	Estructuras de Repetición	
Facultad de Ingeniería	Laboratorio de docencia	

Laboratorios de computación
salas A y B

Profesor: Alejandro Pimentel

Asignatura: Fundamentos de Programación

Grupo: 135

No de Práctica(s): 9

Integrante(s): Lorena Basurto Amezcu

*No. de Equipo de
cómputo empleado:*

No. de Lista o Brigada: 2858

Semestre: 2020-1

Fecha de entrega: Octubre 14, 2019.

Observaciones: Tarde entrega.
Pude ver que si pusiste un #define.
Pero nunca lo utilizaste para el procedimiento.

CALIFICACIÓN: 7

Objetivo:

Elaborar programas en C para la resolución de problemas básicos que incluyan las estructuras de repetición y la directiva *define*.

Desarrollo:

- While

```
while (expresión_lógica) {  
    // Bloque de código a repetir  
    // mientras que la expresión  
    // lógica sea verdadera.  
}
```

- Do-While

```
do {  
    /*  
    Bloque de código que se ejecuta  
    por lo menos una vez y se repite  
    mientras la expresión lógica sea  
    verdadera.  
    */  
} while (expresión_lógica);
```

- For

```
for (inicialización ; expresión_lógica ; operaciones por iteración) {  
    /*  
        Bloque de código  
        a ejecutar  
    */  
}
```

- Define

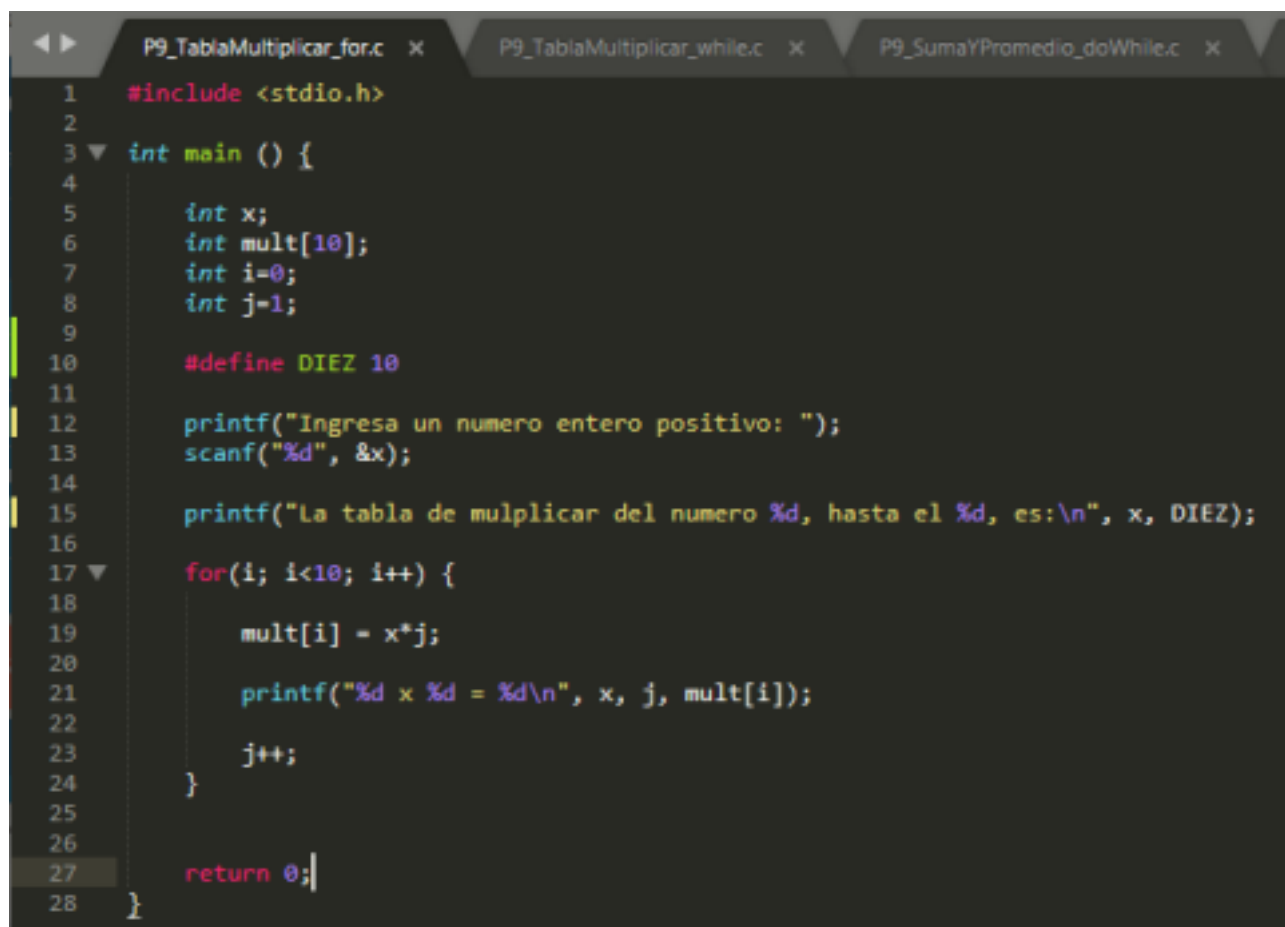
El *define* es una palabra clave que se utiliza para declarar un nombre especial con un significado. Es muy parecido a una variable, con la diferencia de que no se puede cambiar a lo largo del programa.

```
#define MAX 5
```

- **Actividades**

Para cada uno de los siguientes problemas, elegir un tipo de ciclo y resolverlo. Al final, deben usar los tres tipos de ciclos y usar *define* por lo menos una vez.

