

Trabajo Practico N°2
Resuleto

Laboratorio de Computación I
Tecnicatura Superior en Programación.
UTN-FRRQ

FUNCIONES DE ENTRADA- SALIDA

1- Escribir los enunciados necesarios para ingresar por teclado las variables:

- x de tipo entera e y de tipo flotante, separadas por una ,
scanf("%d,%f", &x, &y);
- x de tipo caracter e y de tipo entera, separadas por una /
scanf("%c/%d", &x, &y);
- x de tipo entero largo e y de tipo caracter, separadas por un blanco
long x;
char y;
scanf("%ld %c", &x, &y);

2- Escribir los enunciados necesarios para mostrar:

- La mesa numero **num** esta servida
 - printf("La mesa numero %d esta servida", num);
- Hoy es el dia nro **num** de la semana
 - printf("Hoy es el dia nro %d de la semana", num);
- En la columna nro **num** hay un error
 - printf("En la columna nro %d hay un error", num);
- Los numeros son: **n1, n2, n3**
 - printf("Los números son: %d, %d, %d", n1, n2, n3);
- La cadena "**cad**" contiene **num** caracteres
char cad[] = "cad";
int num;

num = strlen(cad);
printf("La cadena \"%s\" tiene %i caracteres", cad, num);

Donde **num, n1, n2 y n3** son variables enteras y **cad** es una cadena de caracteres

3- Son sintacticamente correctas las siguientes sentencias para ingresos de datos. Tache lo que NO CORRESPONDE

```
{ int a ; char car[10];
scanf( " %d", &a) ;
scanf( " %c", &car[0]) ;
scanf( " %s", car);
scanf( " %s\n", car)
```

SI NO
SI NO
SI NO
SI NO

"El sistema queda esperando el ingreso de otro dato, el cual no se encuentra explicito"

4- Decir si scanf(), printf(), getchar(), clrscr() son:

- a) palabras reservadas del lenguaje C SI NO
- b) funciones cuyos prototipos se encuentran en algún archivo de cabecera cuya extensión es .h SI NO
- c) funciones creadas por el programador SI NO

SENTENCIAS DE CONTROL DE FLUJO

- 5- Decir que muestra cada una de las instrucciones printf.

```

#include <stdio.h>
#include <conio.h>
main( )
{
    int  expresión, x = 1;
    expresion = ! ( (1 == x) && 4)
    if (expresión )
        printf("El resultado de la expresión es verdadero");
    else
        printf("El resultado de la expresion es falso"); /** print **/
    x = 4;
    for( ; x < 10; x += 3);
        printf("%d", x / 2); /** X = 10 => print = 5 **/
        x = 10;
    do
        {
            printf("%d", x--); /** 10 9 8 7 6 **/
        } while ( x >5);
}
  
```

- 6- Incrementar una variable entera **j** desde 0 hasta **n** y luego mostrar el resultado. Hacerlo utilizando 3 bucles diferentes.

```

int j, n, k, total;
scanf("%d,%d",&j, &n);
k = j + n;

while(j != k){
    total = ++j; => total = total + 1
}

do
{
    total = ++j;
}while(j != k);

for(; j != k; J++)
{
    total = ++j;
}

printf("El resultado de %d incrementado %d veces es: %d", j,n,total);
  
```

7- En la expresión **for(i=1; i==10; i+=2)**; cuántas veces se evalúa la condición

○ 1

8- Leer sucesivamente de teclado, hasta que aparezca un número comprendido entre 1 y 5. Desarrollar el algoritmo usando la función :

a) `getchar()` b) `scanf()`

```

char a;
while(a<'1' || a>'5')
{
    fflush(stdin);
    printf("Ingrese un numero: ");
    a = getchar();
}
  
```

```

char a;
while(a<'1' || a>'5')
{
    fflush(stdin);
    printf("Ingrese un numero: ");
    scanf("%c",&a);
}
  
```

9- Codificar en C un programa que lea 20 caracteres indique cuantas "a" se ingresaron, cuantas "e, i, o, u"

```

char letra;
int aux, a=0, e=0, i=0, o=0, u=0;
  
```

```

for(aux=0; aux!=20; aux++){
    printf("Ingrese un caracter:\n ");
    letra = getchar();
  
```

```

    switch(letra){
        case 'a': a++;break;
        case 'e': e++;break;
        case 'i': i++;break;
        case 'o': o++;break;
        case 'u': u++;break;
    }
  
```

```

    printf("a: %d veces\n", a);
    printf("e: %d veces\n", e);
    printf("i: %d veces\n", i);
    printf("o: %d veces\n", o);
    printf("u: %d veces\n", u);
  
```

10- Hacer el algoritmo que imprima los números pares entre 100 y 20 a razón de 6 por línea separados por 3 blancos

```
int i, j, cont=0;
for(i=100; i>=20; i--){
    if(i%2 == 0) {
        printf("%d  ", i);
        cont++;
        if(cont%6==0)
            printf("\n");
    }
}
```

11- Hacer un algoritmo que imprima todos los números primos/pares que hay desde la unidad hasta un número que introducimos por teclado. El programa debe poder ejecutarse mientras el usuario lo requiera.

```
int esPrimo, i, num, cont;
char opc;
do{
    printf("Ingrese un numero: ");
    scanf("%d", &num);

    for(esPrimo=1; esPrimo<num; esPrimo++){
        cont=0;
        for(i=1; i<=esPrimo; i++){
            if(esPrimo%i == 0){
                cont++;
            }
        }

        if(cont<=2)
            printf("%i ", esPrimo);

    }
    fflush(stdin);
    printf("\nEjecutar nuevamente? s/n: ");
    scanf("%c", &opc);
}while(opc!='n');
```

12- Hacer un algoritmo que imprima el mayor y el menor de una serie de 5 números que vamos introduciendo por teclado.

```
int array[5], mayor=0, menor=0, i;

for(i=0; i<5; i++){
    printf("Ingrese un numero: ");
    scanf("%d", &array[i]);
}

menor = array[0];

for(i=0; i<5; i++){
    if(array[i]>mayor) mayor = array[i];
    if(array[i]<menor) menor = array[i];
}

printf("El mayor de los numeros ingresados es: %d\n", mayor);
printf("El menor de los numeros ingresados es: %d\n", menor);
```