

## TRABAJO PRACTICO N1 - C+

EJERCICIO 1: determinar si algún número está repetido.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <stdbool.h>
4
5  int main()
6  {
7
8      int A, B, C, rep;
9
10     printf ("Ingrese tres numeros naturales: ");
11     scanf ("%d %d %d", &A, &B, &C);
12     if (A==B||A==C||B==C)
13         rep=true;
14     else
15         rep=false;
16     if (rep==true)
17         printf("Al menos uno está repetido.");
18     else
19         printf("No hay numeros repetidos.");
20     return 0;
21 }
```

```
"C:\Users\danag\OneDrive\Es  x  +  v
Ingrese tres numeros naturales: 4 8 6
No hay numeros repetidos.
Process returned 0 (0x0)  execution time : 5.760 s
Press any key to continue.
```

EJERCICIO 2: determinar si los valores son crecientes y equidistantes.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <stdbool.h>
4
5  int main()
6  {
7      int A, B, C, D, Cy;
8      printf("Ingrese cuatro numeros naturales: ");
9      scanf ("%d %d %d %d", &A, &B, &C, &D);
10     if ((A<B) && (B<C) && (C<D))
11         if ((B-A==C-B) && (C-B==D-C))
12             Cy=true;
13         else
14             Cy=false;
15     else
16         Cy=false;
17     if (Cy==true)
18         printf("Los numeros son crecientes y equidistantes.");
19     else
20         printf("Los numeros no cumplen al menos una condicion.");
21     return 0;
22 }
23
24
25
```

```
"C:\Users\danag\OneDrive\Es  x  +  v
Ingrese cuatro numeros naturales: 15 30 45 30
Los numeros no cumplen al menos una condicion.
Process returned 0 (0x0)  execution time : 5.840 s
Press any key to continue.
```

EJERCICIO 3

A) Calcular la suma de dos números.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int a, b, suma;
7
8      printf("Ingrese los numeros a sumar: ");
9      scanf ("%d %d", &a, &b);
10     suma = a+b;
11     printf("La suma es de %d", suma);
12
13     return 0;
14 }
15
```

```
"C:\Users\danag\OneDrive\Es  x  +  v
Ingrese los numeros a sumar: 15 10
La suma es de 25
Process returned 0 (0x0)  execution time : 9.205 s
Press any key to continue.
```

B) Calcular el 30% de un valor.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int x, porc;
7      printf("Ingrese el monto: ");
8      scanf("%d", &x);
9      porc = 0.30 * x;
10     printf("El 30 por ciento del monto es %d", porc);
11
12     return 0;
13 }

```

```

C:\Users\danag\OneDrive\Es x + v
Ingrese el monto: 500
El 30 por ciento del monto es 150
Process returned 0 (0x0)   execution time : 5.646 s
Press any key to continue.

```

C) Devolver num1 si num1>num2, num2 si num2>num1 y si son iguales 0.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int num1, num2, cond;
7      printf("Ingrese dos numeros: ");
8      scanf("%d%d", &num1, &num2);
9      if (num1>num2)
10         cond = num1 - num2;
11     else
12         if (num1<num2)
13             cond = num2 - num1;
14         else
15             cond = 0;
16     printf("%d", cond);
17     return 0;
18 }

```

```

C:\Users\danag\OneDrive\Es x + v
Ingrese dos numeros: 15 8
7
Process returned 0 (0x0)   execution time : 4.172 s
Press any key to continue.

```

D) Convertir distancia en pies.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int Dm, Dp, Dpul;
7      printf("Ingrese el valor en metros: ");
8      scanf ("%d", &Dm);
9      Dpul = 39.37 * Dm;
10     Dp = Dpul/12;
11
12     printf("La distancia son %d pies.", Dp);
13
14     return 0;
15 }

```

```

C:\Users\danag\OneDrive\Es x + v
Ingrese el valor en metros: 1000
La distancia son 3280 pies.
Process returned 0 (0x0)   execution time : 5.618 s
Press any key to continue.

```

**EJERCICIO 4:** cantidad de billetes de cada nominación que debo dar si quiero utilizar la menor cantidad de billetes posible.

```

C:\Users\danag\OneDrive\Es x + v
Ingrese el costo y el pago del objeto: 415 500

La cantidad de billetes de 50 es 1
La cantidad de billetes de 25 es 1
La cantidad de billetes de 10 es 1
La cantidad de billetes de 5 es 0
La cantidad de billetes de 1 es 0
Process returned 0 (0x0)   execution time : 3.123 s
Press any key to continue.

