PRACTICA No 5

Objetivo general:
En esta práctica aprenderemos a usar el ciclo de repetición for, el comando gotoxy, el textcolor, el textbackground estos últimos para darle mejor presentación a nuestros programas en conjunto con el ciclo for.

Desarrollo:

En esta práctica realizamos tres programas los cuales en general fueron series de números que cumplen ciertas condiciones.
El primer programa es el siguiente:
El programa debe calcular el valor de e^x mediante la siguiente formula:

$$e^x = 1 + \frac{x^1}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^n}{n!}$$

Donde x puede tomar solo el valor de uno o el valor de cero; n es el limite al que se puede llegar y debe tambien ser introducido por el usuario.

Por tanto la *metodología* del programa es tomar un valor que se vaya incrementando de uno en uno para poder sacar el factorial que se necesita en forma de cadena en la formula además de que este valor sirve para obtener el valor de x elevado a esta variable.
Para hacer el marco partimos que este tiene que ser de 80 por 49 para que este en limite de la pantalla. Ayudados del comando gotoxy que es el que nos manda el cursor a cieta cordenada de la pantalla donde "x" es la horizontal y "y" es la vertical podemos formar un ciclo for que contenga una variable y que el limite de esta sea el máximo que se puede ver en pantalla para cada caso se podra usar un decremento o un incremento según se quiera que se aparezcan el símbolo que formara el marco.
Por lo tanto el programa quedo de la siguiente forma:

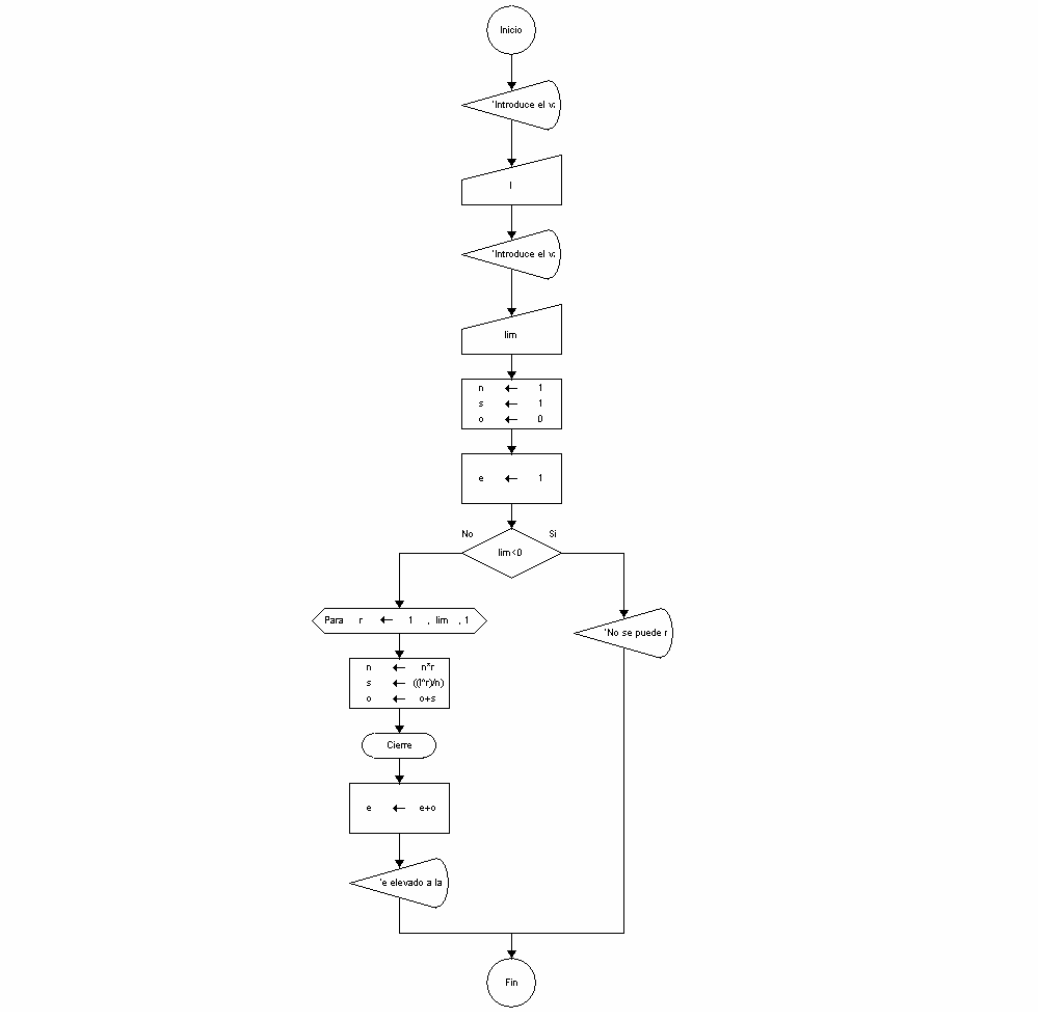
```
/* FECHA: 27-SEP-06
PROGRAMA: 13
OBJETIVO: /*MEJORAR LA APARICIENCIA DE NUESTRO PROGRAMA
APRENDIENDO NUEVOS COMANDOS ADEMAS DE CONOCER EL
CICLO DE REPETICIÓN FOR */
#include <stdio.h>
#include <dos.h>
#include <conio.h>
#include <stdlib.h>
#include <math.h>
void main ()
{
int x,l,i,lim,r;
float s=1.0,e=1.0,n=1.0,o=0.0;
textbackground (RED);
textcolor (GREEN);
clrscr ();

for (x=1;x<=80;x++)
{
gotoxy (x,1);
printf ("*");
delay (25);
}
```

