

Nombre de la práctica	LENGUAJE C Y VARIABLES C			No.	1
Asignatura:	METODOS NUMERICOS	Carrera:	ING. SISTEMAS COMPUTACIONALES	Duración de la práctica (Hrs)	10

Nombre del alumno: Lissette Garcia Nolasco

### I. Competencia(s) específica(s):

### II. Lugar de realización de la práctica (laboratorio, taller, aula u otro):

Aula

### III. Material empleado:

Dev C++

Computadora

### IV. Desarrollo de la práctica:

## CICLO WHILE:

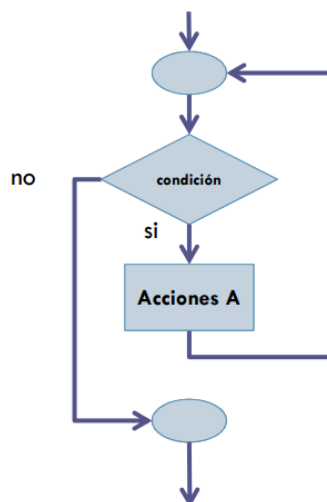
El propósito de este ciclo es repetir un bloque de código mientras una condición se mantenga verdadera.

### ¿Cómo funciona?

Verifica si la condición se cumple

Si es verdadero, ejecuta una o varias instrucciones y nuevamente verifica la condición.

Si es falsa, entonces el ciclo termina.



### Pseudocódigo

Repita\_mientras (<condicion>)

$S_1 \dots$

$\dots$

$S_n \dots$

Fin Repita\_mientras

### Código en C

```
while (<condicion>) {
```

```
 $S_1 \dots$ 
```

```
 $\dots$ 
```

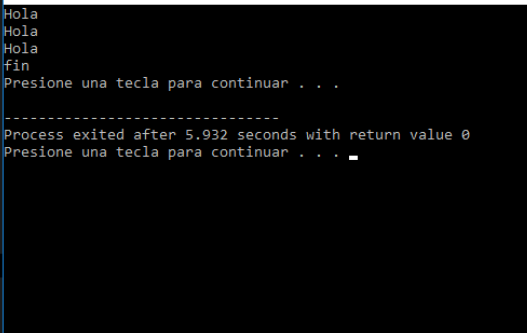
```
 $S_n \dots$ 
```

```
}
```

25/10/2013

## Ejemplo 1.- Imprime 3 veces “hola”:

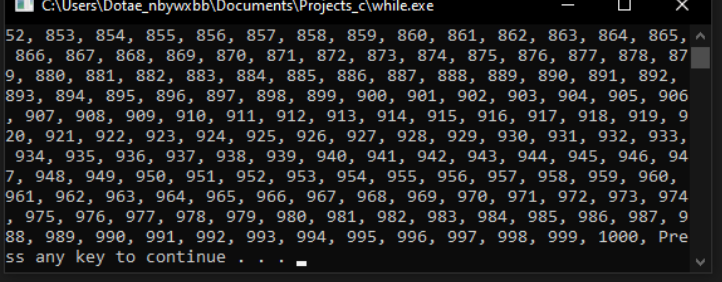
```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int contador=0;
6     while (contador<3){
7         printf ("Hola\n");
8         contador++;
9     }
10    printf ("fin\n");
11    system("Pause");
12    return 0;
13 }
```



## Ejemplo 2.- Impresión de números de 1-1000:

Mediante el ciclo while le estamos indicando que imprima el numero hasta que este cumpla con la condición mencionada inicialmente.

