

Trabajo Practico N°2

***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 ,
 - x de tipo caracter e y de tipo entera, separadas por una /
 - x de tipo entero largo e y de tipo caracter, separadas por un blanco

- 2- Escribir los enunciados necesarios para mostrar:
 - La mesa numero **num** esta servida
 - Hoy es el día nro **num** de la semana
 - En la columna nro **num** hay un error
 - Los numeros son: **n1, n2, n3**
 - La cadena "**cad**" contiene **num** caracteres
 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) ;	SI NO
scanf(" %c", &car[0]) ;	SI NO
scanf(" %s", car);	SI NO
scanf(" %s\n", car)	SI NO

- 4- Decir si scanf(), printf(), getch(), 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");
    x = 4;
    for( ; x < 10; x += 3);
    printf("%d", x / 2);
```

```
        x = 10;  
do  
    {  
        printf("%d", x--);  
    } while ( x >5);  
}
```

- 6- Incrementar una variable entera **j** desde 0 hasta **n** y luego mostrar el resultado. Hacerlo utilizando 3 bucles diferentes.
- 7- En la expresión **for(i=1; i==10; i+=2);** cuántas veces se evalúa la condición
- 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()*
- 9- Codificar en C un programa que lea 20 caracteres indique cuantas "a" se ingresaron, cuantas "e, i, o, u"
- 10- Hacer el algoritmo que imprima los números pares entre 100 y 20 a razón de 6 por línea separados por 3 blancos
- 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.
- 12- Hacer un algoritmo que imprima el mayor y el menor de una serie de 5 números que vamos introduciendo por teclado.