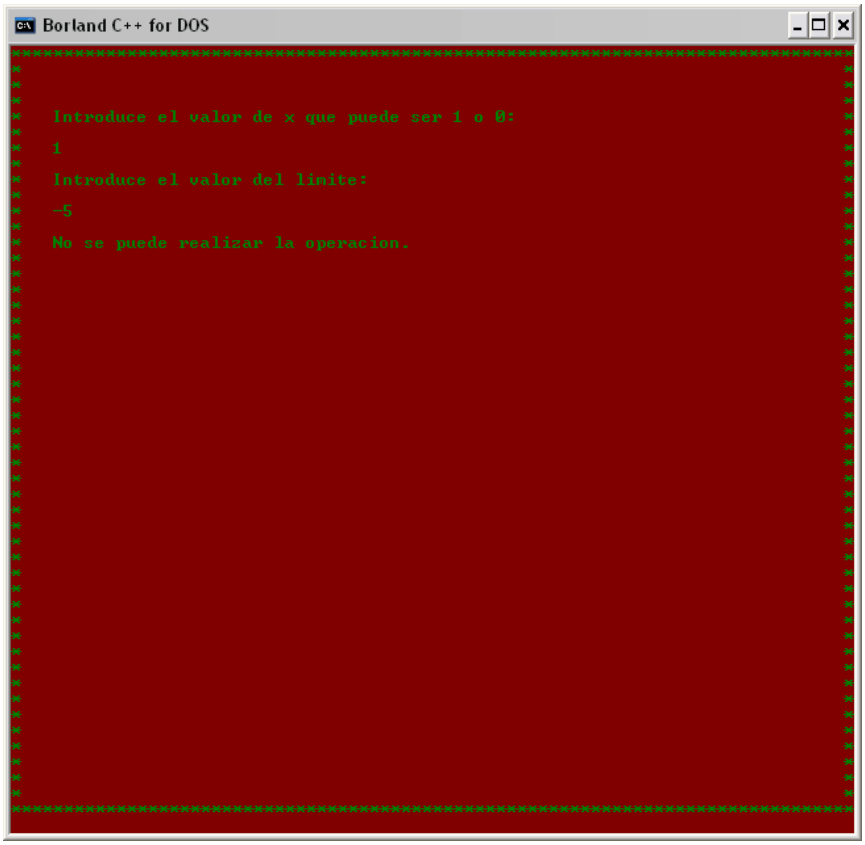
```
for (x=1;x<=49;x++)
{
gotoxy(80,x);
printf ("*");
delay (25);
}
for (x=80; x>=1;--x)
{
gotoxy (x,49);
printf ("*");
delay (25);
}
for (x=49;x>=1;--x)
{
gotoxy (1,x);
printf ("*");
delay (25);
}
gotoxy (5,5);
printf ("Introduce el valor de x que puede ser 1 o 0:");
gotoxy (5,7);
scanf ("%d", & l);
gotoxy (5,9);
printf ("Introduce el valor del limite:\n");
gotoxy (5,11);
scanf ("%d", & lim);
if (lim<0)
  {
  gotoxy (5,13);
  printf ("No se puede realizar la operacion.");
  }
else
    {
    for (r=1; r<=lim;r++)
       {
       n*=r;
       s=((pow(l,r))/(n));
       o+=s;
       }
    e+=o;
    gotoxy (5,15);
    printf ("e elevado a la x es:%f\t",e);
    }
getch ();
}/*CONCLUSIONES: EN ESTE PROGRAMA APRENDIMOS A HACER UN
MARCO CON UN SÍMBOLO MEDIANTE EL COMANDO FOR Y EL
COMANDO GOTOXY ADEMAS DE APLICAR A LA PROBLEMÁTICA
DEL PROBLEMA EL CICLO FOR, EL COMANDO GOTOXY TAMBIEN
NOS SIRVIO PARA UBICAR NUESTROS TEXTOS Y LOS ESCANEOS
DE VARIABLES NECESARIOS, ADEMAS CONOCIMOS NUEVAS
LIBRERIAS TALES COMO DOS Y STLIB, TAMBIEN EL COMANDO DELAY
QUE NOS MARCA LOS TIEMPOS EN QUE SE MUESTRA EL ASTERISCO
(ENTRE MAS  GRANDE SEA EL VALOR DE ESTE MAYOR SERA
EL TIEMPO DE RETARDO).*/
```

El programa funciona de la manera mas simple tomando a r como una variable que inicia en 1 y que se va incrementando en 1 hasta llegar al limite; n se multiplica por el valor de r (asi obtenemos el factorial); La variable s es igual a r elevado a la r entre n que es el factorial hasta donde se lleva a r; o es una variable que inicia en cero y se va incrementando en el valor de s;
Al salir del ciclo for tenemos a la variable e que incia en uno y a este le sumamos el ultimo valor que tuvo o teniendo asi el uno mas todas las operaciones donde se involucra x y el factorial.

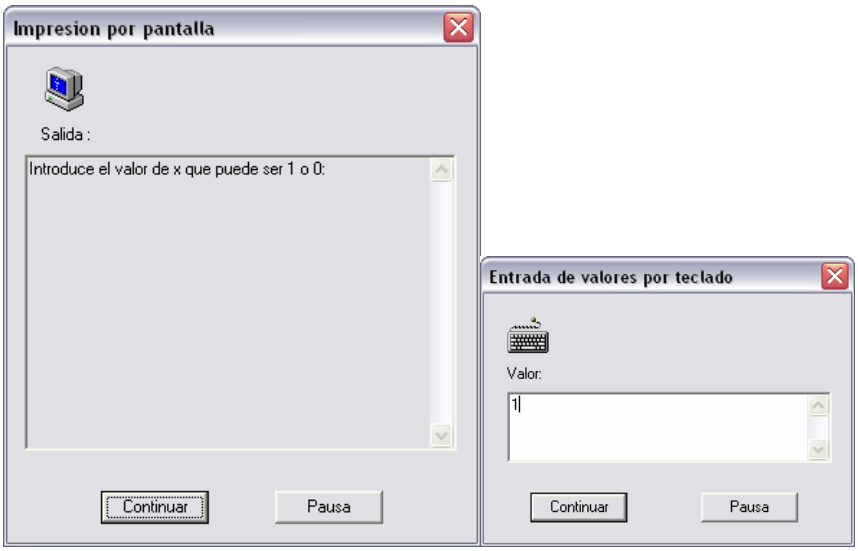
El diagrama de flujo quedo de la siguiente manera y maneja el mismo principio del programa en lenguaje c solo que aquí no se pueden hacer los marcos.

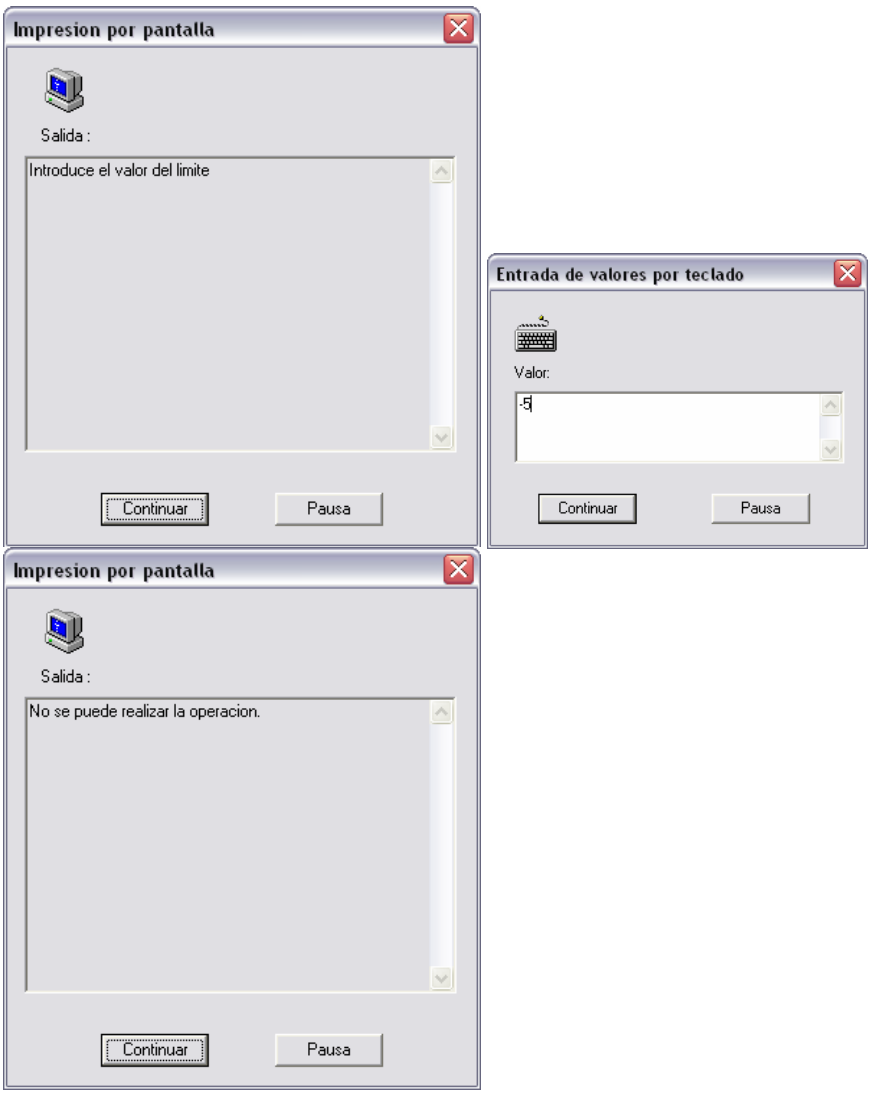


La corrida del programa se ve de la siguiente forma:
Tenemos para el primer caso en que el limite o valor de n sea un entero pero no positivo.