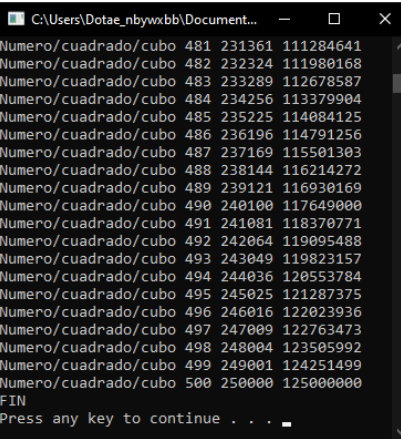
```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int numero=1;
6     while(numero<=1000){
7         printf("%d, ",numero);
8         numero++;
9     }
10    system("Pause");
11    return 0;
12 }
```



## Ejercicio 1.- Imprimir los cuadrados y los cubos de los primeros quinientos números naturales:

Mediante el ciclo while realizamos incrementos y operaciones que nos indicaran el cuadrado y el cubo de cada número menor-igual a 500.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int contador=0;
6     int num=0;
7     int cua,cub;
8
9     while(contador<500){
10        num++;
11        cua=(num*num);
12        cub=(num*num*num);
13        printf("Numero/cuadrado/cubo %d %d %d \n",num,cua,cub);
14        contador++;
15    }
16    puts("FIN");
17    system("Pause");
18    return 0;
19 }
20
```



Ejercicios 2.- Imprimir todos los números divisibles entre 3 mayores a 0 y menores a mil.

Ejemplo:

3, 6, 9, 12, 15, 18, 21, .... 999

Ejercicio 3.- Imprimir todos los números que son divisibles entre 2 y entre 7, mayores a 0 y menores a mil.

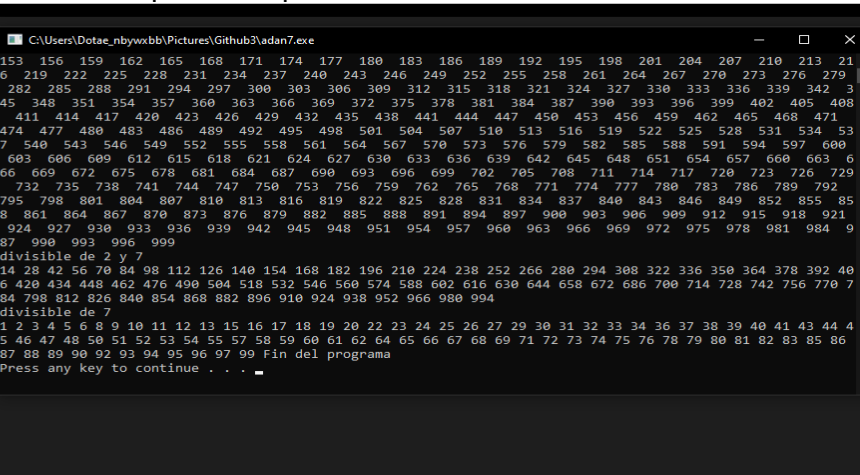
Ejemplo:

14, 28, 42, 56, ...

Ejercicio 4.- Escribir todos los enteros positivos menores que 100 omitiendo aquellos que son divisibles por 7.

Dentro de 3 ciclos le estamos indicando que nos va a imprimir los numero cuyo residuo (%) sea cero de acuerdo a nuestra operación que indicamos

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int x=1,x2=1,x3=1;
6     int y=1,y2=1,y3=1;
7
8     while(y<1000){
9         if(x%3==0){
10            printf("%d ",x);
11        }
12        y++;
13        x++;
14    }
15    puts("\ndivisible de 2 y 7");
16
17    while(y2<1000){
18        if(x2%2==0&&x2%7==0){
19            printf("%d ",x2);
20        }
21        y2++;
22        x2++;
23    }
24    puts("\ndivisible de 7");
25
26    while(y3<100){
27        if(y3%7==0){
28            printf("%d ",y3);
29        }
30        y3++;
31    }
32
33    }
34
35 }
```



Ciclo do -while:

A diferencia del while, esta estructura primero ejecuta el conjunto de instrucciones y después verifica que la condición se cumpla.

¿Cómo funciona?

FO-ACA-11

Versión 1

Fecha: 25/10/2018

Cualquier documento no identificado como **Controlado** se considera **COPIA NO CONTROLADA** y no es auditable.

- Realiza es bloque de código que se encuentra en DO
- Después verifica si la condición se cumple
- Si es verdadera, repite el ciclo
- Si es falsa, entonces el ciclo termina.

