INSTITUTO TECNOLOGICO SUPERIOR DE SAN ANDRÉS TUXTLA
DIVISIÓN INGENIERÍA INFORMATICA

LISTA DE COTEJO PARA EVALUAR INFORME DE INVESTIGACION

VALOR 20%

NOMBRE DE LA ASIGNATURA: INTRODUCCION A LA PROGRAMACION
NO. DE UNIDAD 2
GRUPO: 102-B
ALUMNO LUCHO CHONTAL ESMERALDAD TRINIDAD **Calificacion Obtenida :14%**

Realizar una investigación en diversas direcciones electrónicas sobre los temas: Correspondientes a de unidad 2

Realizar un REPORTE DE LA INVESTIGACION con una redacción satisfactoria. El documento debe CONTENER LOS TEMAS CORRESPONDIENTES A LA UNIDAD 2

ASIGNATURA INTRODUCCION A LA PROGRAMACION
NOMBRE DEL DOCENTE: SERGIO PELAYO VAQUERO
DATOS GENERALES DEL PROCESO DE EVALUACIÓN
NOMBRE DEL ALUMNA: LUCHO CHONTAL ESMERALDA TRINIDAD
PRODUCTO: REPORTE DE INVESTIGACION
INSTRUCCIONES DE APLICACIÓN
Revisar las actividades que se solicitan y marque con una X en los apartados “SI” cuando la evidencia se cumple; en caso contrario marque “NO”. En la columna “OBSERVACIONES” escriba indicaciones que puedan ayudar al alumno a saber cuáles son las condiciones no cumplidas, si fuese necesario.

VALOR DEL REACTIVO	CARACTERÍSTICA A CUMPLIR (REACTIVO)	CUMPLE		OBSERVACIONES
		SI	NO	
2%	Presentación El trabajo cumple con los requisitos de: **a**. Buena presentación		NO	1%
2%	**b**. No tiene faltas de ortografía	SI		1%
2%	**c**. Mismo Formato	SI		2%
2%	**d**. Misma Calidad de hoja e impresión		NO	2%
2%	**e**. Maneja el lenguaje técnico apropiado en el reporte	SI		2%
2%	El reporte incluye todos los subtemas de la unidad	SI		2%
4%	**Desarrollo**: Sigue una metodología y sustenta todos los pasos que se realizaron al aplicar los conocimientos obtenidos, es analítico y bien ordenado.		NO	2
4%	**Resultados y conclusiones**: Cumplió totalmente con el objetivo esperado, tiene aplicaciones concretas		NO	2%
20 %	*CALIFICACION*			14%

INSTITUTO TECNOLOGICO SUPERIOR DE SAN ANDRES TUXTLA

ALUMNO (A)

ESMERALDA TRINIDAD LUCHO CHONTAL

DOCENTE

SERGIO PELAYO VAQUERO

INTRODUCCION A LA PROGRAMACION

UNIDAD 2

INGENIERIA ELECTROMECANICA

102-B

EVIDENCIAS

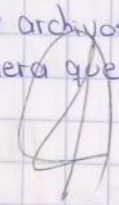
INFORME DE INVESTIGACION

F8 | 11-Oct-2022

Orígenes del lenguaje C

El lenguaje C nació en los laboratorios Bell de At and T y ha sido asociado con el sistema operativo UNIX, ya que su desarrollo se realizó en este sistema y debido a que tanto UNIX como el propio copilador C y la casi totalidad de los programas y herramientas de UNIX, fueron escritos en C.

El lenguaje C fue creado entre los años 1970 y 1972 por Brian Kernighan y Dennos Bitchie para escribir el código del sistema operativo UNIX. Fue uno de lon lenguajes de programación más aceptados por los programadores, porque hace una conjugación en lenguaje de alto nivel y lenguaje máquina. El lenguaje C es el resultado de un proceso de desarrollo que inicio con un lenguaje denominado BCPL. Lo más característico y lo que representa la genialidad de su naturaleza es que es un lenguaje muy sencillo y simple, pero con el cual se abarcan funcionalidades realmente avanzadas y notorias. Se pueden manejar archivos y realizar funciones de tipo matemático de una manera que sorprende desde el primer momento.



