fim do programa
""

```

```

"""
# ----- EXEMPLO806
from math import *;

# PROGRAMA PARA CALCULAR A DISTANCIA ENTRE PONTOS
class PONTO:
    X = Y = Z = 0;

class PONTOS:
    P1 = PONTO ( );
    P2 = [ ];

P = PONTOS ( );
P.P2 = [ 0.0 ] * 3;

D = 0;

print "EXEMPLO806 - DISTANCIA ENTRE PONTOS\n";

print "\n ENTRE COM O PRIMEIRO PONTO : \n";
P.P1.X = input ( ); P.P1.Y = input ( ); P.P1.Z = input ( );

print "\n ENTRE COM O SEGUNDO PONTO : \n";
P.P2[0] = input ( ); P.P2[1] = input ( ); P.P2[2] = input ( );

D = sqrt( pow(P.P2[0]-P.P1.X, 2.0)+
          pow(P.P2[1]-P.P1.Y, 2.0)+
          pow(P.P2[2]-P.P1.Z, 2.0));
print "\n DISTANCIA = ", D;

print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );      # para esperar
# fim do programa
"""

```

```

""
# ----- EXEMPLO807

# ----- EXEMPLO807

# PROGRAMA PARA CALCULAR A DISTANCIA ENTRE PONTOS
from math import *;

P = [ [ 0.0 for X in range(3) ] for Y in range(2) ];
D = 0;

print "EXEMPLO807 - DISTANCIA ENTRE PONTOS\n";

print "\n ENTRE COM O PRIMEIRO PONTO : \n";
P[0][0] = input ( ); P[0][1] = input ( ); P[0][2] = input ( );

print "\n ENTRE COM O SEGUNDO PONTO : \n";
P[1][0] = input ( ); P[1][1] = input ( ); P[1][2] = input ( );
print

print P;

D = sqrt( pow(P[0][0]-P[1][0], 2.0)+
          pow(P[0][1]-P[1][1], 2.0)+
          pow(P[0][2]-P[1][2], 2.0));
print "\n DISTANCIA = ", D;

print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );          # para esperar
# fim do programa
""

```

```

""
# ----- EXEMPLO808

# PROGRAMA PARA CALCULAR A DISTANCIA ENTRE PONTOS
from math import *;

class PONTOS:
    V = [ [] ];

P = PONTOS ( );
P.V = [ [ 0.0 for X in range(3) ] for Y in range(2) ];
D = 0;

print "EXEMPLO808 - DISTANCIA ENTRE PONTOS\n";

print "\n ENTRE COM O PRIMEIRO PONTO : \n";
P.V[0][0] = input ( ); P.V[0][1] = input ( ); P.V[0][2] = input ( );

print "\n ENTRE COM O SEGUNDO PONTO : \n";
P.V[1][0] = input ( ); P.V[1][1] = input ( ); P.V[1][2] = input ( );
print

print P.V;

D = sqrt( pow(P.V[0][0]-P.V[1][0], 2.0)+
          pow(P.V[0][1]-P.V[1][1], 2.0)+
          pow(P.V[0][2]-P.V[1][2], 2.0));
print "\n DISTANCIA = ", D;

print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );      # para esperar
""

```

```

"""
# ----- EXEMPLO809
from math import *;

class PONTOS:
    X = Y = Z = 0;

P = [ PONTOS( ) for X in range(2) ];
D = 0.0;

print "EXEMPLO809 - DISTANCIA ENTRE PONTOS\n";

print "\n ENTRE COM O PRIMEIRO PONTO : \n";
P[0].X = input ( ); P[0].Y = input ( ); P[0].Z = input ( );

print "\n ENTRE COM O SEGUNDO PONTO : \n";
P[1].X = input ( ); P[1].Y = input ( ); P[1].Z = input ( );

D = sqrt( pow(P[0].X-P[1].X, 2.0)+
          pow(P[0].Y-P[1].Y, 2.0)+
          pow(P[0].Z-P[1].Z, 2.0) );
print "\n DISTANCIA = ", D;

print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );          # para esperar
"""

```