```

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int p, c, vuelto;
7      printf("Ingrese el costo y el pago del objeto: ");
8      scanf("%d%d", &c, &p);
9      vuelto = p - c; //vuelto=132
10     printf("\nLa cantidad de billetes de 50 es %d", vuelto/50);
11     vuelto = vuelto%50; //resto 32
12     printf("\nLa cantidad de billetes de 25 es %d", vuelto/25);
13     vuelto = vuelto%25; //resto 7
14     printf("\nLa cantidad de billetes de 10 es %d", vuelto/10);
15     vuelto = vuelto%10; //resto 7
16     printf("\nLa cantidad de billetes de 5 es %d", vuelto/5);
17     vuelto = vuelto%5; //resto 2
18     printf("\nLa cantidad de billetes de 1 es %d", vuelto/1);
19     vuelto = vuelto%1; //resto 0
20
21     return 0;
22 }

```

## EJERCICIO 6 (FOR)

### A) Producto de los primeros N naturales pares.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int N, producto, par, i;
7      printf("Ingrese un numero natural: "); //cantidad de numeros naturales pares que se van a multiplicar
8      scanf("%d", &N);
9      par=1; //Inicializo par
10     producto=1;
11
12     for (i=1; i<=N; i++)
13     {
14         par = 2*i;
15         producto = producto*par;
16     }
17
18     printf("\nEl producto de los primeros %d naturales pares es: %d", N, producto);
19     return 0;
20 }
21
22 //N es la cantidad de pares que usa siempre empezando en 2

```

```

C:\Users\danag\OneDrive\Es x + v
Ingrese un numero natural: 5

El producto de los primeros 5 naturales pares es: 3840
Process returned 0 (0x0)   execution time : 0.979 s
Press any key to continue.

```

### B) Primeros N múltiplos de K.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int N, k, producto, i, multiplik;
7      printf("Ingrese dos numeros enteros: "); //N es la cantidad de multiplos y k es el valor del cual se parte
8      scanf("%d%d", &N, &k);
9      producto = 1; //Inicializo producto
10
11     for(i=1; i<=N; i++)
12     {
13         multiplik = k*i;
14         printf("%d ", multiplik); //Poner %d con un espacio para que me muestre los numeros separados
15     }
16
17     return 0;
18 }

```

```

C:\Users\danag\OneDrive\Es x + v
Ingrese dos numeros enteros: 8 6
6 12 18 24 30 36 42 48
Process returned 0 (0x0)   execution time : 6.237 s
Press any key to continue.

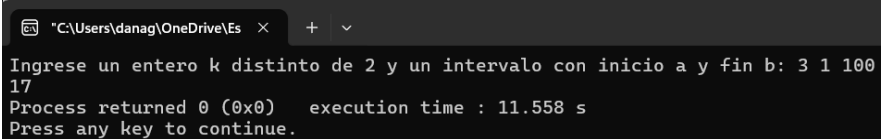
```

### C) Contar cuántos enteros en el intervalo (a, b) son múltiplos de K y no de 2.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int k, a, b, cantidad, i;
7      printf("Ingrese un entero k distinto de 2 y un intervalo con inicio a y fin b: ");
8      scanf("%d%d%d", &k, &a, &b);
9      cantidad = 0;
10
11     for(i=a; i<=b; i++)
12     {
13         if((i%k==0)&&!(i%2==0))
14             cantidad = cantidad + 1;
15     }
16
17     printf("%d ", cantidad);
18
19     return 0;
20 }
21

```

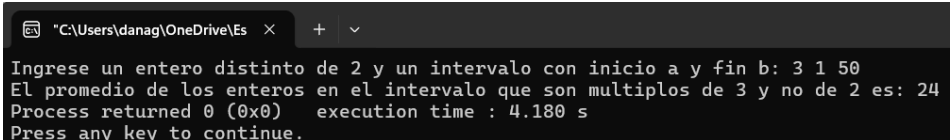


D) Promedio de los enteros en el intervalo [a, b] que son múltiplos de K y no de 2.