Norma

08 de noviembre del 2022

INTRODUCCIÓN

Los estructuras de control de un lenguaje de programación se refieren al orden en que las instrucciones de un algoritmo se ejecutarán. El orden de ejecución de las sentencias o instrucciones determinan el flujo de control.

Las tres estructuras básicas de control son:

- Secuencia.
- Selección.
- Repetición.

Un programa propio puede ser escrito utilizando las tres estructuras de control básicas (Bôhm y Jacopin (1996)).

Un programa se define como propio si cumple lo siguiente:

- Posee un sólo punto de entrada y salida o Fin para control del programa.
- Existen caminos desde la entrada hasta la salida que se pueden seguir y que pasan por todas las partes del programa.
- Todas las instrucciones son ejecutadas y no existen lazos o bucles infinitos.

ESTRUCTURA SECUENCIAL

Una estructura de programa secuencial es cuando las instrucciones se ejecutan una tras otra, a modo de secuencia lineal, es decir, las sentencias se ejecutan sucesivamente, en el orden en que aparecen. No existen "saltos" o bifurcaciones.

Ejemplo 1.

```
INPUT x
INPUT y
auxiliar = x
x = y
y = auxiliar
PRINT x
PRINT y
```

Esta secuencia de instrucciones permite los valores de x e y, con ayuda de una variable auxiliar, intermedia.

- Se guarda una copia del valor de x en auxiliar.
- Se guarda el valor de y en x, perdiendo su valor anterior, guardando una copia del contenido en auxiliar.
- Se copia a y el valor de auxiliar, es el valor inicial de x.
- El resultado es el intercambio de los valores entre x e y.

08 de noviembre del 2022

Ejemplo 2.

```
/* Calculo de raices de una ecuación de grado 2 */
#include <stdio.h>
#include <math.h>
int main (){
double a, b, c, r1, r2;
printf ("\nIntroduce coeficiente de 2o grado: ");
scanf ("%lf", &a);
printf ("\nIntroduce coeficiente de 1er grado: ");
scanf ("%lf" &b);
printf ("\nIntroduce coeficiente de 3er grado: ");
scanf ("%lf" &c);
r1 = (-b + sqrt (b*b - 4*a*c))/(2*a);
r2 = (-b - sqrt (b*b - 4*a*c))/(2*a);
printf ("Las raices son %lf y %lf \n", r1, r2);
return 0;}
```

Si bien el programa anterior es correcto, no es capaz de manejar situaciones excepcionales.

Por ejemplo, ¿qué pasa en la sentencia:

`r1=(-b+sqrt(b*b-4*a*c))/(2*a);` si

`a=0` o `b*b-4*a*c<0`?

En el primer caso, obtendríamos un error de ejecución por intentar hacer una división por cero.

Solución: Primero comprobar (mediante estructuras condicionales), y no efectuar el cálculo si no es posible.

ESTRUCTURA CONDICIONAL

Una instrucción (o estructura) condicional o de selección es aquella que establece que instrucciones deben de ejecutarse o no, en función del valor de una condición. El valor de la condición que determina si se ha de ejecutar o no un conjunto de instrucciones viene dado por un valor booleano que es el

08 de noviembre del 2022

resultado de una expresión booleana. Permite elegir entre diferentes cursos de acción en función de condiciones.
Si la nota del examen es mayor o igual que 5 mostrar "aprobado"
Si la condición es verdadera, entonces se ejecuta la sentencia **mostrar**, y luego el programa continuaría en la sentencia siguiente al si. Si la condición es falsa, la sentencia **mostrar** se ignora y el programa continúa.
Tipos de instrucciones condicionales:

- Condicional simple
  Una condicional simple es una estructura de control que ejecuta un conjunto de líneas de código si es cierta una expresión booleana. En processing un condicional simple se expresa según este código: IF (expresion) {
  Líneas de código que se ejecutan si la expresión es cierta.
  }

  Ejemplo: Si la nota del examen es mayor o igual que 5 mostrar "aprobado".
  Se traduce a C mediante una sentencia condicional simple:
  IF (nota >= 5)
  PrintF ("Aprobado")
  La forma general es:
  IF (<Condicion>) → Una expresión lógica.
  <bloque IF>; Una sentencia o conjunto de sentencias (encerradas entre {}).