- Hacer un programa que pida un número y muestre su tabla de multiplicar (hasta el 10).
 - Hacer un programa que pida y lea 10 números y muestre su suma y su promedio.
 - Hacer un programa que pida un número e indique si es primo o no.
-
- Programa que pida un número y muestre su tabla de multiplicar hasta el 10.



```
1  #include <stdio.h>
2
3  int main () {
4
5      int x;
6      int mult[10];
7      int i=0;
8      int j=1;
9
10     #define DIEZ 10
11
12     printf("Ingresa un numero entero positivo: ");
13     scanf("%d", &x);
14
15     printf("La tabla de mulplicar del numero %d, hasta el %d, es:\n", x, DIEZ);
16
17     for(i; i<10; i++) {
18
19         mult[i] = x*j;
20
21         printf("%d x %d = %d\n", x, j, mult[i]);
22
23         j++;
24     }
25
26
27     return 0;
28 }
```

```
P9_TablaMultiplicar_for.c x P9_TablaMultiplicar_while.c x P9_SumaYPromedio_doWhile.c x
1  #include <stdio.h>
2
3  int main () {
4
5      int x;
6      int mult[10];
7      int i=0;
8      int j=1;
9
10     #define DIEZ 10
11
12     printf("Ingresa un numero entero positivo.\n");
13     scanf("%d", &x);
14
15     printf("La tabla de mulplicar del numero %d, hasta el %d, es:\n", x, DIEZ);
16
17     while(i<10) {
18
19         mult[i] = x*j;
20
21         printf("%d x %d = %d\n", x, j, mult[i]);
22
23         i++;
24         j++;
25     }
26
27
28     return 0;
29 }
```

```
Ingresa un numero entero positivo.
5
La tabla de mulplicar del numero 5, hasta el 10, es:
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
```

```
Ingresa un numero entero positivo.
5
La tabla de mulplicar del numero 5, hasta el 10, es:
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
```

```

Ingresa un numero entero positivo.
33
La tabla de mulplicar del numero 33, hasta el 10, es:
33 x 1 = 33
33 x 2 = 66
33 x 3 = 99
33 x 4 = 132
33 x 5 = 165
33 x 6 = 198
33 x 7 = 231
33 x 8 = 264
33 x 9 = 297
33 x 10 = 330

```

- Programa que pida diez números y muestre su suma y su promedio.

```

P9_TablaMultiplicar_for.c  P9_SumaYPromedio_doWhile.c  P9_TablaMultiplicar_while.c  untitled
1  #include <stdio.h>
2
3  int main () {
4
5      double n1, n2, n3, n4, n5, n6, n7, n8, n9, n10;
6      double Suma = 0;
7      double Promedio;
8      double numeros[10];
9      int i = 0;
10     int j = 0;
11
12     do {
13
14         printf("Ingresa un numero: ");
15         scanf("%lf", &numeros[i]);
16         i++;
17
18     } while(i<10);
19
20     do {
21
22         Suma = Suma + numeros[j];
23         j++;
24
25     } while(j<10);
26
27     Promedio = Suma / 10;
28
29     printf("La suma de los diez numeros es %.2f y el promedio es %.2f", Suma, Promedio);
30
31
32     return 0;
33 }

```

```

Ingresa un numero: 2
Ingresa un numero: 4
Ingresa un numero: 6
Ingresa un numero: 8
Ingresa un numero: 10
Ingresa un numero: 12
Ingresa un numero: 14
Ingresa un numero: 16
Ingresa un numero: 18
Ingresa un numero: 20
La suma de los diez numeros es 110.00 y el promedio es 11.00

```

```

Ingresa un numero: 1
Ingresa un numero: -1
Ingresa un numero: 7
Ingresa un numero: -7
Ingresa un numero: 36
Ingresa un numero: -36
Ingresa un numero: 123456
Ingresa un numero: -123456
Ingresa un numero: 0
Ingresa un numero: 100
La suma de los diez numeros es 100.00 y el promedio es 10.00

```

- Programa que pida un número e indique si es primo o no.

```

P9_TablaMultiplicar_for.c  P9_SumaYPromedio_doWhile.c x
1  #include <stdio.h>
2
3  int main() {
4
5      int n;
6      int x=1;
7      int i=0;
8      int j=0;
9
10     printf("Escribe un numero entero:\n");
11     scanf("%i", &n);
12
13     while(i<=n) {
14
15         i++;
16
17         if( n % i == 0 ) {
18             j++;
19         }
20     }
21
22     if(j==2) {
23         printf("El numero %d es primo.\n", n);
24     }
25     else {
26         printf("El numero %d no es primo.\n", n);
27     }
28
29
30     return 0;
31 }
32

```

```

Escribe un numero entero:
0
El numero 0 no es primo.

```

```

Escribe un numero entero:
1
El numero 1 no es primo.

```

```
Escribe un numero entero:  
2  
El numero 2 es primo.
```

```
Escribe un numero entero:  
3  
El numero 3 es primo.
```

```
Escribe un numero entero:  
4  
El numero 4 no es primo.
```

```
Escribe un numero entero:  
5  
El numero 5 es primo.
```

```
Escribe un numero entero:  
7  
El numero 7 es primo.
```

```
Escribe un numero entero:  
20  
El numero 20 no es primo.
```

```
Escribe un numero entero:  
97  
El numero 97 es primo.
```

Conclusiones:

Las estructuras de repetición *while*, *do while* y *for* en el lenguaje C, son muy útiles en la resolución de problemas básicos. También el comando *define* simplifica el trabajo al asignar un valor que no cambiará y que se puede utilizar a lo largo de todo un programa.