**Sergio Moncada Muñoz**

**1089380570**

**Programación 2**

**Taller while**

// Taller while

#include <stdio.h>

/\* 1) Implementar un algoritmo que muestre todos los múltiplos de 7

comprendidos entre 1 y 1000. \*/

void multiplos7 ()

{

int i;

i=1;

while (i<=1000)

{

if (i%7==0)

{

printf ("%d ", i);

}

i=i+1;

}

}

/\* 2) Implementar un algoritmo que calcule la siguiente suma:

3(1)+3(2)+3(3)+3(4)+…..+3(n), ingresar n por teclado mostrar el

resultado de dicha suma. \*/

int suma (int n)

{

int i, r;

i=1;

while (i<=n)

{

r=r+(3\*i);

i=i+1;

}

return r;

}

/\* 3) Leer por teclado cuatro notas de un estudiante en una sola

variable, cada vez el acumulado se debe guardar en una variable

llamada total. Cuando se lean las cuatro determinar el promedio,

si el promedio es mayor o igual a tres, imprimir “ganaste” de lo

contrario imprimir “perdiste”. \*/

void notas ()

{

int n;

float total, nota;

n=0;

while (n<4)

{

printf ("Ingrese la nota: ");

scanf("%f",&nota);

total=total+nota;

n=n+1;

}

if (total/4>=3.0)

{

printf ("Ganaste");

}

else

{

printf ("Perdiste");

}

}

/\* 4) Un entero positivo n se llama perfecto si n es igual a la suma de

todos sus divisores diferentes de él. Por ejemplo, 6 es perfecto porque

6 = 1 + 2 + 3. Escriba un programa que lea por teclado un número entero

positivo, y nos indique si éste es perfecto o no. Además, el programa debe

imprimir todos los divisores del número. \*/

void perfecto (int n)

{

int i, r;

i=1;

r=0;

printf ("Los divisores de %d son: ", n);

while (i<n)

{

if (n%i==0)

{

printf ("%d ", i);

r=r+i;

}

i=i+1;

}

if (r==n)

{

printf ("\n%d es un numero perfecto\n", n);

}

else

{

printf ("\n%d no es un numero perfecto\n", n);

}

}

//------------------------------------------------------------------------

/\* 5) Leer dos números enteros por teclado y determinar si éstos son números

amigos(Dos números amigos son dos enteros positivos a y b tales que a es la

suma de los divisores propios de b, y b es la suma de los divisores propios

de a. la unidad se considera divisor propio, pero no lo es el mismo número). \*/

void amigos (int a, int b)

{

int r1, r2, i1, i2;

r1=0;

r2=0;

i1=1;

i2=1;

while (i1<a)

{

if (a%i1==0)

{

r1=r1+i1;

}

i1=i1+1;

}

while (i2<b)

{

if (b%i2==0)

{

r2=r2+i2;

}

i2=i2+1;

}

if (r2==a && r1==b)

{

printf ("%d y %d son numeros amigos\n", a, b);

}

else

{

printf("%d y %d no son numeros amigos\n", a, b);

}

}

//------------------------------------------------------------------------

/\* 6) Leer un número entero positivo n por teclado y mostrar por pantalla

la secuencia de fibonacci con n terminos.(ejemplo: si se lee un 3 se debe

mostrar la serie hasta el tercer número 1,1,2. Si se lee un 5, se debe mostrar

la serie hasta el quinto número 1,1,2,3,5). \*/

int fibonacci (int n)

{

int aux, aux2, aux3, i;

i=1;

aux=1;

aux2=1;

while (i<=n)

{

printf ("%d ", aux);

aux3=aux;

aux=aux2;

aux2=aux3+aux2;

i=i+1;

}

}

//------------------------------------------------------------------------

/\* 7) Leer un entero n por teclado y calcular su factorial, n!=1\*2\*3\*4\*....\*n. \*/

void factorial (int n)

{

int i, r;

r=1;

i=1;

while (i<=n)

{

r=r\*i;

i=i+1;

}

printf ("El factorial de %d es %d\n", n, r);

}

//------------------------------------------------------------------------

/\* 8) Implementar un algoritmo que diga cuantos números perfectos hay entre 1 y 1000. \*/

void contador ()

{

int i, cont, n, r;

i=1;

cont=0;

while (i<=1000)

{

n=1;

r=0;

while (n<i)

{

if (i%n==0)

{

r=r+n;

}

n=n+1;

}

if (r==i)

cont=cont+1;

i=i+1;

}

printf ("Entre 1 y 1000 hay %d numeros perfectos\n", cont);

}

//------------------------------------------------------------------------

/\* 9) Implementar un algoritmo que diga cuales números primos hay entre 1 y 1000

(debe mostrar los numeros) \*/

void primos ()

{

int i, n, cont;

i=2;

while (i<=1000)

{

n=2;

cont=0;

while (n<i)

{

if (i%n==0)

cont=cont+1;

n=n+1;

}

if (cont==0)

printf ("%d ", i);

i=i+1;

}

}

//------------------------------------------------------------------------

/\* 10) Hacer un programa que muestre los factoriales de los primeros 10 enteros \*/

void factorial2 ()

{

int i, n, r;

n=1;

while (n<=10)

{

i=1;

r=1;

while (i<=n)

{

r=r\*i;

i++;

}

printf ("Factorial de %d: %d\n", n, r);

n++;

}

}

//------------------------------------------------------------------------

/\* 11) Implementar las tablas de multiplicar del 10 al 20. \*/

void tablas ()

{

int i, n, r;

n=10;

while (n<=20)

{

i=1;

while (i<=10)

{

r=n\*i;

printf ("%d \* %d = %d\n", n, i, r);

i++;

}

putchar ('\n');

n++;

}

}

//------------------------------------------------------------------------

main ()

{

int a, b, n, r;

// multiplos7 ();

/\* printf("Ingrese el numero hasta el que desea calcular la sumatoria: ");

scanf ("%d", &n);

r=suma (n);

printf("El resultado de la sumatoria es %d", r); \*/

// notas ();

//perfecto (28);

/\*printf ("Ingrese el primer numero: ");

scanf("%d", &a);

printf("Ingrese el segundo numero: ");

scanf("%d", &b);

amigos (a, b);\*/

/\*printf ("Ingrese el numero hasta el que desea calcular fibonacci: ");

scanf("%d",&n);

fibonacci (n);\*/

// factorial (6);

// contador ();

// primos ();

// factorial2 ();

// tablas ();

}