- Condicional doble
  Este tipo de estructura permite implementar condicionales en los que hay dos acciones alternativas:

08 de noviembre del 2022

[default: <sentencias>]
}
En donde:
- <expresión> es una expresión entera o carácter.
- <constante 1> ó <constante 2> es un literal tipo entero o tipo carácter.
- switch sólo comprueba la igualdad.
- No debe haber dos casos (case) con la misma <constante> en el mismo switch. Si esto ocurre, sólo se ejecutan las sentencias correspondientes al ca.o que aparezca primero.
- La ejecución de sentencias empieza en la etiqueta de case cuya <constante> es igual a la <expresión>, y continúa hasta el final del switch, o hasta el siguiente break.
- El identificador especial default permite incluir un caso por defecto, que se ejecutará si no se cumple ningún otro. Se suele colocar como el último de los casos.
- En las estructuras condicionales múltiples también se permite el anidamiento.

ESTRUCTURAS REPETITIVAS

Las estructuras repetitivas son también conocidos como bucles, ciclos o lazos. Una estructura repetitiva permite la ejecución de un conjunto de sentencias:

- Hasta que se satisface una determinada condición (controladas por condición o controladas por centinela)
- Un número determinado de veces (controladas por contador)

Tipos de instrucciones repetitivas:

- Bucles controlados por condición
  Se ejecuta el conjunto de sentencias de interés mientras la condición sea verdadera.

08 de noviembre del 2022

Existen dos formas básicas de construcción:

Pre-Test

```
While (<condición>) {
    <cuerpo del bucle>
}
```

Primero se pregunta y luego se ejecuta

Post-Test

```
do {
    <cuerpo del bucle>
} while (<condición>);
```

Primero se ejecuta y luego se pregunta

- Bucles controlados por contador

Se utilizan para repetir un conjunto de sentencias un número fijo de veces. Se necesita una variable controladora, un valor inicial, un valor final y un incremento.

Ejemplo:

```
/* Hallar la media de 5 números enteros.
   Entradas: 5 números enteros (se leen en valor).
   Salidas: La media de los 5 números. */
#include <stdio.h>
int main(){
int i, valor, suma;
double media;
suma = 0;
i = 1;                          // Valor inicial del contador
while (i <= 5){                 // Prueba de límites del valor del contador
   printf("Introduce el numero %d:\n", i);
   scanf("%d", &valor);
   suma = suma + valor;
   i++;
   }
media = suma / 5.0;
printf("La media es %lf\n", media);
return 0;
}
```

08 de noviembre del 2022

CONCLUSION

Todos los modernos lenguajes de programación disponen de estructuras de control de flujo necesarias para desarrollar un programa. Las estructuras de control en la creación de algoritmos y la programación son mecanismos que permiten elegir varias opciones de ejecución o tomar las decisiones adecuadas cuando se están creando programas computacionales que le permitan a los usuarios manipular con facilidad cualquier aplicación.

Las estructuras de control de un lenguaje se refieren al orden que debe llevarse a cabo para ejecutar un algoritmo.

Las estructuras básicas de control son:

- Estructura secuencial: Al referirnos al término secuencial es cuando una sentencia se ejecuta detrás de otra, es decir, esta estructura tiene una entrada, sigue con un proceso de operación y una salida.
- Estructura condicional o selectiva, en este tipo de estructuras se utiliza una expresión lógica que va a evaluar una condición dependiendo de que resultado se obtenga, es decir, estas permiten expresar las elecciones que se hacen durante la resolución de un problema.
- Y por último las estructuras repetitivas, este tipo de estructuras es cuando el bucle se repite mientras se está cumpliendo la condición que es evaluada como verdadera y en caso de que sea evaluado como falsa no toma en cuenta ninguna instrucción del bucle y dicho algoritmo tendrá que seguir con otras instrucciones o finalizar el proceso.