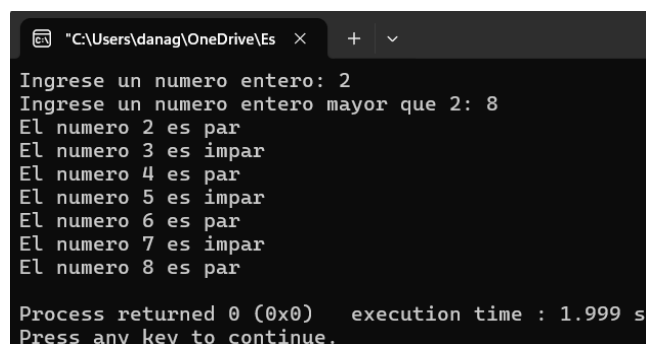
```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int k, a, b, promedio, suma, cantidad, i;
7      printf("Ingrese un entero distinto de 2 y un intervalo con inicio a y fin b: ");
8      scanf("%d%d%d", &k, &a, &b);
9
10     cantidad = 0; suma = 0; promedio = 1;
11
12     for(i=a; i<=b; i++) //Va solo menor a b porque es un intervalo CERRADO
13     {
14         if((i%k==0)&&!(i%2==0)){
15             suma = suma + i;
16             cantidad = cantidad + 1;
17         }
18     }
19
20     promedio = suma/cantidad;
21     printf("El promedio de los enteros en el intervalo que son multiplos de %d y no de 2 es: %d", k, promedio);
22
23     return 0;
24 }
25

```



EJERCICIO 7: Mostrar para cada valor en el intervalo [a, b] si el valor es par o impar.



```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int a, b, i;
7
8      printf("Ingrese un numero entero: ");
9      scanf("%d", &a);
10     printf("Ingrese un numero entero mayor que %d: ", a);
11     scanf("%d", &b);
12
13     if(b<a)
14     {
15         printf("Le pedi un numero entero mayor a %d!", a);
16         //Si se da este caso, no vuelve a pedir que ingrese un numero correcto (sin iteracion)
17     }
18     else
19     {
20         for(i=a; i<=b; i++)
21         {
22             if(i%2==0)
23             {
24                 printf("El numero %d es par\n", i);
25             }
26             else
27             {
28                 printf("El numero %d es impar\n", i);
29             }
30         }
31     }
32     return 0;
33 }

```

#### EJERCICIO 8: Divisores de un número natural N.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int N, i;
7
8      printf("Ingrese un numero natural: ");
9      scanf("%d", &N);
10
11     if (N<=0)
12         printf("Le pedi un numero natural!");
13     else
14     {
15         printf("Los divisores de %d son: ", N);
16         for (i=1; i<=N/2; i++)
17             if(N%i==0)
18                 printf("%d ", i);
19     }
20
21     return 0;
22 }

```

```

"C:\Users\danag\OneDrive\Es x + v
Ingrese un numero natural: 48
Los divisores de 48 son: 1 2 3 4 6 8 12 16 24
Process returned 0 (0x0)   execution time : 7.142 s
Press any key to continue.

```

**EJERCICIO 9:** Preguntar cuantos números va a introducir, pedir esos números e indicar cuántos fueron pares y cuando impares.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int cantnumeros, cantpares, cantimpares, valor, i;
7      printf("Especifique cuantos numeros va a introducir: ");
8      scanf("%d", &cantnumeros);
9
10     cantpares = 0; cantimpares = 0;
11
12     if(cantnumeros<0)
13         printf("Imposible!");
14     else
15     {
16         for(i=1; i<=cantnumeros; i++)
17         {
18             printf("Escriba el valor %i: ", i);
19             scanf("%i", &valor);
20             if (valor%2==0)
21                 cantpares = cantpares + 1;
22             else
23                 cantimpares = cantimpares + 1;
24         }
25         printf("Ha escrito %d numeros pares y %d numeros impares.", cantpares, cantimpares);
26     }
27     return 0;
28 }

```

```

"C:\Users\danag\OneDrive\Es x + v
Especifique cuantos numeros va a introducir: 4
Escriba el valor 1: 15
Escriba el valor 2: 8
Escriba el valor 3: 7
Escriba el valor 4: 2
Ha escrito 2 numeros pares y 2 numeros impares.
Process returned 0 (0x0)   execution time : 7.614 s
Press any key to continue.

