PROGRAMACIÓN BÁSICA

PROF. ALMA GONZÁLEZ

INSTRUCCIÓN: WHILE

- La instrucción while nos permite realizar una serie de instrucciones de forma repetida mientras se siga cumpliendo la condición especificada. Es decir nos permite hacer un ciclo. La sintaxis es:

```
while(condicion){

}
```

La condición usualmente es un comparativo entre una variable y un valor de referencia, o bien entre dos variables. La comparación puede ser del tipo: ==,=!,<,>,<=,>=,

EJEMPLO: WHILE

```c
#include<stdio.h>

int main()
{
   float temp_C,temp_K;
    float inicial=100,final=200,delta;
    int n=10;

    delta=(final-inicial)/n;
    temp_C=inicial;
    while(temp_C<=final){
        temp_K=temp_C+273.15;
        printf("%f %f\n",temp_C,temp_K);
        temp_C=temp_C+delta;        //   temp_C+=delta;

     }
    return 0;
}
```

EJEMPLO 2: WHILE

```
#include<stdio.h>

int main()
{
   float temp_C,temp_K;
    float inicial=100,final=200,delta;
    int n=10;
    int op=1;

    delta=(final-inicial)/n;
    while(op==1){
        temp_K=0.;
        temp_C=inicial;

    while(temp_C<=final){
        temp_K=temp_C+273.15;
        printf("%f %f\n",temp_C,temp_K);
        temp_C=temp_C+delta;       //   temp_C+=delta;

    }
        printf("Deseas hacer otra operacion? Presiona 1 para si, Presiona 2 para no)
\n”);
        scanf("%i",&op);
    }
    return 0;
```

STRINGS (CADENAS DE CARACTERES)

- Un string, o cadena de caracteres, es un arreglo que almacena una secuencia de caracteres. Las variables de este tipo se definen como del tipo char:

  char nombre[10];

  En este caso la variable nombre, admitirá una secuencia de 10 caracteres. Podemos asignarle un valor desde la terminal usando scanf:

  scanf("%s",nombre);

  Si sabemos la extensión del string podemos poner el numero de caracteres exacto, si no sabemos podemos poner un numero razonablemente grande.

  Cada elemento de la variable nombre almacena 1 carácter. Ejemplo: supongamos que el nombre que introducimos en la terminal es alma, entonces:

  nombre[0] ⟶ a

  nombre[1] ⟶l

  nombre[2] ⟶-> m

  nombre[3] ⟶->a

EJEMPLO 3: WHILE USANDO UNA CADENA DE CARACTERES.

```
#include<stdio.h>

int main()
{
   float temp_C,temp_K;
    float inicial=100,final=200,delta;
    int n=10;
    char op[2];

    op[0]='s';
    delta=(final-inicial)/n;
    while(op[0]=='s'){
        temp_K=0.;
        temp_C=inicial;

    while(temp_C<=final){
        temp_K=temp_C+273.15;
        printf("%f %f\n",temp_C,temp_K);
        temp_C=temp_C+delta;        //   temp_C+=delta;

    }
        printf("Deseas hacer otra operacion? (si/no)\n");
        scanf("%s",op);
    }
    return 0;
}
```

EJERCICIO:

- Combinar los programas de conversión de temperatura y conversión de coordenadas en un solo programa que nos permita realizar las operaciones que el usuario defina, en los intervalos que el usuario defina. Que al final nos de la opción de realizar otra operación, y en su caso nos muestre nuevamente todo el menu, o bien salga del menu y termine el programa. Asegurate de definir el menor numero de variables posibles.

INSTRUCCIÓN:
FOR

- La instrucción for nos permite realizar una serie de instrucciones de forma repetida por un determinado numero de veces, hace un ciclo. Está instrucción requiere de llevar un contador que indique el numero de veces que se han realizado las instrucciones. Cuando el contador llega a un numero máximo pre-definido el ciclo termina. La sintaxis es:

```
for(i=0;i<=n_iter;i++){

Instrucciones

}
```

INSTRUCCIÓN:
FOR

```
for(i=0;i<n_iter;i++){
}
```

i: contador variable entera,
declarar variable.
puede ser cualquier otro
nombre de variable.
ej: i,j,k,cont, num, etc...

n_iter: variable entera,
declarar variable. Condición
a checar para seguir realizando el ciclo,
si la condición no se cumple el ciclo termina.

i++: la variable i aumenta
en 1 unidad cada vez que se
completa la serie de
instrucciones.

```c
#include<stdio.h>

int main()
{
   float temp_C,temp_K;
    float Temp_C=100,final=200,delta;
    int n=10,i;

    delta=(final-Temp_C)/n;
   for(i=0;i<n;i++){
        temp_K=temp_C+273.15;
        printf("%f %f\n",temp_C,temp_K);
        temp_C=temp_C+delta;        //   temp_C+=delta;

     }
    return 0;
}
```

Añade una instrucción while para que el usuario pueda pedir ejecutar el programa nuevamente (como en el ejercicio de la semana 4). Y haz que el usuario pueda definir las temperaturas inicial (Temp_C), final (final) y numero de pasos (n).

EJEMPLO 2:

```
#include<stdio.h>
#include<math.h>

int main()
{
    float exp_;
     int n=10,j;

    for(j=0;j<n;j++){
        exp_=exp(j);
        printf("%i \t %f\n”,j,exp_);
     }
    return 0;
}
```

EJEMPLO 3: ES POSIBLE HACER CICLOS FOR ANIDADOS (UNO DENTRO DE OTRO)

```
#include<stdio.h>
#include<math.h>

int main()
{
    float res;
     int n=3,k,j;
     for(j=0;j<n;j++){
      for(k=0;k<n;k++){
         res=j*k*1.0;
         printf("%i %i
%f\n”,j,k,res);
      }
}
    return 0;
}
```

EJERCICIOS:

- Haz un programa que evalúe las funciones: exponencial (exp), logaritmo (log), seno (sin), coseno (cos), y raíz cuadrada (sqrt) de la variable x, en un intervalo y con un espaciado definido por el usuario.

- Haz un programa que calcule el factorial de un numero dado por el usuario, y que pregunte si queremos hacer un nuevo calculo.