```

""
# ----- EXEMPLO810
from math import *;

class PONTOS:
    class PONTO:
        X = Y = Z = 0;
        P1 = PONTO ( );
        P2 = PONTO ( );

# PROGRAMA PARA CALCULAR A DISTANCIA ENTRE PONTOS
P = PONTOS ( );
D = 0;

print "EXEMPLO810 - DISTANCIA ENTRE PONTOS\n";
print "\n ENTRE COM O PRIMEIRO PONTO : \n";
P.P1.X = input ( ); P.P1.Y = input ( ); P.P1.Z = input ( );

print "\n ENTRE COM O SEGUNDO PONTO : \n";
P.P2.X = input ( ); P.P2.Y = input ( ); P.P2.Z = input ( );

D = sqrt(pow(P.P2.X-P.P1.X, 2.0)+
        pow(P.P2.Y-P.P1.Y, 2.0)+
        pow(P.P2.Z-P.P1.Z, 2.0));

print "\n DISTANCIA = ", D;

print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );          # para esperar
# fim do programa
""

```

```

#
# OBS.: RETIRAR OS COMENTARIOS "" ""
#PARA TESTAR CADA EXEMPLO INDIVIDUALMENTE.
#

""
# ----- EXEMPLO901
# PROGRAMA PARA GRAVAR COORDENADAS DE PONTOS
class PONTOS:
    X = Y = Z = 0;

P = PONTOS ( );
X = 0;
A = None;

print "EXEMPLO901 - GRAVAR COORDENADAS DE PONTOS \n";

A = open ( "PONTOS1.TXT", "wt" );

for X in range ( 2 ):
    print "\nENTRE COM AS COORDENADAS DE UM PONTO : \n";
    P.X = input ( ); P.Y = input ( ); P.Z = input ( );
    A.write ( ""+str(P.X)+"\n" );
    A.write ( ""+str(P.Y)+"\n" );
    A.write ( ""+str(P.Z)+"\n" );

A.close ( );

print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );          # para esperar
# fim do programa
""

```



```

"""
# ----- EXEMPLO902
# PROGRAMA PARA LER ARQUIVO DE PONTOS
class PONTOS:
    X = Y = Z = 0;

P = PONTOS ( );
X = 0;
A = None;
line = "";

print "EXEMPLO902 - LER ARQUIVO DE PONTOS \n";

A = open ( "PONTOS1.TXT", "rt" );
for X in range ( 2 ):
    line = A.readline ( );
    P.X = line.strip ( '\n' );
    line = A.readline ( );
    P.Y = line.strip ( '\n' );
    line = A.readline ( );
    P.Z = line.strip ( '\n' );
    print "P=[", P.X, ",", P.Y, ",", P.Z, "];

A.close ( );

print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );          # para esperar
# fim do programa
"""

```

```

"""
# ----- EXEMPLO903
# PROGRAMA PARA GRAVAR COORDENADAS DE PONTOS
class PONTOS:
    X = Y = Z = 0;

P = PONTOS ( );
X = 0;
A = None;

print "EXEMPLO903 - GRAVAR COORDENADAS DE PONTOS \n";

A = open ( "PONTOS2.DAT", "wb" );

for X in range ( 2 ):
    print "\nENTRE COM AS COORDENADAS DE UM PONTO : \n";
    P.X = input ( ); P.Y = input ( ); P.Z = input ( );
    b = pack ( "iii", P.X, P.Y, P.Z );
    A.write ( b );
A.close ( );

print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );      # para esperar
# fim do programa
"""

```

```

""
# ----- EXEMPLO904
# ----- EXEMPLO904
# PROGRAMA PARA LER ARQUIVO DE PONTOS
from struct import *;
import sys;

class PONTOS:
    X = Y = Z = 0;

P = PONTOS ( );
X = 0;
A = None;
line = "";

print "EXEMPLO904 - LER ARQUIVO DE PONTOS \n";