INSTITUTO TECNOLOGICO SUPERIOR DE SAN ANDRES TUXTLA

ALUMNO (A)

ESMERALDA TRINIDAD LUCHO CHONTAL

DOCENTE

SERGIO PELAYO VAQUERO

INTRODUCCION A LA PROGRAMACION

UNIDAD 2

INGENIERIA ELECTROMECANICA

102-B

REPORTE DE PRACTICA

INSTITUTO TECNOLOGICO SUPERIOR DE SAN ANDRÉS TUXTLA
DIVISIÓN INGENIERÍA INFORMATICA
LISTA DE COTEJO PARA EVALUAR REPORTE DE PRACTICAS
VALOR TOTAL 40%

NOMBRE DE LA ASIGNATURA: Introducción a la Programación **Calificación Obtenida : 30%**

GRUPO 102-B

UNIDAD NO. 2

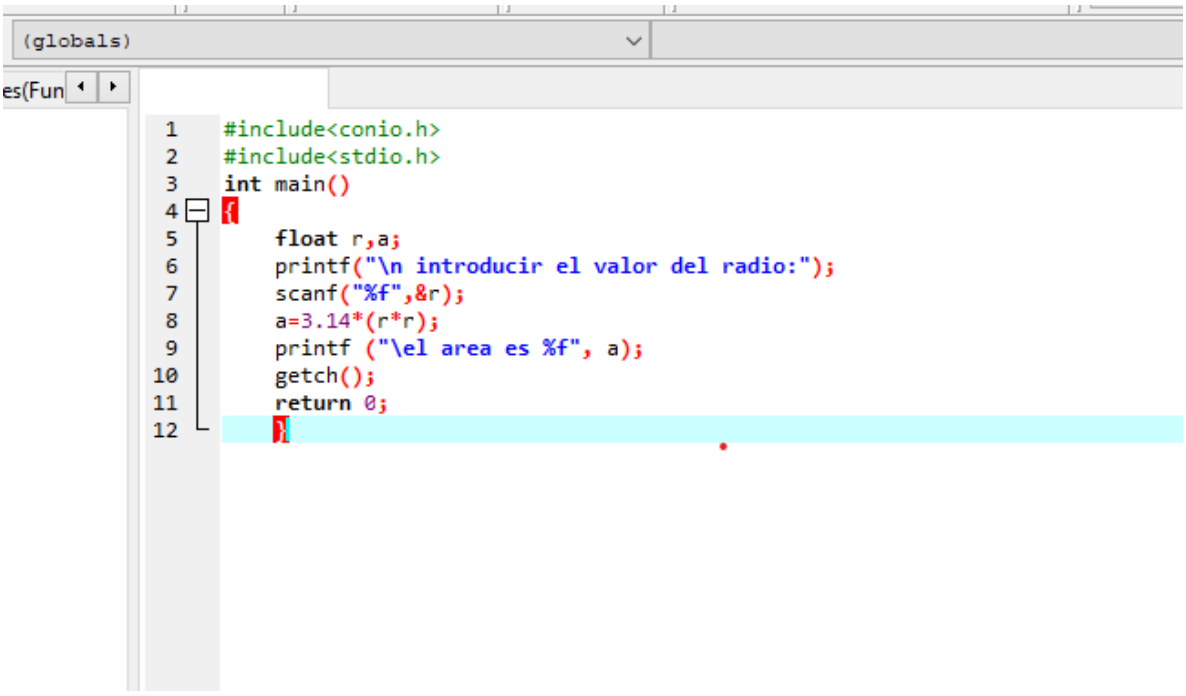
ALUMNO LUCHO CHONTAL ESMERALDA TRINIDAD

INSTRUCCIONES

Revisar los Ejercicios que se solicitan y marque en los apartados "SI" cuando la evidencia a evaluar se cumple; en caso contrario marque "NO". En la columna "OBSERVACIONES" ocúpela cuando tenga que hacer comentarios referentes a lo observado.