```

**EJERCICIO 10:** Preguntar cuantos números va a introducir, pedir esos números y mostrar el mayor, el menor y la media.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int cantnumeros, mayor, menor, media, i, suma, valor;
7      printf("Especifique cuantos numeros va a introducir: ");
8      scanf("%d", &cantnumeros);
9
10     suma = 0;
11
12     if(cantnumeros<0)
13     {
14         printf("Imposible!");
15     }
16     else
17     {
18         printf("Escriba el valor 1: ");           //Escribo el primer valor
19         scanf("%i", &valor);
20         menor = valor; mayor = valor; suma = valor;
21     }
22     {
23         for(i=2; i<=cantnumeros; i++)
24         {
25             printf("Escriba el valor %i: ", i);   //Escribo los valores siguientes 2 3 4 5
26             scanf("%i", &valor);
27             if (valor>mayor)
28                 {mayor = valor;}
29             if (valor<menor)
30                 {menor = valor;}
31             suma = suma + valor;
32         }
33     }
34     media = suma/cantnumeros;
35     printf("\nEl menor numero es: %d\n", menor);
36     printf("El mayor numero es: %d\n", mayor);
37     printf("La media es: %d\n", media);
38 }
39
40 return 0;

```

```

"C:\Users\danag\OneDrive\Es x + v
Especifique cuantos numeros va a introducir: 4
Escriba el valor 1: 15
Escriba el valor 2: 10
Escriba el valor 3: 8
Escriba el valor 4: 2

El menor numero es: 2
El mayor numero es: 15
La media es: 8

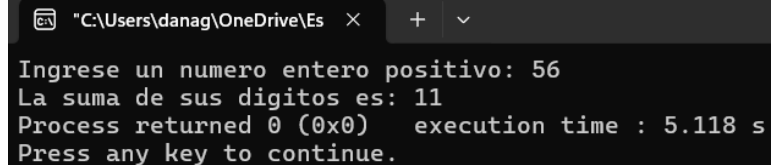
Process returned 0 (0x0)   execution time : 4.734 s
Press any key to continue.

```

## EJERCICIO 11 (WHILE)

A) Sumar los dígitos de un número entero positivo N.

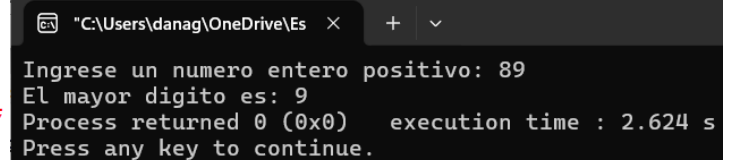
```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int N, SumaDig;
7      printf("Ingrese un numero entero positivo: ");
8      scanf("%i", &N);
9
10     SumaDig = 0;
11
12     while (N>0)
13     {
14         SumaDig = SumaDig + (N%10);
15         N = N/10;
16     }
17     printf("La suma de sus digitos es: %d", SumaDig);
18
19     return 0;
20 }
21
```



```
"C:\Users\danag\OneDrive\Es x + v
Ingrese un numero entero positivo: 56
La suma de sus digitos es: 11
Process returned 0 (0x0) execution time : 5.118 s
Press any key to continue.
```

B) Encontrar el mayor dígito en N.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int N, MayorDig, Digito;
7      printf("Ingrese un numero entero positivo: ");
8      scanf("%i", &N);
9
10     MayorDig = 0;
11
12     while (N>0)
13     {
14         {
15             Digito = N%10; //Recupero el ultimo digito y lo guardo
16             N = N/10; //Le saco ese digito a N para seguir con los otros
17         }
18         if(Digito>MayorDig) //Proceso el digito (en este caso veo si es el mayor)
19         {MayorDig = Digito;}
20     }
21     printf("El mayor digito es: %d", MayorDig); //VER QUE PASA CUANDO N<0, EL MAYOR DIGITO ES CERO
22
23     return 0;
24 }
```



```
"C:\Users\danag\OneDrive\Es x + v
Ingrese un numero entero positivo: 89
El mayor digito es: 9
Process returned 0 (0x0) execution time : 2.624 s
Press any key to continue.
```