Ejemplo 1 "Impresión de un valor 3 veces en un ciclo do-while":

Este ciclo se repetirá las veces necesarias hasta cumplir la condición del while, que en este caso es un menor que 3 entonces cuando el contador sea igual con 3 este ciclo saldrá y continuara con las siguientes líneas.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int i=0;
6     do{
7         printf("valor de i = %d \n",i);
8         i++;
9     }while(i<3);
10
11     printf("Fin\n");
12
13
14     system("Pause");
15     return 0;
16 }
```

C:\Users\Dotae\_nbywxbb\Documents\Projects\_c\ejemploDoWhile.exe

```
valor de i = 0
valor de i = 1
valor de i = 2
Fin
Press any key to continue . . .
```

## ARREGLOS

¿Qué es un arreglo?

Variable que hace referencia a varias posiciones de memoria.

Cada posición se identifica con un índice.

El índice comienza en 0.

Declaración:

```
int main (){
    int miArreglo[8];
    system("Pause");
    return 0;
} // Fin main
```

Corchetes para  
indicar que es  
arreglo.

Ejercicio 1 “

- Crear un programa que declare un arreglo llamado "vector" de 10 posiciones.
- Asignar el valor de 10 a cada posición del arreglo.
- Mostrar todas las posiciones del arreglo”:

Mediante ciclos while hicimos el llenado de cada posición del arreglo hasta 10, de esa misma manera mandamos a imprimir cada posición del arreglo agregándole un salto de línea para mostrar aparte cada posición.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int vector[10];
6     int x=0,i=0;
7
8     while (i<10){
9         vector[i]=10;
10        i++;
11    }
12
13    while (x<10){
14        printf("%d \n",vector[x]);
15        x++;
16    }
17
18    system("Pause");
19    return 0;
20 }
```

C:\Users\Dotae\_nbywxbb\Documents\Projects\_c\vector.exe

10  
10  
10  
10  
10  
10  
10  
10  
10  
10  
Press any key to continue . . .

## Ejercicio 2

- Crear un arreglo de 100 posiciones.
- Llenar el arreglo con la tabla del 2.
- Mostrar el arreglo en pantalla”:

En esta ocasión hicimos lo mismo, pero aumentando el tamaño del arreglo y en esta ocasión lo llenamos con la tabla de 2, esto se realizo con un acumulador que a cada posición que guardaba le hacia un postincremento de 2 unidades.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int arreglo[100];
6     int i=0,x=0,y=0;
7
8     while(i<=99){
9         arreglo[i]=y+=2;
10        i++;
11    }
12
13    while(x<=99){
14        printf("%d\n",arreglo[x]);
15        x++;
16    }
17
18    system("Pause");
19    return 0;
20 }
```

C:\Users\Dotae\_nbywxbb\Documents\Projects\_c\arreglo.exe

2  
4  
6  
8  
10  
12  
14  
16  
18  
20  
22  
24  
26  
28  
30  
32  
34  
36  
38

## Ejercicio 3”

- Crear un arreglo de 100 posiciones.
- Llenar el orden inverso al índice.
- Mostrar el arreglo en pantalla”:

Para este caso ocupamos el mismo código base del programa anterior, con la diferencia de que ahora nuestro contador lo inicializamos en 100 y le fuimos haciendo postdecrementos para ir llenando cada posición con un numero menor.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int arreglo[100];
6     int i=0,x=100,y=0;
7
8     while(i<=99){
9         arreglo[i]=y++;
10        i++;
11    }
12
13    while(x>=0){
14        printf("%d\n",arreglo[x]);
15        x--;
16    }
17
18    system("Pause");
19    return 0;
20 }
21
```

## Ejercicio 4 “

Dados:

A = [3, 5, 6, 8, 4, 7, 8, 5, 3, 1]

B = [3, 4, 6, 8, 9, 1, 2, 3, 0, 9]

Realizar las siguientes operaciones:

- $A[3] \bmod (B[2]/2)$
- $B[A[1]] - A[9]$
- $A[0] + A[1+2]$
- $A[5] + B[5]$
- $(A[3]/B[2])/2$ :