Valor del reactivo	*Características a cumplir (Reactivo)*	*CUMPLE*		*OBSERVACIONES*
		Si	*NO*	
8%	¿Comprendió el Planteamiento?	6%		
8 %	¿Identifico los Datos de Entrada Y/O Variables A Utilizar para solucionar los ejercicios?	6%		
8 %	¿Uso su pensamiento lógico / matemático para la solución de los ejercicios?	6 %		
8 %	¿Estructuro el procedimiento para resolver el problema de manera analítica y reflexiva ?	6 %		
8%	¿se obtuvo el resultado correcto?	6%		
40%	***CALIFICACIÓN:***	30%		

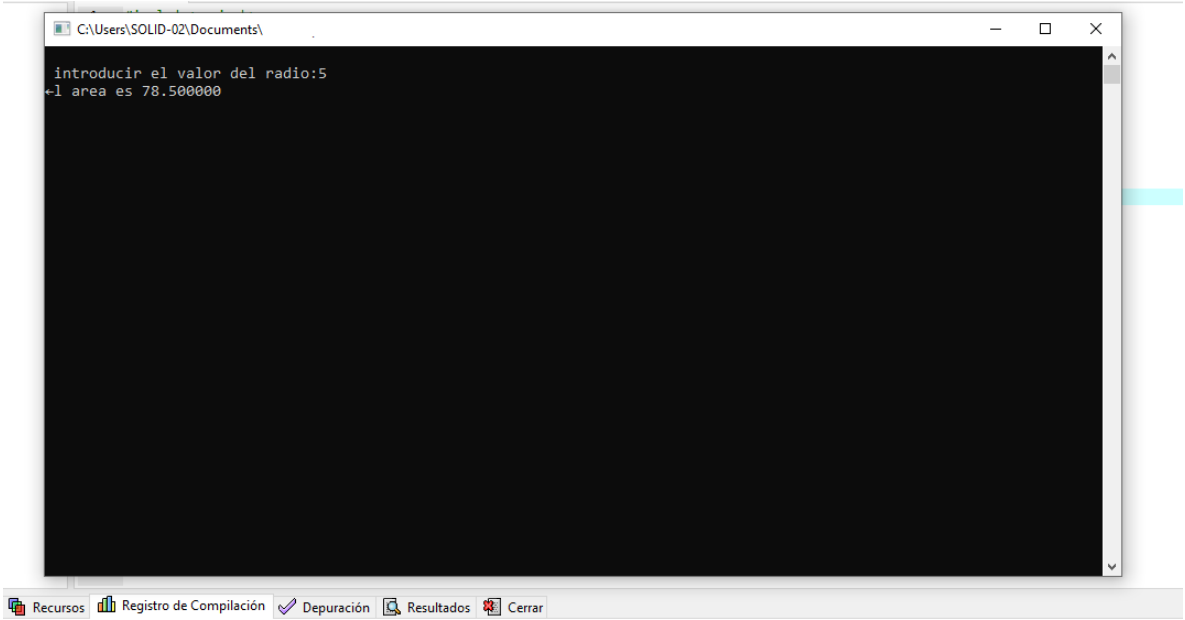
Instrucción: calcular el area de un circulo.



```
(globals)

es(Fun

1   #include<conio.h>
2   #include<stdio.h>
3   int main()
4   {
5       float r,a;
6       printf("\n introducir el valor del radio:");
7       scanf("%f",&r);
8       a=3.14*(r*r);
9       printf ("\el area es %f", a);
10      getch();
11      return 0;
12      }
```



```
C:\Users\SOLID-02\Documents\

introducir el valor del radio:5
←l area es 78.500000
```

Recursos | Registro de Compilación | Depuración | Resultados | Cerrar

INSTITUTO TECNOLOGICO SUPERIOR DE SAN ANDRES TUXTLA

ALUMNO (A)

