**Instituto Politécnico Nacional**

**ESCOM**

**Práctica 4**

**Estructuras de control en Lenguaje C**

**Algoritmia y Programación Estructurada**

**Docente: Mtro. Miguel Ángel Rodríguez Castillo**

**Alumnos: García King Ricardo Neftaly**

**Vaquera Aguilera Ethan Emiliano**

**1CV4**

Índice

Planteamiento del Problema 2

Diseño y Funcionamiento de la Solución 3

Implementación de la Solución 5

Funcionamiento 8

Conclusiones y Bibliografía 12

Anexos 13

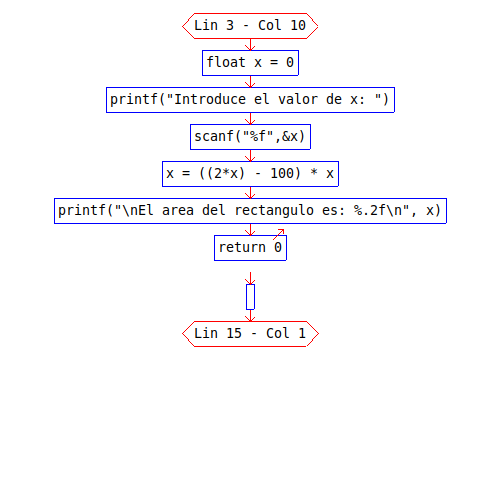
Planteamiento del Problema

1.-Desarrolla un programa que calcule el área de un terreno rectangular de lados 2x-100 y x respectivamente, para valores de x entre 50.1 y 300.

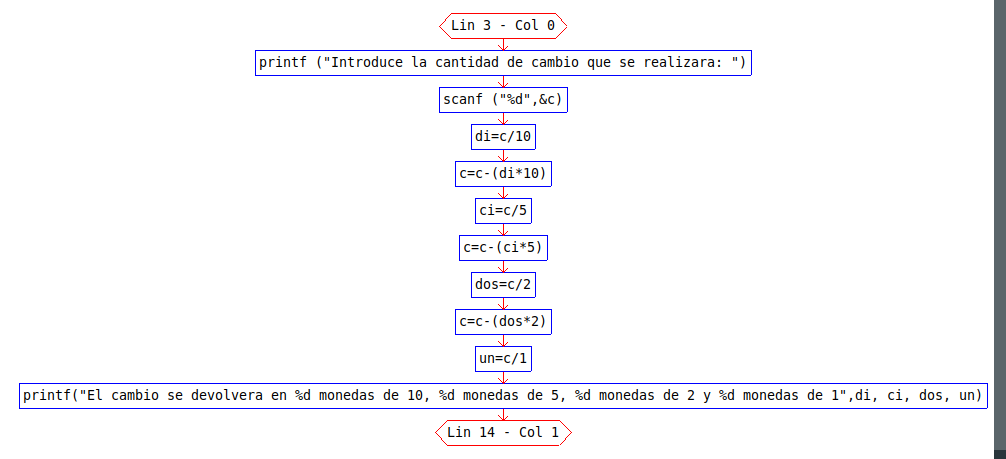
2. Calcular el número de monedas de un cambio a devolver ingresado por el usuario; si se saben los valores de denominación posibles: $10, $5, $2 y $1.\*Considerar que las monedas de cada denominación son infinitas.

