

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación

PROGRAMACIÓN I

UNIDAD 1. ESTRUCTURAS DE REPETICIÓN



Docente:

Prof.^a. Erika Bonfil Barragán

EQUIPO 8

Jesús Huerta Aguilar		202041509
Javier De La Luz Ruiz		202033810
Ernesto Flores Cesáreo		202066335

Fecha de elaboración:

26/08/2021

NRC: 18438

Sección: 007

SEGUNDO SEMESTRE

Puebla, Pue.

Fecha de entrega: 27/08/2021

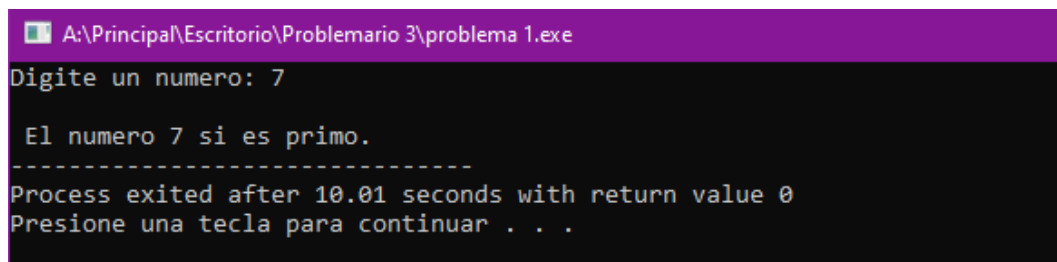
BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA
FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

1. Escriba un programa que determine si un número es primo o no.

CODIGO:

```
1. //Jesús Huerta Aguilar, Javier de La Luz Ruiz, Ernesto Flores Cesareo
2. //Programación I - Numero primo
3. #include <stdio.h>
4.
5. int main(){
6.
7.     int num, i, cont= 0;
8.
9.     printf("Digite un numero: ");
10.    scanf("%i", &num);
11.
12.    for (i=1; i<=num; i++)
13.    {
14.        if (num%i==0)
15.        {
16.            cont++;
17.        }
18.    }
19.
20.    if (cont>2){
21.        printf("\n El numero %d NO es primo.", num);
22.    }
23.    else{
24.        printf("\n El numero %d SI es primo.", num);
25.    }
26.
27.    return 0;
28. }
```

EJECUCIÓN:



```
A:\Principal\Escritorio\Problematario 3\problema 1.exe
Digite un numero: 7

El numero 7 si es primo.
-----
Process exited after 10.01 seconds with return value 0
Presione una tecla para continuar . . .
```

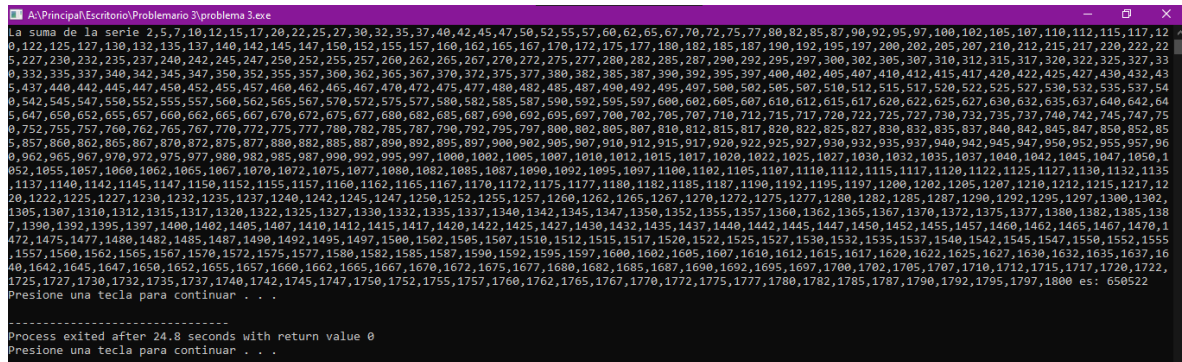
3. Escriba un algoritmo que obtenga la suma e imprima los números de la siguiente serie:

a. 2,5,7,10,12,15,17,...,1800

```
1. //Jesús Huerta Aguilar, Javier de La Luz Ruiz, Ernesto Flores Cesareo
2. //Programación I - "Programa: Algoritmo que suma e imprima los números de
   la serie: 2,5,7,10,12,15,17,...,1800"
3. #include <conio.h>
4. #include <stdio.h>
5. int main(){
6.     //Declarar variables
7.     int band,val,suma;
8.     //Asignar Valores
9.     band = 1;
10.    val=2;
11.    suma = 2;
12.    //Desarrollo de la serie
13.    printf("La suma de la serie ");
14.    do{
15.        //Imprimir valores de la serie
16.        if(val < 1800){
17.            printf("%d,",val);
18.        }
19.        else{
20.            printf("%d",val);
21.        }
22.        //Suma por valores
23.        if (band==1){
24.            val = val + 3;
25.            band = 0;
26.        }
27.        else{
28.            val = val + 2;
29.            band = 1;
30.        }
31.        suma = suma + val;
32.    } while (val <= 1800);
33.    //Resultado de suma de todos los valores
34.    printf(" es: %d\n",suma);
35.    //getch();
36.    system("pause");
37.    return 0;
38. }
```

BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA
FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

EJECUCIÓN:



5. Realice un programa que muestre los primeros n términos de la serie de Fibonacci.
Recuerde que los 2 primeros números son 0 y 1. El resto se calcula como la suma de los 2 números inmediatos que la preceden.
 - a. 0,1,1,2,3,5,8,13,21,34,...

CODIGO:

```
1. //Jesús Huerta Aguilar, Javier de La Luz Ruiz, Ernesto Flores Cesareo
2. //Programación I - "Programa: n primeros terminos de La serie de Fibonacci
   i"
3. #include <conio.h>
4. #include <stdio.h>
5. int main(){
6.     //Declarar variables
7.     int a,b,c,n,i;
8.     //Asignar valores iniciales
9.     a = 0;
10.    b = 1;
11.    //Entrada de n
12.    printf("Digite la cantidad de terminos que desea saber de la serie de
        Fibonacci%c: ",169);
13.    do{
14.        scanf("%d",&n);
15.        if(n <= 0){
16.            printf("\n[!] ERROR: Ingrese una cantidad entera valida [!] >");
17.        }
18.    }while(n <= 0);
19.    //Salidas asignadas
20.    switch(n){
21.        case 1:
22.            printf("%d",a);
23.            break;
24.        case 2:
```

BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA
FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

```
25.     printf("%d,%d",a,b);
26.     break;
27.     default:
28.         printf("%d,%d",a,b);
29.         for(i=1;i<=n-2;i++){
30.             c = a + b;
31.             if(i < n-2){
32.                 printf("%d",c);
33.             }
34.             else{
35.                 printf("%d",c);
36.             }
37.             a = b;
38.             b = c;
39.         }
40.     }
41.     //getch();
42.     printf("\n");
43.     system("pause");
44.     return 0;
45. }
```

EJECUCIÓN:

```
A:\Principal\Escritorio\Problema 3\Problema 5.exe
Digite la cantidad de terminos que desea saber de la serie de Fibonacci: 10
0,1,1,2,3,5,8,13,21,34
Presione una tecla para continuar . . .

-----
Process exited after 16.18 seconds with return value 0
Presione una tecla para continuar . . .
```

```
A:\Principal\Escritorio\Problema 3\Problema 5.exe
Digite la cantidad de terminos que desea saber de la serie de Fibonacci: 1
0
Presione una tecla para continuar . . .

-----
Process exited after 9.914 seconds with return value 0
Presione una tecla para continuar . . .
```

```
A:\Principal\Escritorio\Problema 3\Problema 5.exe
Digite la cantidad de terminos que desea saber de la serie de Fibonacci: 2
0,1
Presione una tecla para continuar . . .

-----
Process exited after 9.32 seconds with return value 0
Presione una tecla para continuar . . .
```

9. La siguiente se llama la conjetura de ULAM:
- Comience con cualquier entero positivo.
 - Si es par, divídalo entre 2; si es impar multiplíquelo por 3 y agréguele 1.
 - Obtenga enteros sucesivamente, repitiendo el proceso.

Al final, obtendrá el número 1, independientemente del número entero inicial.

Por ejemplo: 26,13,40,20,10,5,16,8,4,2,1

Elabore un programa en C que lea un entero positivo y obtenga la sucesión de ULAM.