La corrida en dfd se veria asi:

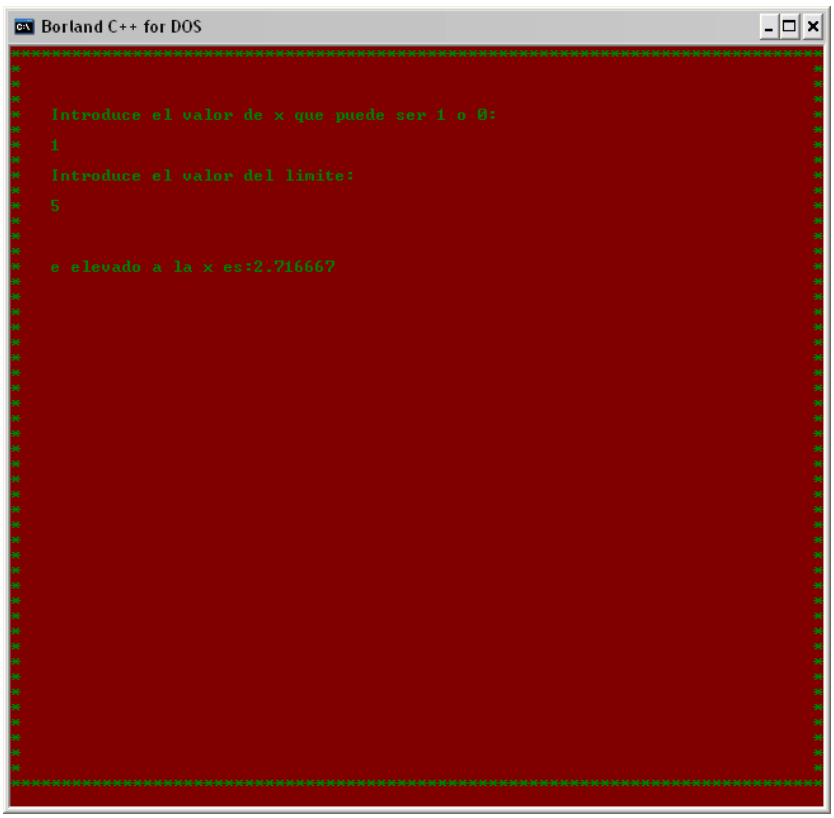


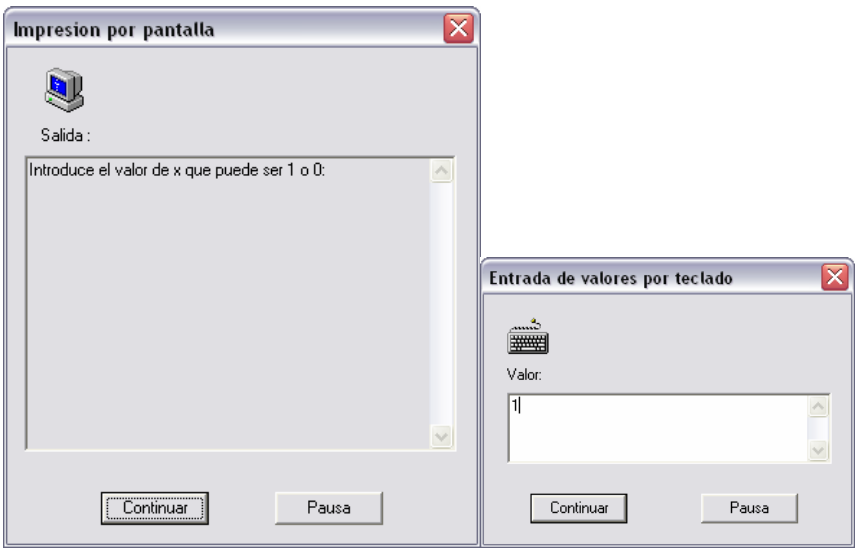


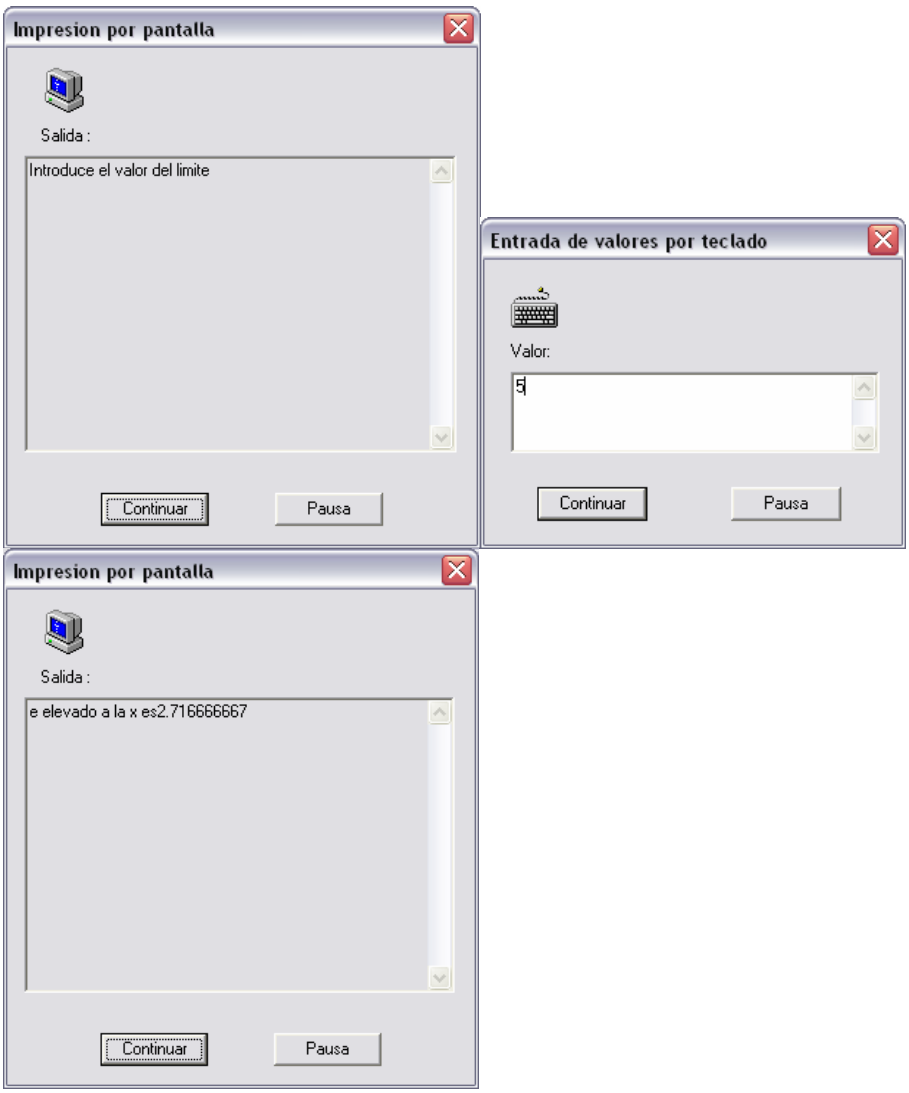
El segundo caso es que si se tenga un valor entero positivo por lo que tendremos en pruba de escritorio un valor de x = 1 y un limite de 4.Por lo que:

$$e^x = 1 + \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1^4}{4} = 2.71667$$

