Práctica Nro 1 – Lenguaje C

Ej1 – Completar la siguiente tabla para los tipos de datos en C:

Tipos de datos de C	Descripción	Equivalente en Pascal	Ejemplos de constantes
char			
float			
int			
long			
char *			

Ej 2 – Sean las siguientes declaraciones de variables e inicializaciones. Completar el cuadro con la salida en cada caso:

```
int A = 34, B =1234, C = -3;
char ch1 ='A';
float F = 22,47;
```

printf("A=%d",A);	
printf("A=%5d",A);	
printf("A=%4d",A);	
printf("A=%-5d",A);	
printf("A=%-4d",A);	
printf("A=%05d",A);	
printf("B=%-05d",A);	
printf("C=%+d",A);	
printf("B=%3d",B);	
printf("C=%c",C);	
printf("C=%3c",C);	
printf("C=%-3c",C);	
printf("F=%f",F);	
printf("F=%7.2f",F);	
printf("F=%07.2f",F);	
printf("F=%-7.2f",F);	
printf("F=%7.1f",F);	
printf("%c",ch1);	
printf("%3c",ch1);	
printf("%s",”ELEFANTE”);	

Ej 3 – Dadas las declaraciones siguientes, completar la función ***scanf()*** con la expresión más apropiada

```
char x, cadena[25];
int n;
scanf("%d",.....);
scanf("%c",.....);
scanf("%s",.....);
```

Ej 4 – Indicar la salida de los siguientes programas, sin ejecutarlos:

a)

```
#include <stdio.h>
void main()  {
    int res, a=10 , b=4;

    res = 9 - b * b + a++;
    if ( res % 2 == 0 )
        printf ("El resultado %d es par. \n", res);
     else
        printf ("El resultado %d es impar. \n", res);
    printf ("El valor de a es %d. \n", a);
}
```

b)

```
#include <stdio.h>
int main(){
    int a = 4, b = 6;
     a -= b++ * 2;
    printf("a =  %d\t b= %d\n", a, b);
  return 0;
}
```

Ej 5 –

a) Reescribir las siguientes sentencias a su equivalente con *if*:

- x = (y<z) ? y : z ;
- printf ("%s", (i%2 == 0) ? "es par" : "es impar");

b) Reescribir la siguiente sentencia utilizando el operador condicional:

```
if (x > 0)
        y = 1;
else
        y = 0;
```

Ej 6 – Reescribir las siguientes sentencias según el ejemplo:

Sentencia	**Equivale a**
n--;	n = n – 1;
n += 4;	
n -= a*2;	
n *= 3+2;	

Ej 7 – Indicar la salida de los siguientes fragmentos de código, sin ejecutarlos:

```
#define n 5
 int suma, i;
```

a)

```
suma = 0;
  for (i=1; i<=n; i ++)
    suma += i;
  printf("%d\n", suma);
```

b)
```
suma = 0;
for (i=1; i<n; i+=2 )
   suma += i;
printf("%d -> %d\n", i,suma);
```

c)
```
for (suma = 1,i=n; i>0; i-- ) {
   suma *= i;
   printf("%d -> %d\n", i, suma);
}
```

Ej 8 – Desarrollar programas para:

- **a)** ingresar N números enteros y calcular y mostrar el promedio de los positivos.
- **b)** ingresar N caracteres y determinar y mostrar la cantidad de vocales indicando si la A es la vocal que más veces apareció.
- **c)** mostrar el valor medio de dos enteros leídos
- **d)** leer N caracteres y hallar y escribir la cantidad de letras mayúsculas ingresadas.
- **e)** leer 3 valores correspondientes a un ángulo en grados, minutos y segundos y obtener y mostrar el valor equivalente en radianes

Ej 9 – Transformar las siguientes sentencias utilizando la instrucción **switch** (x es int, F es char).

a)
```
if (x == 4)
   y = 7;
else
   if (x == 5)
      y = 9;
   else
      if(x == 9)
         y = 14;
      else
         y = 22;
```

b)
```
if (F == 's' || F == 'S')
    r = x + y;
else
  ff (F == 'r' || F == 'R')
     r = x - y;
  else
     if (F == 'm' || F == 'M')
        r = x * y;
     else
       if (F == 'd' || F == 'D')
          y = x/y;
       else
          y = 0;
```

Ej 10 – Detectar y corregir los errores de los siguientes fragmentos de código:

a) mostrar los números naturales divisibles por 5 y menores a 25
```
#define N 50
int i;
    for(i = 5; i <= N; i+=5);
        printf("%d", &i);
```

b) intercambiar dos enteros
```
void InterCambia (int *a, int *b){
  int aux;
  aux = *a;
  a = *b;
  b = &aux;
}
```

c) retornar el mayor valor de un arreglo de N valores naturales
```
int Mayor (int V[], int N){
  int i=0, max=0;
  while (i<N)
      if (V[i] > max)
         return V[i];
      else
         i++;
}
```

Ej 11 – Desarrollar una macro que devuelva cada uno de los siguientes resultados:
- **a)** el cuadrado de un número.
- **b)** la suma de dos números.
- **c)** para un valor que representa grados en radianes.
- **d)** el mínimo de dos números.