En este programa no se utilizaron ciclos, pues en el solo es importante ver el orden de los operadores, tratar de comprender que operaciones se realizan primero y cuales al final.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int a[11]={3,5,6,8,4,7,8,5,3,1};
6     int b[11]={3,4,6,8,9,1,2,3,0,9};
7
8     printf("%d \n",a[3]%(b[2]/2));
9     printf("%d \n",b[a[1]]-a[9]);
10    printf("%d \n",a[0]+a[1+2]);
11    printf("%d \n",a[5]+b[5]);
12    printf("%d \n",a[3]/b[2]/2);
13
14
15    system("Pause");
16    return 0;
17 }
```

## Ejercicio 5 “

- Crea un arreglo de 20 posiciones.
- Asígnale a cada elemento un valor.
- Calcula el promedio de todos los elementos.
- Calcula la multiplicación de todos los elementos”:

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int a[21]={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20};
6     int mul;
7     double prom;
8
9     prom=(a[0]+a[1]+a[2]+a[3]+a[4]+a[5]+a[6]+a[7]+a[8]+a[9]+a[10]+a[11]+a[12]+a[13]+a[14]+a[15]+a[16]+a[17]+a[18]+a[19])/20;
10    mul=a[0]*a[1]*a[2]*a[3]*a[4]*a[5]*a[6]*a[7]*a[8]*a[9]*a[10]*a[11]*a[12]*a[13]*a[14]*a[15]*a[16]*a[17]*a[18]*a[19];
11    printf("Promedio de los numeros: %.2f \n",prom);
12    printf("Multiplicacion de los numeros: %d \n",mul);
13
14    system("Pause");
15    return 0;
16 }

```

C:\Users\Dotae\_nbywxbb\Documents\Projects\_c\ejerciciomultdiv.exe

Promedio de los numeros: 10.00  
Multiplicacion de los numeros: -2102132736  
Press any key to continue . . .

## CICLO FOR:

Ejemplo 1 “Mostrar la serie: 2, 3, 4, 6, 6, 9, 8, 12, 10...”:

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main (){
5     int longitudSerie = 50;
6     int i;
7
8     for (i = 1; i<=(longitudSerie/2); i++){
9         printf("%d, ", 2*i);
10        printf("%d, ", 3*i);
11    }
12
13 }

```

C:\Users\Dotae\_nbywxbb\Documents\Projects\_c\ejemplos\ejemplo\_uno\_for.exe

2, 3, 4, 6, 6, 9, 8, 12, 10, 15, 12, 18, 14, 21, 16, 24, 18, 27, 20, 30, 22, 33, 24, 36, 26, 39, 28, 42, 30, 45, 32, 48, ^  
34, 51, 36, 54, 38, 57, 40, 60, 42, 63, 44, 66, 46, 69, 48, 72, 50, 75, Press any key to continue . . .

## Ejercicio1 “

Escribe un programa que reciba un número N del usuario y haga la suma de todos los números desde 1 hasta N.

Ej. >> 5                      1+2+3+4+5 = 15

Escriba un programa en C que utilice un ciclo para producir la siguiente tabla de valores:

A	A+2	A+4	A+6
3	5	7	9
6	8	10	12
9	11	13	15
12	14	16	18

Para este programa se realizo en dos partes:

En la primer parte tenemos dos variables, una que guardara el valor que el usuario ingrese y otra que funcionara como acumulador e imprimirá números hasta llegar al que ingreso el usuario y al final sumara todos estos números.

En la segunda parte el usuario debe ingresar 4 numeros, a estos números se les hacen algunas sumas para poder cumplir con lo indicado por el ejercicio. Eso podría estar dentro de un ciclo, pero se decidió hacer de esta manera.

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int in,i,uno,dos,tres,cuatro,o,u;
6
7     puts("Ingrese un valor");
8     scanf("%d",&in);
9     for(i=1;i<=in;i++){
10         printf("%d ",i);
11         u=i+i;
12     }
13     printf("\n");
14
15     puts("Ingrese valores;");
16     scanf("%d",&uno);
17     scanf("%d",&dos);
18     scanf("%d",&tres);
19     scanf("%d",&cuatro);
20
21     puts("A      A+2    A+4    A+6");
22     for(o=0;o<1;o++){
23         printf("%d      %d      %d      %d\n",uno,uno+2,uno+4,uno+6);
24         printf("%d      %d      %d      %d\n",dos,dos+2,dos+4,dos+6);
25         printf("%d      %d      %d      %d\n",tres,tres+2,tres+4,tres+6);
26         printf("%d      %d      %d      %d\n",cuatro,cuatro+2,cuatro+4,cuatro+6);
27     }
28
29     system("Pause");
30     return 0;
31 }