Por lo que las corridas se veran de la siguiente forma:


Borland C++ for DOS
Introduce el valor de x que puede ser 1 o 0:
1
Introduce el valor del limite:
5
e elevado a la x es:2.716667


Impresion por pantalla
Salida :
Introduce el valor de x que puede ser 1 o 0:
Continuar
Pausa
Entrada de valores por teclado
Valor:
1
Continuar
Pausa



El segundo programa es el relaizar la llamada serie Fibonacci con la cual se debe obtener una serie similar a esta: 1 2 3 5 8 13 21
Se puede obsevar que a partir del dos el numero que sigue es la suma de los dos números anteriores.
Como *metodología* decidi usar un ciclo para o "for" y dentro de este tres variables:
Una s que inicia en cero una l que inicia en uno; dentro del ciclo for reiniciamos a s en cero luego le asignamos la suma de i+l, luego a i le asignamos el valor de l y a l el valor de s este ciclo se repetirá mientras que s sea menor al limite por lo que el valor de s tomara el valor de la suma de los dos anteriores valores de i y l y i y l guardaran ahora el valor anterior y el valor de s tal que se puede realizar la serie fibonacci, después usamos un condicional if para poder mandara imprimir todos los números menores al limite ya que se puede repetir en el ultimo ciclo que s rebase el valor del limite; el valor de z es usado para ubicar mediante el gotoxy (5, z) el valor de la serie por lo que este ira en una columna que siempre empezara en la columna 5 y comenzara a descender desde la fila 10(valor inicial de z).
El codigo fuente del programa quedo de la siguiente forma:

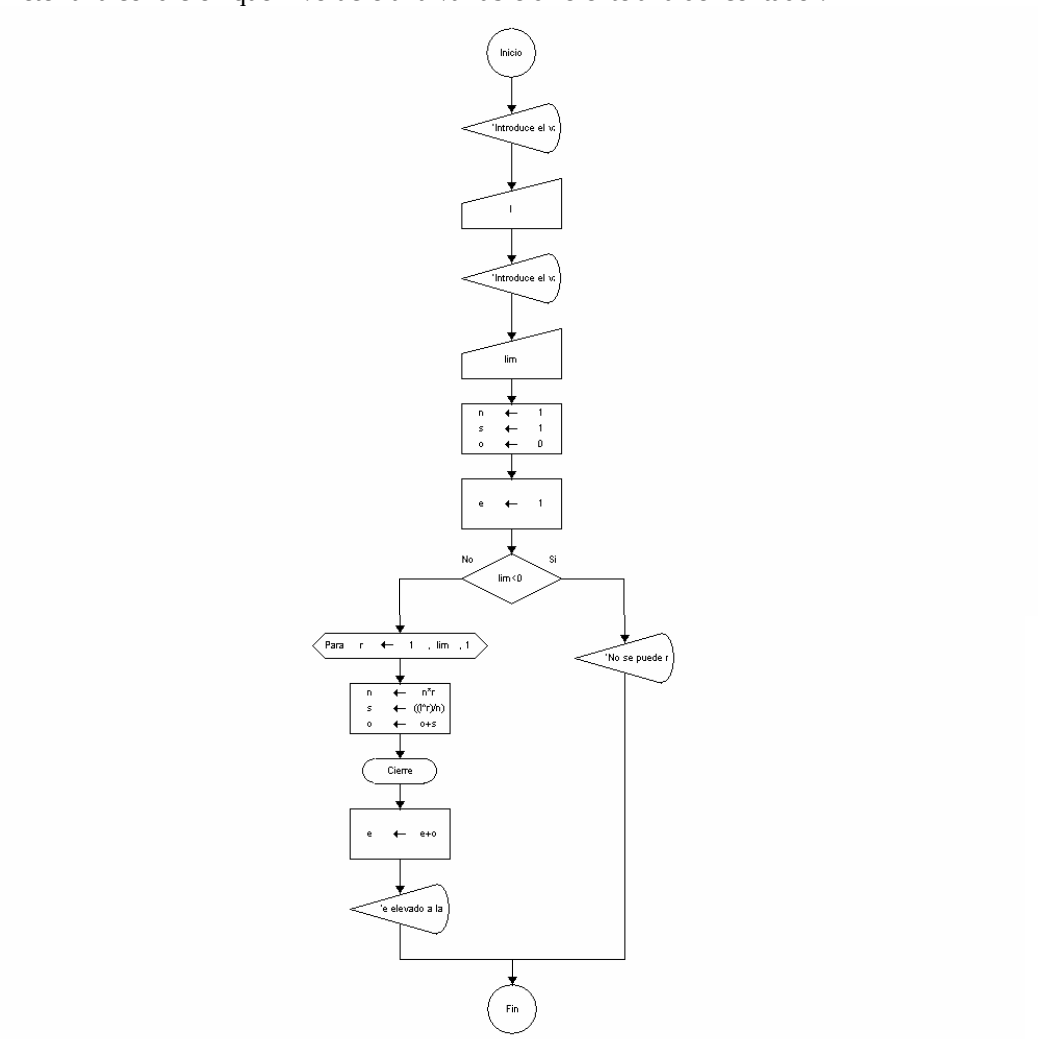
/* FECHA: 27-SEP-06
PROGRAMA: 14
OBJETIVO: /*OBSERVAR QUE EL MARCO QUE HEMOS

HECHO ANTERIORMENTE SE APLICA A NUEVOS PROGRAMAS PUDIENDO MEJORAR LA APARIENCIA DE TODOS ADEMAS DE VER QUE EL CICLO FOR ES UNA MEJORA DEL CICLO WHILE EN MUCHOS CASOS YA QUE CONTIENE IMPLICITO UN CONTADOR Y UNA CONDICION PARA QUE SE DECIDA ENTRAR O NO AL CICLO*/

```
#include <stdio.h>
#include <dos.h>
#include <conio.h>
#include <stdlib.h>
#include <math.h>
void main ()
{
int x,s=0,i,l=1,lim,z=10;
textbackground (CYAN);
textcolor (RED);
clrscr ();
for (x=1;x<=80;x++)
{
gotoxy (x,1);
printf ("*");
delay (25);
}
for (x=1;x<=49;x++)
{
gotoxy(80,x);
printf ("*");
delay (25);
}
for (x=80; x>=1;--x)
{
gotoxy (x,49);
printf ("*");
delay (25);
}
for (x=49;x>=1;--x)
{
gotoxy (1,x);
printf ("*");
delay (25);
}
gotoxy (5,5);
printf ("Introduce el valor del limite:");
gotoxy (5,7);
scanf ("%d", & lim);
for (i=0; s<=lim; i)
  {
  s=0;
  s+=(i+l);
  i=l;
  l=s;
  z++;
  if (s<=lim)
    {
    gotoxy (5,z);
```

```
    printf ("%d", s);
    }
  }
getch ();
}
/*CONCLUSIONES: EL CICLO FOR NOS AHORRA ESPACIO EN
CUANTO A UNA DIFERENCIA CON EL CICLO DE REPETICIÓN WHILE
ADEMAS DE QUE NOS PERMITE USAR ESTE CICLO SIN UN INCREMENTO
DE UNA CONTANTE DADA COMO EN ESTE PROGRAMA DONDE APARTE
LA CONDICION NO DEPENDE DIRECTAMENTE DEL CONTADOR.
TAMBIEN OBSERVAMOS QUE PODEMOS CAMBIAR EL COLOR DEL
FONDO Y DEL TEXTO CON LOS COMANDOS TEXTBACKGROUN Y
TEXTCOLOR Y ASI DARLE MEJOR PRESENTACION Y UNA
PERSONALIZACIÓN A NUESTRA PROGRAMA*/
```

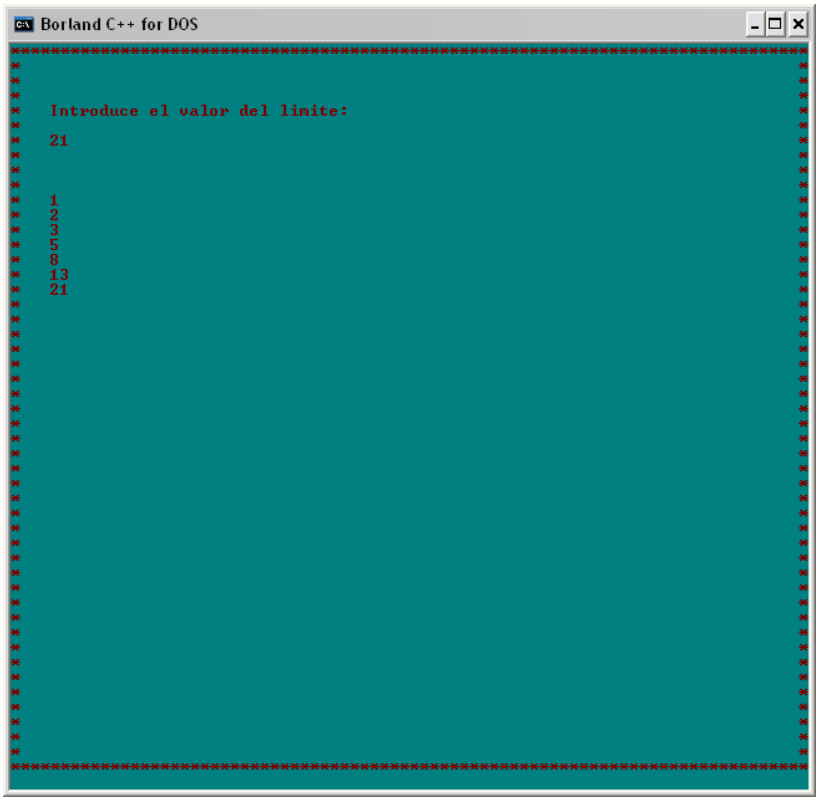
El diagrama de flujo se observa a continuación con un while ya que en el for del dfd no se puede meter una condicion que involucre una variable diferente a la del contador.



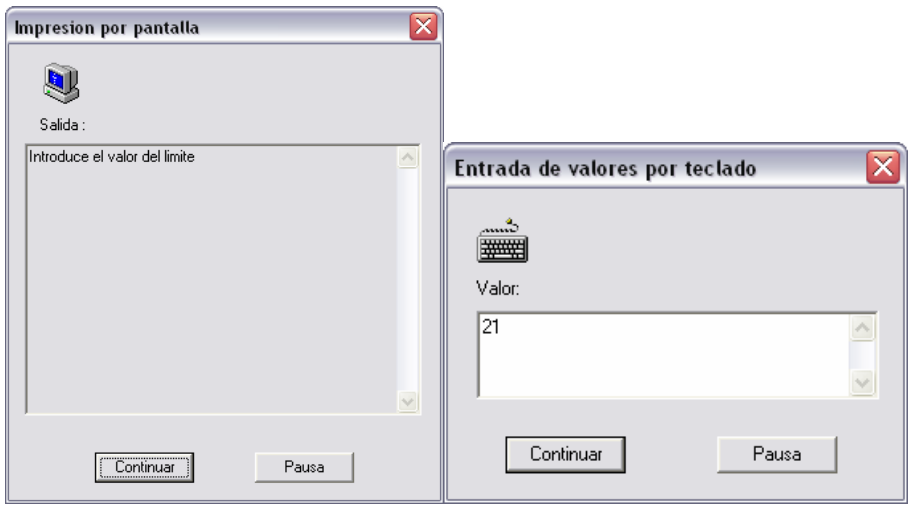
A manera de prueba de escritorio muestro la siguiente tabla donde se observa como van cambiando las variables mientras que s es menor al limite.

S=(L+I)	I	L

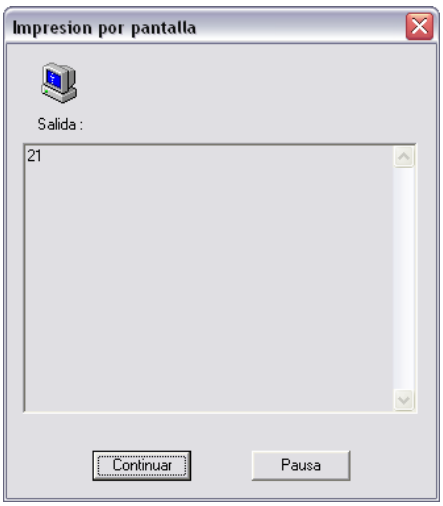
0	0	1
1	1	1
2	1	2
3	2	3
5	3	5
8	5	8
13	8	13
21	13	21



La corrida en dfd se veria de la siguiente forma:




Impresion por pantalla
Salida :
1
Continuar
Pausa
Impresion por pantalla
Salida :
2
Continuar
Pausa
Impresion por pantalla
Salida :
3
Continuar
Pausa
Impresion por pantalla
Salida :
5
Continuar
Pausa
Impresion por pantalla
Salida :
8
Continuar
Pausa
Impresion por pantalla
Salida :
13
Continuar
Pausa



El siguiente programa es el de la criba de Erastotenes que no es mas que obtener todos los números primos del 1 hasta el limite introducido por el usuario.