A = open ( "PONTOS2.DAT", "rb" );

for X in range ( 2 ):
    b = A.read ( calcsize ("iii") );
    P.X, P.Y, P.Z = unpack_from ( "iii", b );
    print "P=[", P.X, ",", P.Y, ",", P.Z, "];

A.close ( );

print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );          # para esperar
# fim do programa
""

```

```

""
# ----- EXEMPLO905

# PROGRAMA PARA COPIAR O ARQUIVO COM COORDENADAS DE PONTOS
from struct import *;
import sys;

class PONTOS:
    X = Y = Z = 0;

P = PONTOS ( );
X = 0;
A1 = None;
A2 = None;
line = "";

print "EXEMPLO905 - COPIAR COORDENADAS DE PONTOS \n";

A1 = open ( "PONTOS1.TXT", "rt" );
A2 = open ( "PONTOS3.DAT", "wb" );

for X in range ( 2 ):
    line = A1.readline ( );
    P.X = int(line.strip ( '\n' ));
    line = A1.readline ( );
    P.Y = int(line.strip ( '\n' ));
    line = A1.readline ( );
    P.Z = int(line.strip ( '\n' ));
    b = pack ( "iii", P.X, P.Y, P.Z );
    A2.write ( b );
    print "COPIADO P=[" , P.X, ", " , P.Y, ", " , P.Z, "]" ;

A1.close ( );
A2.close ( );

print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );      # para esperar
# fim do programa
""

```

```

"""
# ----- EXEMPLO906
# PROGRAMA PARA ACRESCENTAR COORDENADAS DE PONTOS
class PONTOS:
    X = Y = Z = 0;

P = PONTOS ( );
X = 0;
A = None;
line = "";

print "EXEMPLO906 - ACRESCENTAR COORDENADAS DE PONTOS \n";

A1 = open ( "PONTOS1.TXT", "rt" );
A2 = open ( "PONTOS4.TXT", "wt" );

line = A1.readline( );
while line != "":
    A2.write ( line );
    line = A1.readline( );

A1.close ( );
A2.close ( );

A2 = open ( "PONTOS4.TXT", "a" );

for X in range ( 2 ):
    print "\nENTRE COM AS COORDENADAS DE UM PONTO : \n";
    P.X = input ( ); P.Y = input ( ); P.Z = input ( );
    A2.write ( ""+str(P.X)+"\n" );
    A2.write ( ""+str(P.Y)+"\n" );
    A2.write ( ""+str(P.Z)+"\n" );
print;

A2.close ( );

print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );      # para esperar
# fim do programa
"""

```

```

""
# ----- EXEMPLO907
# PROGRAMA PARA ACRESCENTAR COORDENADAS DE PONTOS
from struct import *;
import sys;

class PONTOS:
    X = Y = Z = 0;

P = PONTOS ( );
X = 0;
A = None;
line = "";

print "EXEMPLO907 - ACRESCENTAR COORDENADAS DE PONTOS \n";

A1 = open ( "PONTOS2.DAT", "rb" );
A2 = open ( "PONTOS5.DAT", "wb" );

b = A1.read ( calcsize ("iii" ) );
while b:
    A2.write ( b );
    b = A1.read ( calcsize ("iii" ) );

A1.close ( );
A2.close ( );

A2 = open ( "PONTOS5.DAT", "ab+" );

for X in range ( 2 ):
    print "\nENTRE COM AS COORDENADAS DE UM PONTO : \n";
    P.X = input ( ); P.Y = input ( ); P.Z = input ( );
    b = pack ( "iii", P.X, P.Y, P.Z );
    A2.write ( b );
print;

A2.close ( );

print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );      # para esperar
# fim do programa
""

```

```

""
# ----- EXEMPLO908

class PONTOS:
    X = Y = Z = 0;