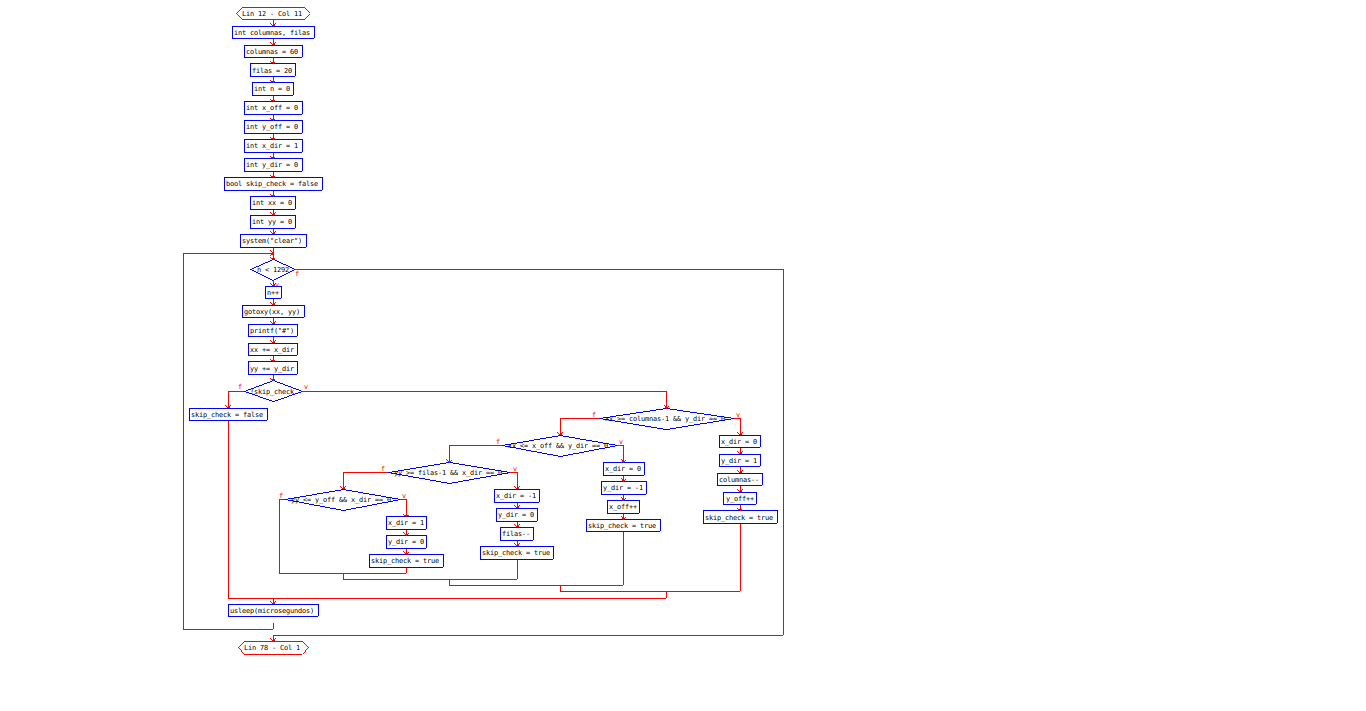
3.- Realice un programa que imprima la siguiente figura poco a poco hasta llegar al centro de la pantalla. Rectángulos concéntricos.