Por lo tanto a modo de *metodología* y explicación del programa esta lo siguiente:
Primero iniciamos un contador "x" que va desde 1 hasta el límite aumentando de 1 en 1, luego tenemos a otro contador "y" que inicia también en 1 y que va desde 1 hasta el limite aumentando también de 1 en 1. Luego ponemos una condicional if dentro del for de contador "y", esta condicional evalúa si "x" modulo de "y" es igual a cero y si "x" y "y" son iguales si la afirmación es correcta cumple con la primera condición de un numero primo que su división entre si mismo de cómo resultado la unidad, entonces si se cumple entramos a un ciclo for de contador "z" donde este va desde 2 hasta un numero antes de "x"(z<x)por ir aumentando de 1 en 1 luego dentro de este ciclo evaluamos si "x" modulo de "z" es igual a cero y si "z" es diferente de "x" (Esto ultimo esta por demás ya que sabemos que "z" no tomara nunca el valor de "x") si esto se cumple quiere decir que el numero que estamos evaluando es un múltiplo de un numero anterior a el, por lo que no cumple la condición de que solo sea divisible entre él y la unidad (como todo numero real es divisible entre la unidad esta demás comprobarlo) por lo que debemos hacer que el ciclo for de "z" termine ya que determinamos que no es un numero primo le asignamos a "z" el valor de "x" con lo que automáticamente nos sacara del ciclo permitiendo que "x" y "y" se aumente en uno; Si la condición no es verdadera nos vamos al "else" donde imprimiremos el valor de "x" pero solo si "z+1 es igual a x" esto quiere decir que estamos en el ultimo valor que puede tomar "z" (un numero anterior a x) habiendo ya evaluado todos los valores anteriores a x y no encontrado ningún numero que nos divida a nuestro candidato y que su residuo sea cero estamos en condiciones de decir que es primo y lo mandamos a imprimir en la pantalla
El código fuente del programa quedo de la siguiente forma:

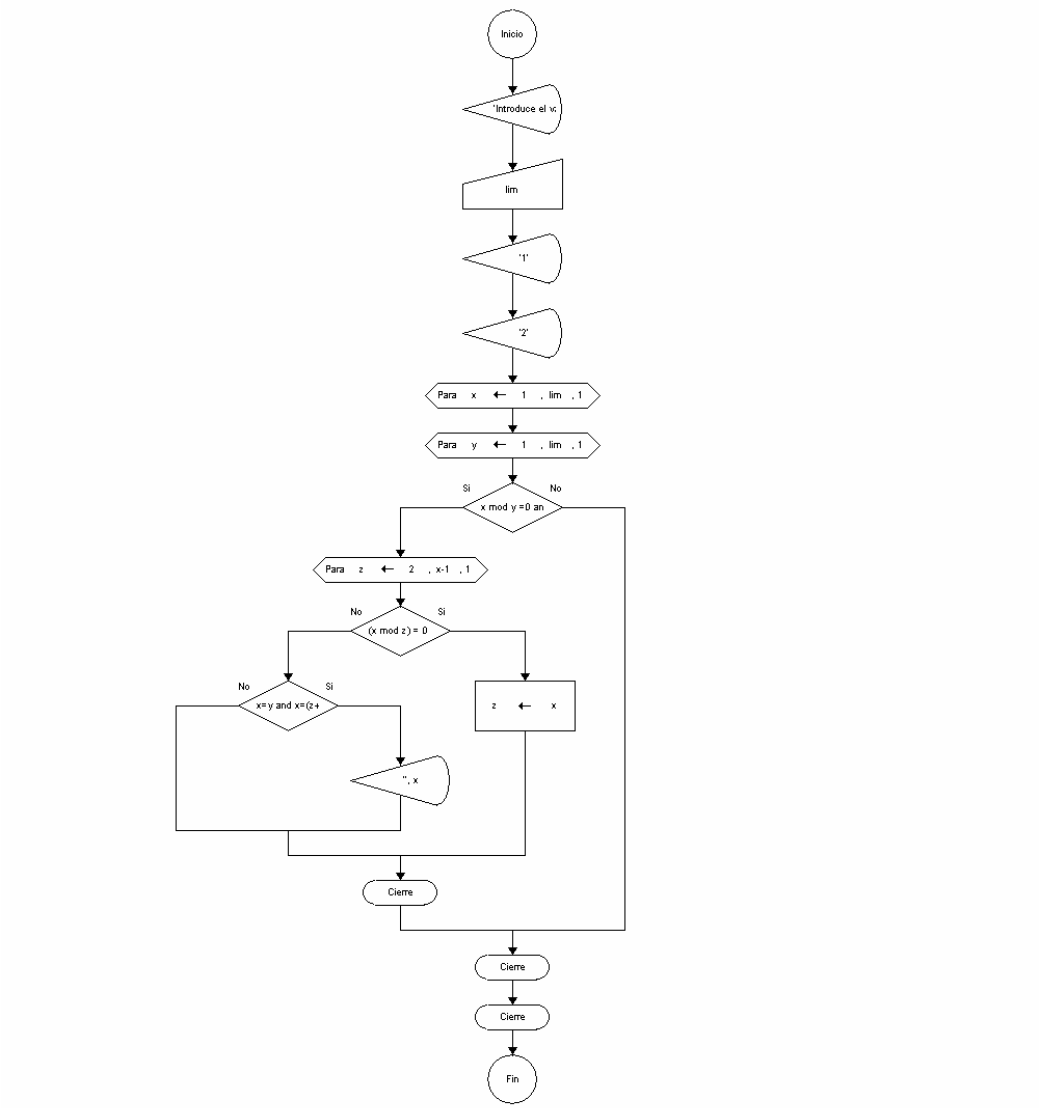
```
/*
FECHA: 27-SEP-06
PROGRAMA: 14
OBJETIVO: /*MEDIANTE EL ANALISIS DE VARIABLES OBTENER MEDIANTE
EL USO DEL CICLO DE REPETICIÓN FOR LOS NUMEROS PRIMOS EN FORMA
DE SERIE DADO UN LIMITE ADEMAS DE AGREGAR UN MARCO Y VER LA
GRAN UTILIDAD Y AHORRO DE ESPACIO QUE NOS DA EL CILO FOR*/
#include <stdio.h>
#include <dos.h>
#include <conio.h>
#include <stdlib.h>
#include <math.h>
void main ()
{
```

```
int x,y,l,z,m=8,lim,n=6,p=6;
textbackground (LIGHTGREEN);
textcolor (BLUE);
clrscr ();
for (l=1;l<=80;l++)
{
gotoxy (l,1);
printf ("*");
delay (10);
}
for (l=1;l<=49;l++)
{
gotoxy(80,l);
printf ("*");
delay (10);
}
for (l=80; l>=1;--l)
{
gotoxy (l,49);
printf ("*");
delay (10);
}
for (l=49;l>=1;--l)
{
gotoxy (1,l);
printf ("*");
delay (10);
}
gotoxy (5,5);
printf ("Introduce el valor del limite:\t");
scanf ("%d", & lim);
gotoxy (5,7);
printf ("1");
gotoxy (5,8);
printf ("2");
for (x=1; x<=lim; x++)
  {
  for (y=1; y<=lim; y++)
    {
    if (x%y ==0 && x==y)
          {
          for (z=2; z<x; z++)
            {
            if (x%z == 0 && z!=x)
              {
              z=(x);
              }
            else
               {
               if (x==y && x==(z+1))
                     {
                     if (m<48)
                       {
                       m++;
                       gotoxy (5,m);
```

```
                    printf ("%d",x);
                    }
                  else
                     {
                     if (n<48)
                       {
                       n++;
                       gotoxy (20,n);
                       printf ("%d",x);
                       }
                     else
                           {
                           p++;
                           gotoxy (30,p);
                           printf ("%d",x);
                  }
                     }
                  }
              }
           }
        }
     }
  }
getch ();
}
/*CONCLUSIONES: EL PROGRAMA QUE SE REALIZO
NECESITABA DE MUCHO ANALISIS LO QUE NOS HABLA
DE QUE PARA REALIZAR UN PROGRAMA SE NECESITA DE LOS
CONOCIMIENTOS Y EL INGENIO DEL PROGRAMADOR COMO BASE
PRINCIPAL PARA EL DESARROLLO PRACTICO DE ESTE PROGRAMA*/
```