P = PONTOS ( );
PROCURADO = PONTOS ( );
X = 0;
A = None;
line = "";
ACHAR = False;

print "EXEMPLO908 - PROCURAR COORDENADAS DE PONTOS \n";

print "\nENTRE COM AS COORDENADAS DE UM PONTO : \n";
PROCURADO.X = str(input ( ));
PROCURADO.Y = str(input ( ));
PROCURADO.Z = str(input ( ));

A = open ( "PONTOS1.TXT", "rt" );

line = A.readline ( );
while line != "" and not ACHAR:
    P.X = line.rstrip ( );
    line = A.readline ( );
    P.Y = line.rstrip ( );
    line = A.readline ( );
    P.Z = line.rstrip ( );

    if ( P.X==PROCURADO.X and P.Y==PROCURADO.Y and P.Z==PROCURADO.Z ):
        ACHAR = True;

    line = A.readline ( );

A.close ( );

if( ACHAR ):
    print "\nCOORDENADAS ENCONTRADAS";
else:
    print "\nCOORDENADAS NAO ENCONTRADAS";

print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );          # para esperar
# fim do programa
""

```

```

""
# ----- EXEMPLO909
# PROGRAMA PARA ALTERAR COORDENADAS DE PONTOS
from struct import *;
import sys;

class PONTOS:
    X = Y = Z = 0;

P = PONTOS ( );
A = None;
LAST_INDEX = 0;
INDEX = 0;
OPTION = '0';

print "EXEMPLO909 - ALTERAR COORDENADAS DE PONTOS \n";

A = open ( "PONTOS5.DAT", "r+b" );

LAST_INDEX = A.tell ( );
b = A.read ( calcsize ("iii") );
while b:
    P.X, P.Y, P.Z = unpack_from ( "iii", b );
    print "P=[", P.X, ",", P.Y, ",", P.Z, "]";
    INDEX = A.tell ( );
    print " ALTERAR [0=NAO, 1=SIM]? ";
    OPTION = raw_input ( );
    if OPTION == '1':
        print "\nENTRE COM AS NOVAS COORDENADAS DO PONTO : \n";
        P.X = input ( ); P.Y = input ( ); P.Z = input ( );
        b = pack ( "iii", P.X, P.Y, P.Z );
        A.seek ( LAST_INDEX, 0 );
        A.write ( b );
        A.seek ( INDEX, 0 );
        LAST_INDEX = A.tell ( );
        b = A.read ( calcsize ("iii") );

A.close ( );

print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );          # para esperar
# fim do programa
""

```



```

""
# ----- EXEMPLO910

# PROGRAMA PARA LER ARQUIVO DE PONTOS DIRETAMENTE
from struct import *;
import sys;

class PONTOS:
    X = Y = Z = 0;

P = PONTOS ( );
PROCURADO = PONTOS ( );
X = 0;
A = None;
LAST_INDEX = 0 ;
INDEX    = 0 ;
OPTION   = '0';
TAMANHO  = calcsiz ( "iii" );

print "EXEMPLO910 - LER ARQUIVO DE PONTOS DIRETAMENTE \n";

A = open ( "PONTOS5.DAT", "r+b" );

INDEX = input ( "POSICAO? " );      # a primeira e' igual a zero
if INDEX >= 0:
    A.seek ( INDEX*TAMANHO, 0 );
    b = A.read ( TAMANHO );
    while b and not OPTION == '1':
        P.X, P.Y, P.Z = unpack_from ( "iii", b );
        print "P=[", P.X, ", ", P.Y, ", ", P.Z, "]";
        print INDEX, " ENCERRAR [0=NAO, 1=SIM]? ";
        OPTION = raw_input ( );
        if OPTION == '0':
            INDEX = input ( "POSICAO? " );
            if INDEX >= 0:
                A.seek ( INDEX*TAMANHO, 0 );
                b = A.read ( TAMANHO );
            else:
                print "ERRO: POSICAO INVALIDA.";
                OPTION = '1';
        else:
            print "ERRO: POSICAO INVALIDA.";

print "\nPRESSIONAR <Enter> PARA TERMINAR";
raw_input ( );      # para esperar
# fim do programa
""

```