```

C:\Users\Dotae\_nbywxb\Documents\Projects\_c\ciclofor.exe

Ingrese un valor  
5  
+1 +2 +3 +4 +5 =10  
Ingrese valores;  
3  
4  
5  
6  
A A+2 A+4 A+6  
3 5 7 9  
4 6 8 10  
5 7 9 11  
6 8 10 12  
Press any key to continue . . .

## FOR ANIDADO:

### Ejemplo 1:

Un for anidado consta de un ciclo dentro de otro para poder realizar diferentes operaciones que suelen requerir de un orden. Se uso el (\t) para dar un espacio al frente de lo impreso y poder agregar algo mas que deseemos.

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int i,j;
6
7     for(i=0;i<5;i++){
8         printf("para i: %d \t",i);
9         for(j=0;j<3;j++){
10             printf("j%d,",j);
11         }
12         printf("\n\n");
13     }
14
15     system("Pause");
16     return 0;
17 }

```

C:\Users\Dotae\_nbywxb\Documents\Projects\_c\ej1.exe

para i: 0 j0,j1,j2,  
para i: 1 j0,j1,j2,  
para i: 2 j0,j1,j2,  
para i: 3 j0,j1,j2,  
para i: 4 j0,j1,j2,  
Press any key to continue . . .

Ejercicio 1 “Escribe un programa que reciba un número entero N del usuario e imprima una escalinata de N pisos de asteriscos”:

```

*
**
***
****

```

Al igual que en el programa anterior se debe de ingresar un numero y en base a eso el



programa realizara impresiones hasta cumplir con la condición del ciclo, el cual se terminara una vez que el valor sea igual al ingresado por el usuario.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int i;
6     int j;
7     int k;
8     int g=0;
9
10    printf("Ingrese un numero: \n");
11    scanf("%d",&k);
12
13    for(i=0;i<k;i++){
14        printf("\t");
15        for(j=0;j<i;j++){
16            printf("\t");
17        }
18        for(g=0;g<k;g++){
19            printf(" ");
20        }
21        for(j=i;j<k;j++){
22            printf("\t");
23        }
24        printf("\n\n");
25    }
26
27    system("Pause");
28    return 0;
29 }
```

C:\Users\Dotae\_nbywxbb\Documents\Projects\_c\for\_anidado\_uno.exe

Ingrese un numero:  
5

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

Press any key to continue . . .

Ejercicio 2 “Haz un programa que muestre las tablas de multiplicar como se muestra a continuación.

- M y N son ingresadas por el usuario”:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int i;
6     int j;
7     int m;
8     int n;
9     int a;
10
11    puts("Ingrese m:");
12    scanf("%d",&m);
13    puts("Ingrese n:");
14    scanf("%d",&n);
15
16    for(i=0;i<=m;i++){
17        for(j=0;j<=n;j++){
18            a=i*j;
19            printf(" %d\t",a);
20        }
21        printf("\n\n");
22    }
23
24    system("Pause");
25    return 0;
26
27 }
```

C:\Users\Dotae\_nbywxbb\Documents\Projects\_c\for\_anidado\_dos.exe

Ingrese m:  
5

Ingrese n:  
5

0	0	0	0	0	0
0	1	2	3	4	5
0	2	4	6	8	10
0	3	6	9	12	15
0	4	8	12	16	20
0	5	10	15	20	25

Press any key to continue . . .