En el anterior programa no se hablo de la impresión del programa pero es inferido que estamos mostrando columnas mediante l comando gotoxy y para no salirnos del margen tenemos que si la fila rebasa cierto limite pasamos a otra columna mediante el uso del else y el inicio de una nueva variable que determina la fila y si necesitamos mas columnas solamente agregamos mas variables que cumplan la misma base que nos da como resultado if en cascada. Como comentario también cabe decir que nuestro programa no calcula ni el 1 ni el dos como números primos por que son dos números especiales dentro de ellos por lo que hay que imprimirlos directamente.

El diagrama de flujo en dfd del programa donde obtenemos los números primos es el siguiente:



La corrida del programa se da de la siguiente forma teniendo a manera de *prueba de escritorio* el siguiente análisis empezando con el tres por tomar en cuenta la nota anterior .

Tenemos a “x” igual a 3 y a “y que inicia en 1 y llega hasta el valor de 3 momento en el cual se cumple la primera condición entonces comienza el ciclo for de “z” en dos y empieza evaluando en su ciclo si “x” modulo de “z” es igual a cero pero no encuentra verdadero esto mientras que la segunda condición donde “x es igual “z+1” es verdadera por lo que procede a la impresión y sale de este ciclo y y llega hasta el limite que en este caso será 5 pero no se cumple la primera condición por lo que saliendo de este ciclo hace que se vuelva al ciclo for de”x” por lo que “x” ahora vale cuatro entonces de nuevo y empieza valiendo 1 y llega hasta tres sin que se cumpla la primera condición pero al llegar a cuatro se cumple la condición y entra al ciclo for de “z” pero inmediatamente se cumple la condición que esta dentro de este ultimo ciclo ya que 4 entre dos da como residuo cero por lo que se la da a “z” el valor de “x” y entonces se intenta entrar de nuevo al ciclo pero no se cumple la condición por lo que se vuelve al ciclo for de “y” que va hasta el limite sin que se cumpla la primera condición después de salir el contador “x” ahora vale 5 y de nuevo se incia a “y” en uno y pasa por los números 1,2,3,4 sin que se cumpla la primera condición de que “x modulo de y” sea cero y “x sea igual a y” hasta que “y” llega al cinco se cumple la condición por lo que se inicia el contador z en dos y se evalúa si “x modulo de z e igual a cero” pero el contador pasa por 2,3,4 y al llegar a este cuatro se cumple la condición de que “x sea igual a (z+1)” por lo que se imprime el valor de “x” que en este caso es

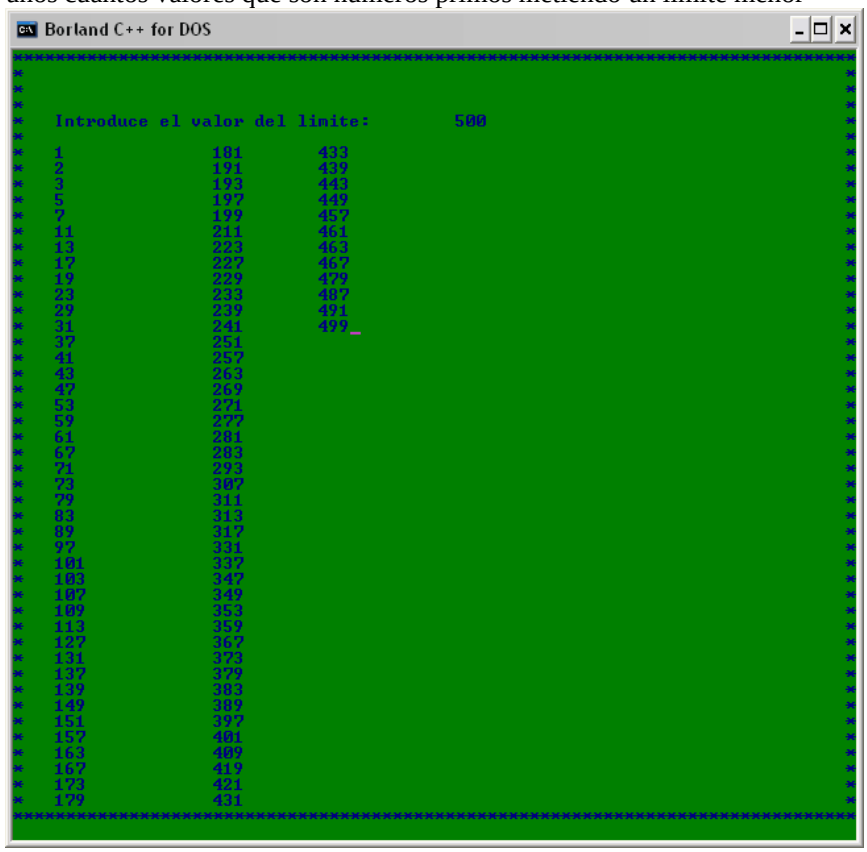
cinco por lo que podemos afirmar que si se encontraron los valores deseados pudiendo afirmar que el algoritmo sirve y que el programa también funciona.