CODIGO:

```
1. //Jesús Huerta Aguilar, Javier de La Luz Ruiz, Ernesto Flores Cesareo
2. //Programación I - Conjetura de ULAM
3. #include <stdio.h>
4.
5. int main(){
6.
7.     int num, i;
8.
9.     printf("Digite un numero entero positivo: ");
10.    scanf("%d", &num);
11.
12.    if (num>0){
13.        printf("\n El numero es positivo \n");
14.
15.        while (num!=1)
16.        {
17.            if (num%2==0){
18.                num=num/2;
19.                printf("\n %d", num);
20.
21.            }else{
22.                num=num*3+1;
23.                printf("\n %d", num);
24.            }
25.        }
26.
27.    }else
28.        printf("\n El numero no es positivo");
29.    system("pause");
30.    return 0;
31. }
```

EJECUCIÓN:

```
A:\Principal\Escritorio\Problemas 3\problema 9.exe
Digite un numero entero positivo: 10

El numero es positivo

5
16
8
4
2
1
-----
Process exited after 5.658 seconds with return value 0
Presione una tecla para continuar . . .
```

11. Realice un programa que compruebe si 2 números son amigos. Dos números son amigos si la suma de sus divisores del primero, excepto el, es igual al segundo y viceversa.
- Suma de los divisores de 220: $1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110 = 284$
 - Suma de los divisores de 284: $1, 2, 4, 71, 142 = 220$

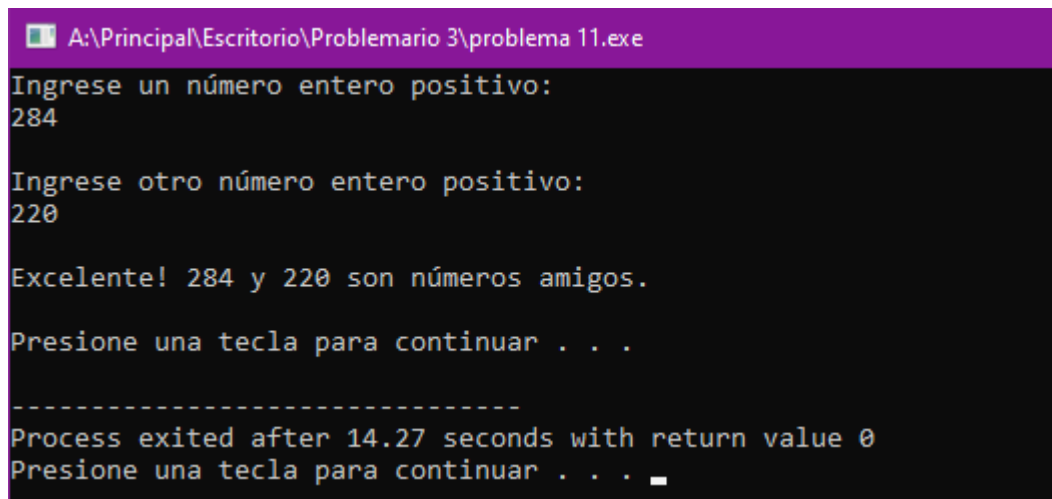
CODIGO:

```
1. //11. Programa que comprueba si dos numeros enteros son amigos.
2.
3. #include <stdio.h>
4.
5. int main(){
6.
7.     int x,y,i,divuno,divdos;
8.
9.     divuno = 0;
10.    divdos = 0;
11.
12.    printf ("Ingrese un n%cmero entero positivo:\n",163);
13.    scanf ("%d",&x);
14.    printf ("\nIngrese otro n%cmero entero positivo:\n",163);
15.    scanf ("%d", &y);
16.
17.    for (i = 1; i <= x/2; i++){
18.        if (x % i == 0) divuno += i;
19.    }
20.    for (i = 1; i <= y/2; i++){
21.        if (y % i == 0) divdos += i;
22.    }
23.
```

BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA
FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

```
24.     if (x == divdos && y == divuno){
25.         printf("\nExcelente! %d y %d son n%cmeros amigos.\n\n", x, y, 163
    );
26.     }
27.     else{
28.         printf("\nL%cstima, %d y %d no son n%cmeros amigos\n\n", 160, x,
    y, 163);
29.     }
30.
31.     system ("pause");
32.     return 0;
33. }
```

EJECUCIÓN:



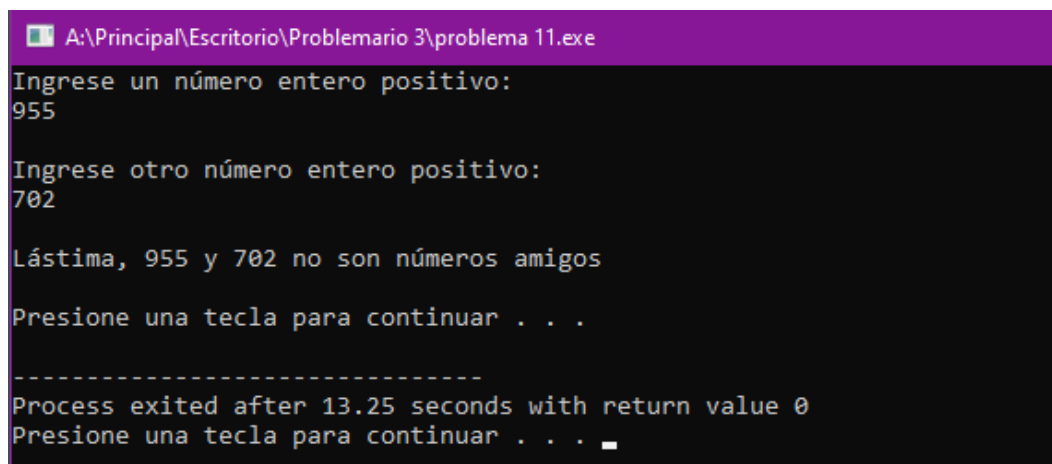
```
A:\Principal\Escritorio\Probleuario 3\problema 11.exe
Ingrese un número entero positivo:
284

Ingrese otro número entero positivo:
220

Excelente! 284 y 220 son números amigos.

Presione una tecla para continuar . . .

-----
Process exited after 14.27 seconds with return value 0
Presione una tecla para continuar . . .
```



```
A:\Principal\Escritorio\Probleuario 3\problema 11.exe
Ingrese un número entero positivo:
955

Ingrese otro número entero positivo:
702

Lástima, 955 y 702 no son números amigos

Presione una tecla para continuar . . .

-----
Process exited after 13.25 seconds with return value 0
Presione una tecla para continuar . . .
```

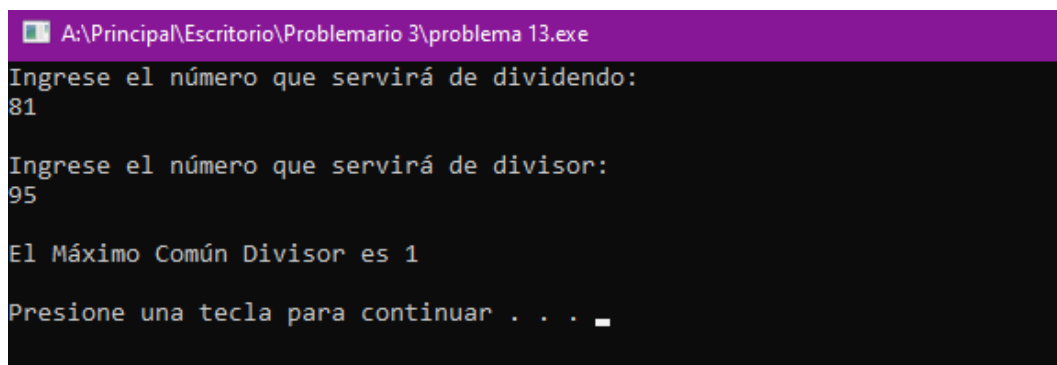

BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA
FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

13. Realice un programa que calcule el máximo común divisor de dos números mediante el algoritmo de Euclides.

CODIGO:

```
1. //13. Programa que calcula el mcd de dos números mediante el algoritmo de
   Euclides
2.
3. #include <stdio.h>
4.
5. int main(){
6.
7.     int x,y,res;
8.
9.     printf ("Ingrese el número que servirá de dividendo:\n",163,160);
10.    scanf ("%d",&x);
11.    printf ("\nIngrese el número que servirá de divisor:\n",163,160);
12.    scanf ("%d",&y);
13.
14.    res = x % y;
15.
16.    if (res == 0){
17.        printf ("\nEl Máximo Común Divisor es %d\n\n",160, 163, y);
18.    }
19.    while (res != 0){
20.        x = y;
21.        y = res;
22.        res = x % y;
23.        if (x % y == 0 ){
24.            printf ("\nEl Máximo Común Divisor es %d\n\n",160, 163, y);
25.        }
26.    }
27.
28.    system ("pause");
29.    return 0;
30.}
```

EJECUCIÓN:



```
A:\Principal\Escritorio\Problematario 3\problema 13.exe
Ingrese el número que servirá de dividendo:
81
Ingrese el número que servirá de divisor:
95
El Máximo Común Divisor es 1
Presione una tecla para continuar . . . _
```

```
A:\Principal\Escritorio\Problema 3\problema 13.exe
Ingrese el número que servirá de dividendo:
88

Ingrese el número que servirá de divisor:
64

El Máximo Común Divisor es 8

Presione una tecla para continuar . . .

-----
Process exited after 18.79 seconds with return value 0
Presione una tecla para continuar . . . _
```