ESMERALDA TRINIDAD LUCHO CHONTAL

DOCENTE

SERGIO PELAYO VAQUERO

INTRODUCCION A LA PROGRAMACION

UNIDAD 2

INGENIERIA ELECTROMECANICA

102-B

EXAMEN

INSTITUTO TECNOLOGICO SUPERIOR DE SAN ANDRÉS TUXTLA
DIVISIÓN DE INGENIERIA INFORMATICA
LISTA DE COTEJO PARA EVALUAR EXAMEN PRÁCTICO

NOMBRE DE LA ASIGNATURA: INTRODUCCIÓN A LA PROGRAMACION

GRUPO 102-B

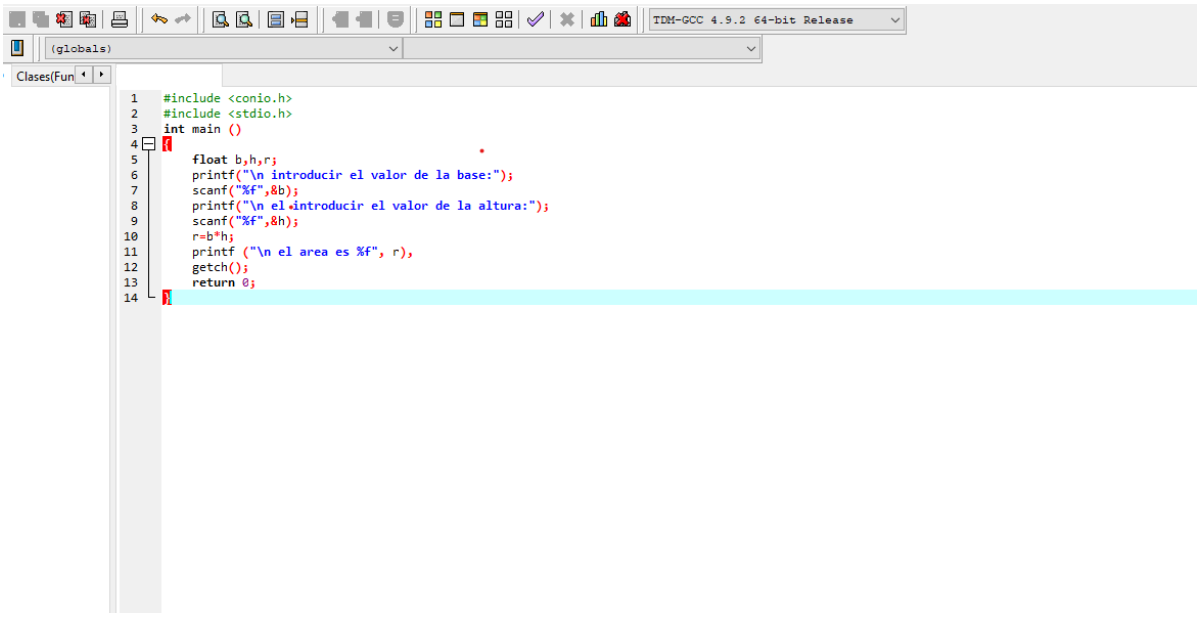
UNIDAD NO. 2

ALUMNO: ***LUCHO CHONTAL ESMERALDA TRINIDAD*** . **Calificación Obtenida : 40%**

Valor del reactivo	***Características a cumplir (Reactivo)***	***CUMPLE***		***OBSERVACIONES***
		Si	***NO***	
8%	¿Comprendió el Planteamiento?	8%		
8%	¿Identificó el número de variables a utilizar?	8%		
8%	¿Declaro correctamente los tipos de variables	8%		
8%	¿Compiló y Depuró de forma correcta el programa?	8%		
8%	¿Se ejecutó con éxito el programa?	8%		

CALIFICACION OBTENIDA :40%

Instrucción: Hacer un algoritmo en C++ que calcule el área de un rectángulo.



```
#include <conio.h>
#include <stdio.h>
int main ()
{
    float b,h,r;
    printf("\n introducir el valor de la base:");
    scanf("%f",&b);
    printf("\n el introducir el valor de la altura:");
    scanf("%f",&h);
    r=b*h;
    printf ("\n el area es %f", r),
    getch();
    return 0;
}
```