Diseño y funcionamiento de la solución

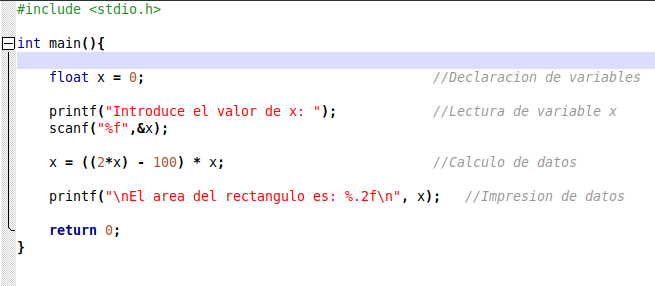


1.-

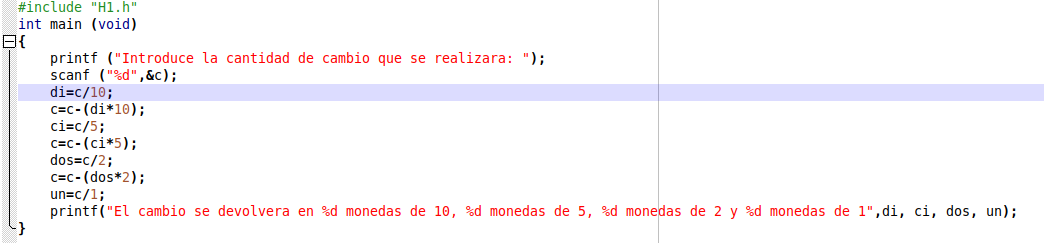
2.-

3.-

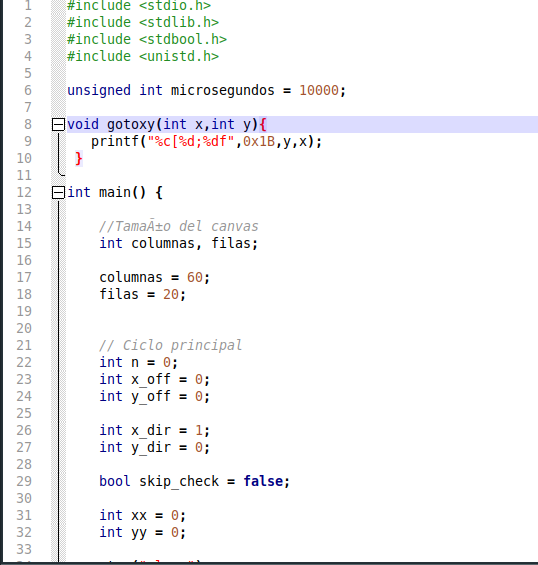
Implementación de la Solución

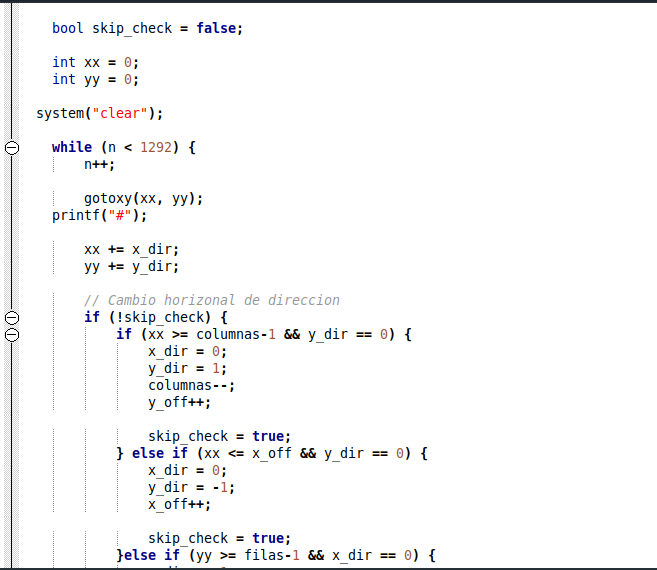


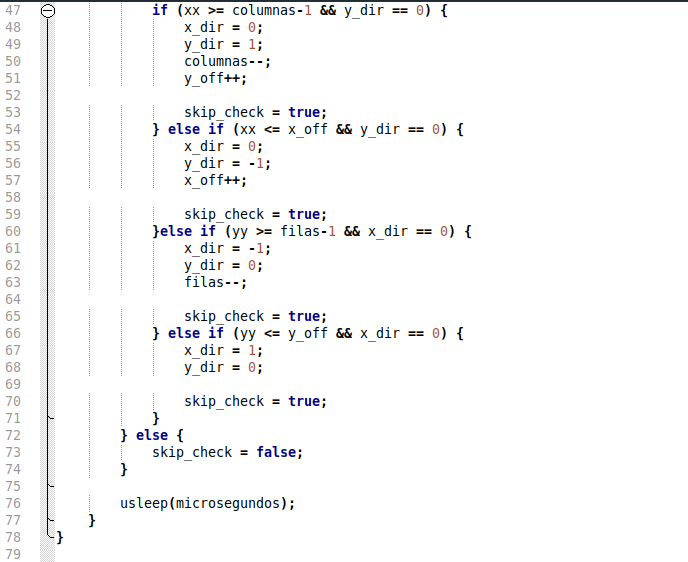
1.-

2.-

3.-







Funcionamiento

Algoritmo 1.-

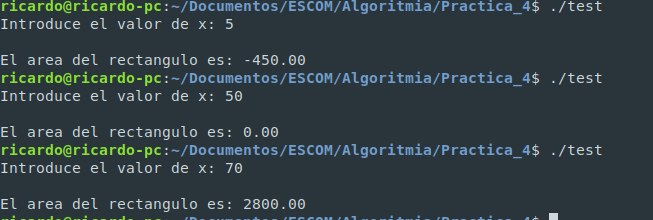
* Prueba de Escritorio

|  |  |
| --- | --- |
| **X** | **resultado** |
| 5 | -450 |
| 50 | 0 |
| 70 | 2800 |

* Resultados de Salida

|  |  |  |
| --- | --- | --- |
| **ENTRADA** | **PROCESO** | **SALIDA** |
| x | float x = 0;  printf("Introduce el valor de x: ");  scanf("%f",&x);  x = ((2\*x) - 100) \* x;  printf("\nEl area del rectangulo es: %.2f\n", x); | resultado |

* Captura de Pantalla



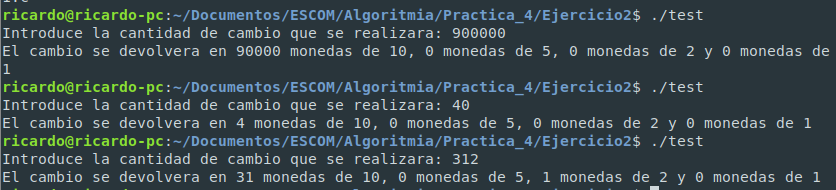
Algoritmo 2.-

* Prueba de Escritorio

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **c** | **di** | **ci** | **dos** | **un** |
| 900000 | 90000 | 0 | 0 | 0 |
| 40 | 4 | 0 | 0 | 0 |
| 312 | 31 | 0 | 1 | 0 |

* Resultados de Salida

|  |  |  |
| --- | --- | --- |
| **Entrada** | **Proceso** | **Salida** |
| c | di=c/10;  c=c-(di\*10);  ci=c/5;  c=c-(ci\*5);  dos=c/2;  c=c-(dos\*2);  un=c/1; | di, ci, dos, un |

* Captura de Pantalla

Algoritmo 3.-

* Prueba de Escritorio

--Demasiadas variables e iteraciones

* Resultado de Salida

|  |  |  |