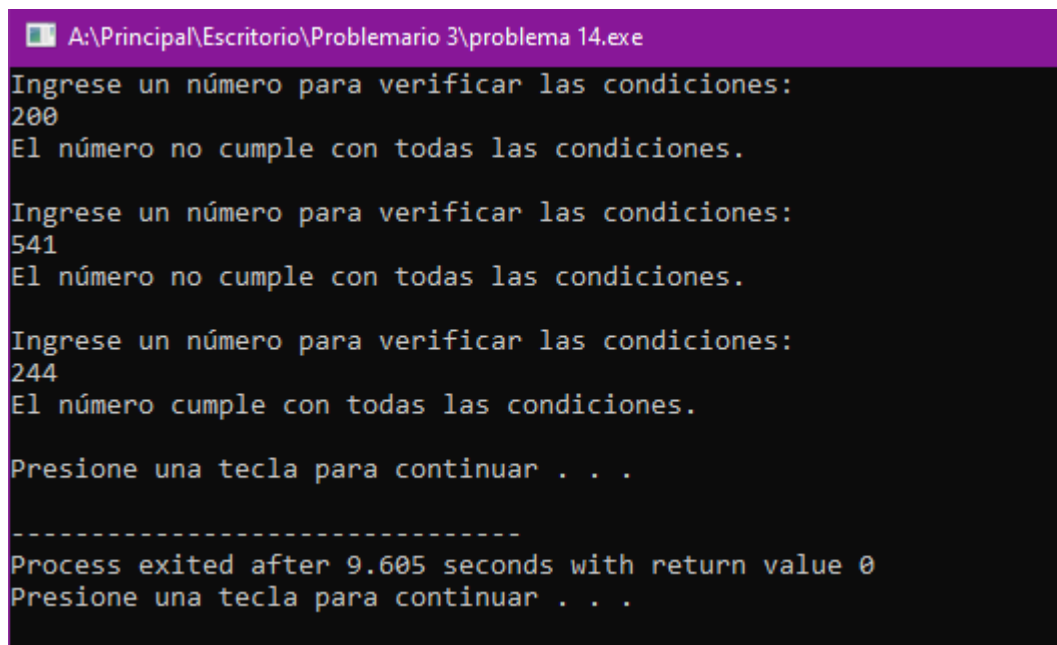
14. Escribir un programa que lea números enteros de teclado hasta que encuentre uno que cumpla las siguientes condiciones:
- Múltiplo de 2
 - No múltiplo de 5
 - Mayor de 100
 - Menor de 10,000

CODIGO:

```
15. /*14. Programa que Lee números enteros hasta que cumpla ciertas condiciones
    es
16. 1. Sea múltiplo de 2
17. 2. No múltiplo de 5
18. 3. Mayor que 100
19. 4. Menor que 10,000
20. */
21.
22. #include <stdio.h>
23.
24. int main () {
25.
26.     int x, sansu;
27.
28.     do {
29.         printf("Ingrese un número para verificar las condiciones:\n", 163);
30.         sansu = 0;
31.         scanf("%d", &x);
32.
33.         if (x >= 100 && x <= 10000) {
34.             if (x % 2 == 0 && x % 5 != 0) {
35.                 printf("El número cumple con todas las condiciones.\n", 1
36. 63);
```

```
36.         sansu=1;
37.     }
38.     else{
39.         printf ("El n%cmero no cumple con todas las condiciones.\n",163);
40.     }
41. }
42. else{
43.     printf ("El n%cmero no est%c en el rango solicitado.\n",163,1
60);
44. }
45.
46.     printf ("\n");
47.
48. }while(sansu == 0);
49.     system("pause");
50.     return 0;
51. }
```

EJECUCIÓN:



```
A:\Principal\Escritorio\Problematario 3\problema 14.exe
Ingrese un número para verificar las condiciones:
200
El número no cumple con todas las condiciones.

Ingrese un número para verificar las condiciones:
541
El número no cumple con todas las condiciones.

Ingrese un número para verificar las condiciones:
244
El número cumple con todas las condiciones.

Presione una tecla para continuar . . .

-----
Process exited after 9.605 seconds with return value 0
Presione una tecla para continuar . . .
```