

Engenharia informatica

Escreva a função e_caracter que verifica se um determinado caracter é um numero ou não.

Escreva um Programa de teste da função.

```
#include <stdio.h>

Int e_caracter (char c) {

    Return( c => '0' && c <= '9');

}

Escreva todos os caracteres que não são numeros

Main () {

    Char c1;

    While(1) { // termina c1 ctrl-c

        C1=getchar ();

        If (!e_caracter (c1)){ // se não for numeros

            Putchar ©;

        }

    }

}
```

Escreva funções e_maiuscula que transforma qualquer caracter na sua maiuscula correspondente, escreva uma função que teste essa condição.

```
#include <stdio.h>

Int e_maiuscula (char c) {

    If (c => 'a' && c <= 'z') { /* se minuscula */

        Return c+ 'A' - 'a';

    } else{

        Return c; /*devolve a mesma */

    }

}
```

Escreva todos os caracteres em maiusculos

```
Main () {  
    Char c;  
    While (1){ /* termina ctrl +c  
        = getchar ();  
        Putchar (e_maiuscula (c) )  
    }  
}
```

Implente a função

```
Float pot (float x, int n )  
Devolve o valor de x^n  
X^0=1.0  
X^n= x*x *x( n vezes)  
#include <stdio.h>  
Flot pot (float x, int n) {  
    Res*=n; /* res =res +n; */  
}  
Return res;  
Main () {  
    Printf ("%f%f%f", pot (2.0,4), pot (1.234,4), pot (3.0,0)
```

Dadas as funções tiro e liro

```
Void tiro ( int x){  
  
    Switch (x) {  
  
        Case 1: printf("\n Este teste é mesmo dificil");  
  
        Case 2: printf("\n A, de quem diga o contrario ....");  
  
        Case 3: printf("\n estou no tiro e x=%d", x);  
  
        Liro (x++);  
  
        Break  
  
        Printf ("\\n vou sair do tiro")  
  
        Default: printf("#\\n entrei pelo default");  
  
        Liro (x)  
  
    }  
  
}
```

```
Void liro (int x){
```

```
    Switch (x) {  
  
        Case 2: return;  
  
        Printf("\n Olereilereilolela      "); break;  
  
        Case 3: Printf("\n Eu percebo .... De c      "); break;  
  
        Printf("\n sou mesmo bom      "); break;  
  
        Case 4: Printf("\n não percebo nada disto"); break;  
  
        Liro (2); return;  
  
        Default: printf( "n\\ lá estou mais uma vez");  
  
        Tiro(--);  
  
    }
```

Vetores e arrays

Conjunto de elementos consecutivos, todos o mesmo tipo, que podem ser acedidos individualmente a partir de um único nome.

Declaração de vectores

Tipo – tipo de dados de cada um dos elementos do vector.

Nome_vector - nome pelo qual seá o vector conhecido

Nº de elementos – valor constante, indica quantos elementos tem o vector.

Tipo nome_vector (nº de elementos)

Exemplos

```
Int LEI [68]; /* LEI é um vector com 68 inteiros*/
```

```
Float LCC [300]; /* LCC é um vector com 300 reais */
```

Indice – serve para identificar cada elemento do vector em relação a que posição no vector

LCC [i] – posição do indice i do vector lcc

```
Int LEI [5];
```

LEI [0]	LEI [1]	LEI [2]	LEI [3]	LEI [4]
---------	---------	---------	---------	---------

Os indices variam entre 0 e n-1.

Coloque o vector nº de alunos na segunda posição do vector LEI [1]=201501234;

2	201501234	6		4
LEI [0]	LEI [1]	LEI [2]	LEI [3]	LEI [4]

Coloque na ultima posição o dobro do valor do primeiro elemento que é 2.

$$\text{LEI [4]} = \text{LEI [0]} * 2$$

Coloque na terceira posição a soma do primeiro com o ultimo elemento.

$$\text{LEI [2]} = \text{LEI [0]} + \text{LEI [4]}$$

Inicialização automatica de vectores

Tipo vector_nom[n]= {valor1, valor2,..., valorn};

Declare e initialize um vector com todas as vogais do alfabeto.

Char vogal [5]={‘a’, ‘e’, ‘i’, ‘o’, ‘u’};

Nota = Se um vector for declarado com os elementos e forem colocados apenas k valores (k<n)

Na inicialização com os elementos do vector , então os primeiros k elementos do vector serão inicializados com os respectivos valores e os restantes serão inicializados com valor zero.

Int vogal [5]={‘1’, ‘2’};

Escreva um programa que realize a leitura dos salarios pagos a um funcionario durante um ano. Em seguida deverá mostrar os valores mensais e o total anual.

```
Float salario [12];  
  
Float total;  
  
Inr i;  
  
For (i=0; i< 12; i++){    for (int i=1; 1<=12, i++){  
  
    Printf("Introduza o salario do mês %d", i+1);  
  
    Scanf("%f", &salario[i]);  
  
}  
  
Puts("mês valor");  
  
For (i=0; total :0.0; i<12; i++){    for (int i=1; 1<=12, i++){  
  
    Printf(%3d  %9.2f\n", i+1, salario [i]);  
  
    Total +=salario[i];  
  
}  
  
Printf ("total Anual “0.22f\n”; total );  
  
}
```

Vectores – funções

```
Void inic 1 ( int s [10] )  
Int i;  
For( i=0; i<10, i++) {  
    s[i]=0;  
}  
}  
Void inic 2 ( int s [20] ) {  
Int i;  
For ( i=0; i<20; i++) {  
    S [i]=0;  
}  
}  
Main () {  
Int v[10];  
Int x[20];  
inic 1[v]; /* inicializ o vector v com e função inic 1*/  
inic 2[x]; /* inicializ o vector x com e função inic 2*/
```

Funções e procedimentos

Escreva um programa que coloque na tela o seguinte output, escrevendo de um ciclo.