| --- | --- | --- |
| **Entrada** | **Proceso** | **Salida** |
| -- | while (n < 1292) {  n++;    gotoxy(xx, yy);  printf("#");    xx += x\_dir;  yy += y\_dir;    if (!skip\_check) {  if (xx >= columnas-1 && y\_dir == 0) {  x\_dir = 0;  y\_dir = 1;  columnas--;  y\_off++;    skip\_check = true;  } else if (xx <= x\_off && y\_dir == 0) {  x\_dir = 0;  y\_dir = -1;  x\_off++;    skip\_check = true;  }else if (yy >= filas-1 && x\_dir == 0) {  x\_dir = -1;  y\_dir = 0;  filas--;    skip\_check = true;  } else if (yy <= y\_off && x\_dir == 0) {  x\_dir = 1;  y\_dir = 0;    skip\_check = true;  }  } else {  skip\_check = false;  } | Rectángulo concéntrico |

* Captura de Pantalla



Conclusiones

1.- Se tuvo dificultades al hacer pruebas de escritorio en el algoritmo 3 por la complejidad de sus iteraciones y la cantidad de variables

2.- Por falta de tiempo (y compatibilidad) no se alcanzó a hacer uso de la cabecera .h en el algoritmo 1 y 3

Anexos

* Se consultó documentación sobre creación de videojuegos para la lógica del tercer algoritmo, pues con un main loop se pueden dibujar animaciones e interacción con ellas, pero se tuvo un problema que hasta la fecha se desconoce, el centro de la impresión del rectángulo está hueco, se puede solucionar imprimiendo una cadena de caracteres final, pero esto no sería dinámico y por lo tanto no respetaria la lógica del programa.