Ej 12 – Enumerar los valores de todos los componentes de los siguientes arreglos. Indicar cuáles arreglos de caracteres podrían ser utilizados correctamente como cadenas.
- **a)** int v1[4] = {0};
- **b)** int v2[1]={6};
- **c)** int v3[] = {2,4,6};
- **d)** char s1[4] = {'h','o','y'};
- **e)** char s2[] = {'h','o','y'};
- **f)** char s3[4]= {'h','o','y','\0'};

Ej 13 – Dado un vector de enteros de n elementos, desarrollar una función que encuentre y devuelva la posición del máximo. Implementar 3 veces utilizando for, while y do while.

Ej 14 – Desarrollar un programa que contenga una función void que reciba una cadena que representa una fecha de la forma aaaammdd y devuelva un struct con 3 campos: dd, mm y aaaa. La lectura de la cadena y la impresión del registro deben estar en el *main()*

Ej 15 – Desarrollar un programa que solicite por teclado N y los siguientes datos de N alumnos: Nombre (cadena de 30), Numero de Matricula (entero), Nombre de Carrera (cadena de 30).
Almacenar estos datos en un arreglo de structs.
Muestre por pantalla los datos ingresados con un alumno por línea solo para aquellos que estudian Ingeniería en Informática

Ej 16 – Resolver nuevamente los incisos del **ej8** desarrollando funciones.

Ej 17 – Desarrollar un programa que imprima el equivalente en letras de un número introducido de un rango de 0 a 999.
Por ejemplo: se introdujo **119**, imprime **ciento diecinueve**.

Ej 18 – Desarrollar una función que pase la primera letra a mayúscula de cada palabra de una cadena de caracteres.
Por ejemplo "**calle jujuy 1085 6 a"** pasa a **"Calle Jujuy 1085 6 A".**

Ej 19 – Desarrollar un programa que grabe en un archivo binario NUMENT.DAT números enteros ingresados por teclado, finaliza el ingreso cuando aparecen dos números consecutivos iguales (esos datos van al archivo). Mostrar el archivo generado.

Ej 20 – Desarrollar un programa que lea un archivo de texto que contiene un número entero en cada línea y almacene los números pares es un vector y los impares en otro ignorando los 0.
Luego regrabar el archivo con los pares primero y los impares después.
NOTA: Suponer que en el archivo hay como máximo 50 elementos

Ej 21 – Dado un archivo binario en el que cada registro es de 17 dígitos para tres campos, 5 iniciales de ciudad, 5 superficie y 7 cantidad de habitantes. Escribir un programa que por pantalla muestre la ciudad con mas densidad de habitantes y la de menor densidad.

Ej 22 – Modifique el programa anterior para generar un vector de registros con las 10 primeras ciudades del archivo. Luego encuentre y muestra las iniciales de las ciudades que tengan la mayor y la menor de densidad poblacional entre esas 10.

Ej 23 – En una escuela secundaria para el último año se tiene un archivo de texto **Notas.txt** con las notas finales de las 10 materias, el formato una línea por alumno con Número de matrícula (0 a 99) y las 10 notas finales separadas por espacios (pueden no estar los 100 alumnos ni vienen ordenados)
Desarrollar una función para generar una matriz de 100x10 con las notas finales donde las filas representan el número de matrícula y las columnas las materias (las filas que no representen alumnos pues no se encuentran los datos deben quedar en 0) y luego desarrollar:

- **a)** una función (int) que devuelva la cantidad de alumnos recibidos (todas las notas más de 4)
- **b)** una función (int) que devuelva la cantidad de alumnos que deben rendir más de 2 materias (deben rendir aquellas en las que no tienen al menos 4).
- **c)** una función (void) que devuelva en dos parámetros el alumno recibido con mayor nota promedio (número de matrícula y nota promedio final)

Ej 24 – Dado un archivo de texto que representa la facturación diaria de una empresa. Hay dos tipos de líneas, la de cabecera (una por factura y es la primera), y las de detalle que están a continuación y representan los artículos de la factura.
La cabecera empieza con la letra **F** seguido 12 dígitos del número de factura y 30 caracteres el nombre del cliente.
La de detalle empieza con la letra **D** seguido de 6 dígitos código de artículo, 20 caracteres de descripción de artículo, 7 dígitos para el precio los dos últimos son decimales, y 5 dígitos de cantidad.
Desarrollar un programa que lea el archivo de facturación y genere un archivo binario que por cada factura genere un registro con número de factura, cantidad de ítems e importe total.