C) Determinar si un dígito D está presente en N.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <stdbool.h>
4
5  int main()
6  {
7      int N, Dig;
8      bool EstaDig;
9      printf("Ingrese un numero entero positivo de mas de 1 cifra: ");
10     scanf("%i", &N);
11     printf("Ingrese un numero: ");
12     scanf("%i", &Dig);
13
14     EstaDig = false;
15
16     while ((EstaDig == false) && (N>0))
17     {
18         if(Dig == (N%10))
19         {
20             EstaDig = true;
21         }
22         else
23         {
24             N = N/10;
25         }
26     }
27     if (EstaDig == true)
28     {
29         printf("El digito %d esta presente en el numero.", Dig);
30     }
31     else
32     {
33         printf("El digito %d NO esta presente en el numero.", Dig);
34     }
35     return 0;
36 }

```

```

"C:\Users\danag\OneDrive\Es  x + v
Ingrese un numero entero positivo de mas de 1 cifra: 456
Ingrese un numero: 2
El digito 2 NO esta presente en el numero.
Process returned 0 (0x0)   execution time : 3.190 s
Press any key to continue.

```

D) Contar cuántos dígitos impares tiene N.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int N, DigImpares;
7      printf("Ingrese un numero entero positivo: ");
8      scanf ("%d", &N);
9
10     DigImpares = 0;
11
12     while(N>0)
13     {
14         if (!(N%10)%2 == 0)    //Controlo el ultimo digito y veo si son impares
15         {
16             DigImpares = DigImpares + 1;    //Cuento digitos
17         }
18         N = N/10;    //Saco el ultimo digito
19     }
20     printf("El numero tiene %d digito/s impar/es.", DigImpares);
21     return 0;
22 }

```

```

"C:\Users\danag\OneDrive\Es  x + v
Ingrese un numero entero positivo: 455
El numero tiene 2 digito/s impar/es.
Process returned 0 (0x0)   execution time : 3.024 s
Press any key to continue.

```

E) Promedio de los dígitos impares de N.



```

1  #include <stdio.h>
2  #include <stdlib.h>
3  int main()
4  {
5      int N, cant, suma;
6      float prom;
7
8      do
9      {
10         printf("Ingrese un numero entero positivo: ");
11         scanf("%d", &N);
12     }
13     while (!(N>0));
14     suma = 0;
15     cant = 0;
16     while (N>0)
17     {
18         if(!(N%2 == 0))
19         {
20             suma = suma + N%10;
21             cant = cant + 1;
22         }
23         N = N/10;
24     }
25     if(cant!=0)
26         prom = suma/(float) cant;
27     else
28         prom = 0;
29
30     if (cant>0)
31         printf("El promedio de los digitos impares es: %f", prom);
32     else
33         printf("No habia digitos impares en el numero.");
34
35     return 0;
36 }

```

```

"C:\Users\danag\OneDrive\Es  X  +  v
Ingrese un numero entero positivo: 357
El promedio de los digitos impares es: 5.000000
Process returned 0 (0x0)   execution time : 12.837 s
Press any key to continue.

```

**EJERCICIO 12:** Determinar si un entero positivo N es un número primo.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <stdbool.h>
4
5  int main()
6  {
7      int N, i;
8      bool EsPrimo;           //Los numeros primos solo son divisibles por ellos mismos y por 1
9      printf("Ingrese un numero entero positivo: ");
10     scanf("%d", &N);
11
12     EsPrimo = true;
13     i = 2;                   //Inicializo en 2 porque 4 es el primer primo
14
15     while ((EsPrimo == true)&&(i<=N/2)) //Mientras el numero sea primo y N/2 sea mayor a i=2
16     {
17         if(N%i == 0)         //Si N es multiplo de i?
18         {
19             EsPrimo = false;
20         }
21         i = i+1;
22     }
23
24     if(EsPrimo == true)
25     {
26         printf("El numero %d es primo.", N);
27     }
28     else
29     {
30         printf("El numero %d NO es primo.", N);
31     }
32     return 0;
33 }

```