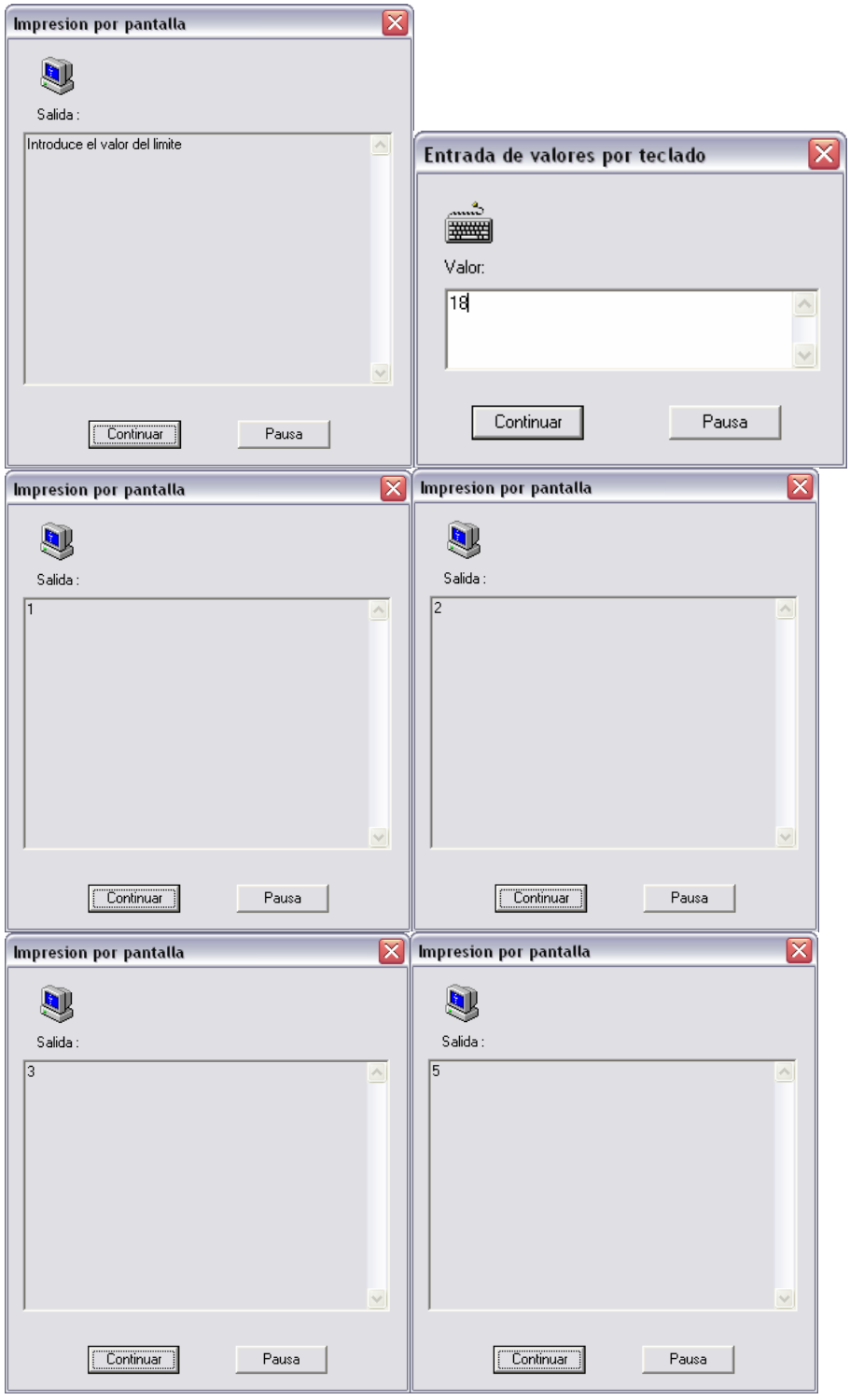
X	Y	Z	IMPRESIÓN(z+1=x)
3	1	-	
-	2	-	
-	3	1	
-	-	2	3
-	4	-	
	5	-	
4	1	-	
-	2	-	
-	3	-	
-	4	2	No impresión por que 4%2==0
-	5	-	
5	1	-	
-	2	-	
-	3	-	
-	4	-	
-	5	2	
	-	3	
	-	4	5

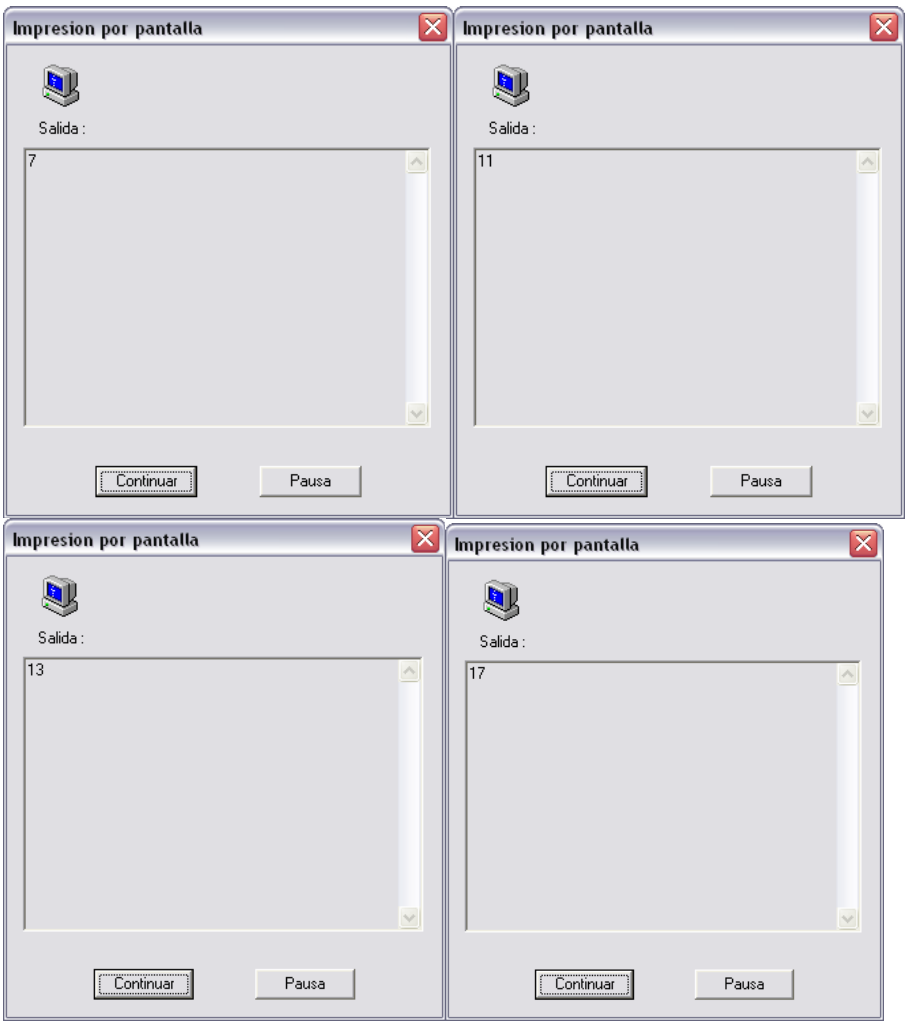
- No se mueve el valor.

Por lo tanto probáremos nuestro programa con un rango mas grande pues el análisis dice que será correcto. La corrida se ve de la siguiente forma:

Introducimos un limite de 500 y observamos que efectivamente el programa realiza su función de manera correcta.La corrida del diagrama de flujo es mucho mayor por lo que nos limitamos unos cuantos valores que son números primos metiendo un limite menor




Impresion por pantalla
Salida :
Introduce el valor del limite
Continuar
Pausa
Entrada de valores por teclado
Valor:
18
Continuar
Pausa
Impresion por pantalla
Salida :
1
Continuar
Pausa
Impresion por pantalla
Salida :
2
Continuar
Pausa
Impresion por pantalla
Salida :
3
Continuar
Pausa
Impresion por pantalla
Salida :
5
Continuar
Pausa



Conclusiones generales:

En esta práctica aprendimos a usar los comandos de textbackground que sirve para cambiar el color al fondo de pantalla el comando textcolor que le asigna un color definido a el texto en la corrida del programa el comando gotoxy que nos posiciona el texto y el cursor en una determinada área de la pantalla mediante coordenadas, el ciclo de repetición for que nos fue muy útil en los diversos programas y que junto con el comando gotoxy y delay nos forman un marco que le da presentación a nuestro trabajo, además de notar que los conocimiento y el ingenio que cada programador tiene es determinante para el programa que se obtiene.