**

Numero entre 1 e 5

**

1

2

3

4

5

Main () {

For (i=1; i<=20; i++) {

Putchar ('*');

Putchar ('\n');

Puts ("numero em 1 a 5");

}

For (i=1; i<=20; i++) {

Putchar ('*');

Putchar ('\n');

}

For (i=1; i<=5 i++) {

Printf ("%d\n", i);

}

For (i=1; i<=20; i++) {

Putchar ('*');

Putchar ('\n');

}

```

#include <stdio.h>

Linha () {
    Int i;
    For(i=1; 1<=20;i++){
        Putchar ('*');
        Putchar ('\n');
    }
}

Main ( ) {
    Int i;
    Linha ();
    Puts ("numero entre 1 e 5");
    Linha ()
    For (i=1; 1<=5;i++){
        Print (%d\n",i);
    }
    Linha ();
}

```

Construa um programa com o seguinte ouput usando funlções

```

*
**
***
****
*****
*****
**
*

```

```
# include <stdio.h>

Linha (int num){

Int i;

For (i=1; i<=num; i++){
    Putchar ('*');
    Putchar ('\n');

}

Main () {
    Linha (1); linha (2); linha(3); linha (4);

    Linha(5); linha(4); linha(3); linha (2);

}
}
```

```
Main () {  
    Função ('A','123','23456');  
}  
  
Funcao (char a, int n, float x){  
}
```

Escreva um programa em que a função linha escreva qualquer caracter e não apenas asterísticos

```
#include <stdio.>  
  
Linha (int num, char a) {  
  
    Int i;  
  
    Putchar (a);  
  
    Putchar ('\n');  
}  
  
}  
  
Main () {  
  
    linha (3,'+');  
  
    linha (5,'-');  
  
    linha (7,'/');  
  
    linha (5,'*');  
  
    linha (3,'+');
```

Função Return

Função que permite terminar a execução de uma função e voltar ao programa que -.....

A execução e instrução return função main faz com que o programa termine.

```
Main () {  
    Printf ("Angola")  
    Return ; /* fim da função e programa  
    Printf "world\n"); /* linha não executada  
}
```

Função com retorno de valor

Nome de dois numeros

```
N=soma (3,4);  
Printf("%d\n",n);
```

```
Int soma (int x, int y) {  
    Int resultado;  
    Resultado=x+y;  
    Return resultado  
}
```

Escreva um programa que solicite dois numeros ao utilizador e apresente no ecrã o resultado de sua soma e o dado de cada um deles.

```
/* devolve a soma de dois caracter

Int soma (int a, int b) {

Return a+b;

}

/* devolve o dado de qualquer inteiro

Int dados (int x) {

Retirn 2* x;

}

Main () {

Int n, i, total ;

Printf ("Introduza dois muneros:");

Scanf ("%d%D",&n, %i);

Total=soma (n,i); /* atributo do resultado da função

Printf("%d*%d=%d\n",n,i, total);

Printf("%d*%d=%d\ e 2*%d=%d\n", n", dobro(n),i);
```

Introduza dois numeros 4+

4+7=11

2*4=8 e 2*7=14

Escreve uma função que retorna o maior entre dois inteiros

```
Int max (int n1, int n2){  
    If(n1> n2)  
        Return n1;  
    Else  
        Return n2;  
}
```

Função e procedimentos

Um função tem sempre um tipo e um valor de retorno associado enquanto um procedimento não devolve qualquer valor

Função -> int max (int n1, int n2)

Procedimento -> linha ()

Sempre que no cabeçario de uma função não é colocada o tipo de retorno , este é substituído pelo tipo de int.

Invocação das funções

X = soma (23, y) + dado (x+2);

Printf ("%d%d ", dado (5), soma (dobro (2), 3+2));

If (soma (x, y,) >0)

Getchar ()

O tipo void

Função não retorna valor

Void limite (int num, char a) {

}

String (.....)

Put s

Puts ("Ana Manuel") ; <-> printf (" Ana Manuel \n")

Scarf

Ao contrario outro tipo , a variavel que recebe string no procedido de um &.

```
#include <stdio.h>

Main () {

    Char nome [50] , apelido [50];

    Printf (" Introduza o nome:");

    Scarf ("%s", nome);

    Printf ("Introduza o apelido");

    Scarf("%s" , apelido);

    Printf ("Nome completo %s%s\n", nome, apelido );

}
```

```
Gets  
#  
Char nome  
Gets (nome);
```

Escreva um programa que leia nome e os apresente no ecrã até que vazio .

Seja introduzido pelo operador main () {

```
Char nome [100];  
While (1) { /* ciclo infinito */  
    Puts ("nome:");  
    Gets (nome);  
    If(nome [0] == '\0') {  
        Break;  
    } else {  
        Printf (" nome introduzido :%s\n", nome );  
    }  
}
```