```

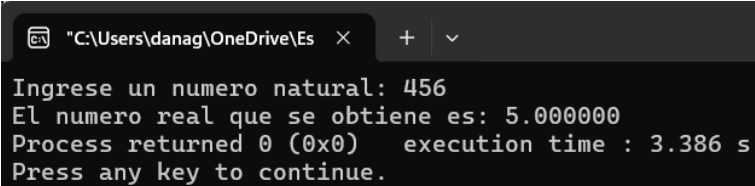
"C:\Users\danag\OneDrive\Es  X  +  v
Ingrese un numero entero positivo: 7
El numero 7 es primo.
Process returned 0 (0x0)   execution time : 1.044 s
Press any key to continue.

```

## PRIMITIVAS

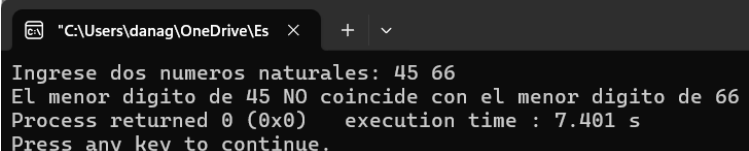
**EJERCICIO 13:** Obtener un número real R como resultado de dividir la suma de los dígitos de un número natural N por la cantidad de dígitos de N.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int SumaDig(int E)           //Primera primitiva: suma de digitos
5  {
6      int suma;
7      suma = 0;
8
9      while(E>0)
10     {
11         suma = suma + E%10; //Acumulo el ultimo digito
12         E = E/10;
13     }
14     return (suma);
15 }
16
17 int CantDig (int L)           //Segunda primitiva: cantidad de digitos
18 {
19     int cant;
20     cant = 0;
21
22     while (L>0)
23     {
24         cant = cant + 1; //Sumo los digitos
25         L = L/10;
26     }
27     return (cant);
28 }
29
30 int main()
31 {
32     int N;
33     float R;
34
35     printf("Ingrese un numero natural: ");
36     scanf("%d", &N);
37
38     R = SumaDig(N)/(float) CantDig(N); //Invoco las primitivas anteriores
39     printf("El numero real que se obtiene es: %lf ", R);
40
41     return 0;
42 }
```



```
"C:\Users\danag\OneDrive\Es  x + v
Ingrese un numero natural: 456
El numero real que se obtiene es: 5.000000
Process returned 0 (0x0)   execution time : 3.386 s
Press any key to continue.
```

**EJERCICIO 14:** Determinar si el menor dígito de un número natural a coincide con el menor dígito de un número natural b.



```
"C:\Users\danag\OneDrive\Es  x + v
Ingrese dos numeros naturales: 45 66
El menor digito de 45 NO coincide con el menor digito de 66
Process returned 0 (0x0)   execution time : 7.401 s
Press any key to continue.
```

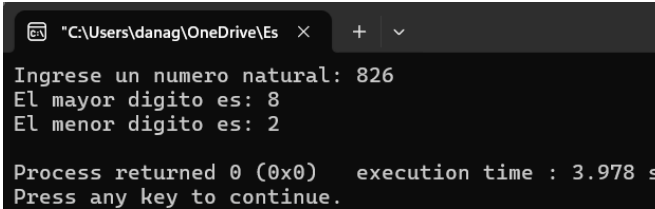
```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <stdbool.h>
4
5  int MenorDigito(int N)
6  {
7      int menor, digito;
8      menor = 10;
9
10     while(N>0)
11     {
12         digito = N%10;
13         N = N/10;
14         if (digito < menor)
15         {
16             menor = digito;
17         }
18     }
19     return(menor);
20 }
21
22 int main()
23 {
24     int a, b;
25     bool igual;
26
27     printf("Ingrese dos numeros naturales: ");
28     scanf ("%d%d", &a, &b);
29
30     if(MenorDigito(a) == MenorDigito(b))          //EN EL IF EL IGUAL ES DOBLE ==
31     {
32         printf("El menor digito de %d coincide con el menor digito de %d", a, b);
33     }
34     else
35     {
36         printf("El menor digito de %d NO coincide con el menor digito de %d", a, b);
37     }
38     return 0;
39 }

```

**EJERCICIO 15:** Devolver el menor y mayor dígito de un número natural N.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int MayorDigito(int M)      //Primera primitiva: mayor digito
5  {
6      int mayor, digito;
7      mayor = 0;
8
9      while(M>0)
10     {
11         digito = M%10;
12         M = M/10;
13         if(digito > mayor)
14         {
15             mayor = digito;
16         }
17     }
18     return (mayor);
19 }
20
21 int MenorDigito (int N)    //Segunda primitiva: menor digito
22 {
23     int menor, digito;
24     menor = 9;             //el mayor digito va a ser 9 porque 10 ya son dos digitos
25
26     while(N>0)
27     {
28         digito = N%10;
29         N = N/10;
30         if(digito < menor)
31         {
32             menor = digito;
33         }
34     }
35     return (menor);
36 }
37
38 int main()
39 {
40     int E, mayor, menor;
41
42     printf("Ingrese un numero natural: ");
43     scanf("%d", &E);
44
45     mayor = MayorDigito(E);
46     printf("El mayor digito es: %d\n", mayor);
47     menor = MenorDigito(E);
48     printf("El menor digito es: %d\n", menor);
49
50     return 0;
51 }
```



```
"C:\Users\danag\OneDrive\Es x + v
Ingrese un numero natural: 826
El mayor digito es: 8
El menor digito es: 2

Process returned 0 (0x0)   execution time : 3.978 s
Press any key to continue.
```

**EJERCICIO 16:** Determinar si la suma de los dígitos de un número N es mayor al producto de los dígitos de M.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <stdbool.h>
4
5
6  int SumaDigitos(int N)
7  {
8      int suma;
9      suma = 0;
10
11     while (N>0)
12     {
13         suma = suma + N%1000;
14         //La cantidad de ceros determina cuantos digitos me lee del numero que ingresa el usuario
15         N = N/10;
16     }
17
18     return (suma);
19 }
20
21
22 int ProductoDigitos(int M)
23 {
24     int producto;
25     producto = 1;
26
27     while (M>0)
28     {
29         producto = producto * M%1000;
30         //La cantidad de ceros determina cuantos digitos me lee del numero que ingresa el usuario
31         M = M/10;
32     }
33
34     return (producto);
35 }
36
37
38 int main()
39 {
40     int N, M;
41     bool mayor;
42
43     printf("Ingrese dos numeros naturales: ");
44     scanf("%d%d", &N, &M);
45     mayor = false;
46     //es true o false?
47     if(SumaDigitos(N) > ProductoDigitos(M))
48     {
49         mayor = true;
50         printf("La suma de los digitos de %d es mayor al producto de los digitos de %d", N, M);
51     }
52     else
53     {
54         mayor = false;
55         printf("La suma de los digitos de %d NO es mayor al producto de los digitos de %d", N, M);
56     }
57
58     return 0;
59 }
60
```

```
"C:\Users\danag\OneDrive\Es  X  +  v
Ingrese dos numeros naturales: 45 69
La suma de los digitos de 45 NO es mayor al producto de los digitos de 69
Process returned 0 (0x0)   execution time : 3.662 s
Press any key to continue.
```

**EJERCICIO 17:** Determinar si la suma de los dígitos de un número natural N es divisible por el mayor de sus dígitos.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <stdbool.h>
4
5  int SumaDigitos (int N)
6  {
7      int suma;
8      suma = 0;
9
10     while (N>0)
11     {
12         suma = suma + N%10;
13         N = N/10;
14     }
15     return (suma);
16 }
17
18 int MayorDigito (int M)
19 {
20     int mayor, digito;
21     mayor = 0;
22
23     while (M>0)
24     {
25         digito = M%10;
26         M = M/10;
27         if (digito > mayor)
28         {
29             mayor = digito;
30         }
31     }
32     return (mayor);
33 }
34
35 int main()
36 {
37     int K;
38     bool EsDivisible;
39
40     printf("Ingrese un numero natural: ");
41     scanf ("%d", &K);
42     EsDivisible = true;
43
44     if (SumaDigitos(K) % MayorDigito(K) == 0)
45     {
46         EsDivisible = true;
47         printf("La suma de los digitos de %d es divisible por el mayor de sus digitos %d", K, MayorDigito(K));
48     }
49     else
50     {
51         EsDivisible = false;
52         printf("La suma de los digitos de %d NO es divisible por el mayor de sus digitos %d", K, MayorDigito(K));
53     }
54
55     return 0;
56 }
```



Ingrese un numero natural: 256  
La suma de los digitos de 256 NO es divisible por el mayor de sus digitos 6  
Process returned 0 (0x0) execution time : 1.983 s  
Press any key to continue.