



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Alejandro Esteban Pimentel Alarcón

Asignatura: Fundamentos de Programación

Grupo: 135

No de Práctica(s): 8

Integrante(s): Torres Alcántara Alan Eliezer

*No. de Equipo de
cómputo empleado:* Somalia 42

No. de Lista o 9032

Semestre: 2020 - 1

Fecha de entrega: 16/09/2019

Observaciones:

CALIFICACIÓN: _____

Estructuras de selección

Alejandro Pimentel
profpimentel9@gmail.com

Facultad de Ingeniería
UNAM
<https://qr.go.page.link/aeaVL>



Objetivo

Elaborar programas en lenguaje C que incluyan las estructuras de selección if, if-else, switch y ternaria para la resolución de problemas básicos.

ESTRUCTURAS DE SELECCIÓN

Las estructuras de control de flujo en un lenguaje especifican el orden en que se realiza el procesamiento de datos. Las estructuras de selección (o condicionales) permiten realizar una u otra acción con base en una expresión lógica.

Las acciones posibles para realizar son mutuamente excluyentes, es decir, solo se puede ejecutar una a la vez dentro de toda la estructura.

Lenguaje C posee 3 estructuras de selección: la estructura if-else, la estructura switch y la estructura condicional o ternaria.

Estructura de control selectiva if:

En esta estructura se evalúa la expresión lógica y, si se cumple (si la condición es verdadera), se ejecutan las instrucciones del bloque que se encuentra entre las llaves de la estructura. Si no se cumple la condición, se continúa con el flujo normal del programa.

La estructura de control de flujo más simple es la estructura condicional if, su sintaxis es la siguiente:

IF

```
if (expresión_lógica) {  
    // bloque de código a ejecutar  
}
```

Estructura de control selectiva switch-case:

La estructura switch-case evalúa la variable que se encuentra entre paréntesis después de la palabra reservada switch y la compara con los valores constantes que posee cada caso (case). Los tipos de datos que puede evaluar esta estructura son enteros, caracteres y enumeraciones.

Al final de cada caso se ejecuta la instrucción break, si se omite esta palabra reservada se ejecutaría el siguiente caso, es decir, se utiliza para indicar que el bloque de código a ejecutar ya terminó y poder así salir de la estructura. Si la opción a evaluar no coincide dentro de algún caso, entonces se ejecuta el bloque por defecto (default).

El bloque por defecto normalmente se escribe al final de la estructura, pero se puede escribir en cualquier otra parte. Si se escribe en alguna otra parte el bloque debe terminar con la palabra reservada break.

La sintaxis de la estructura switch-case es la siguiente:

SWITCH

```
switch (opcion_a_evaluar){
    case valor1:
        /* Código a ejecutar*/
        break;
    case valor2:
        /* Código a ejecutar*/
        break;
    ...
    case valorN:
        /* Código a ejecutar*/
        break;
    default:
        /* Código a ejecutar*/
}
```

Estructura de control selectiva condicional;

Consta de tres partes, una condición y dos acciones a seguir con base en la expresión condicional. Si la condición se cumple (es verdadera) se ejecuta la instrucción que se encuentra después del símbolo '?'; si la condición no se cumple (es falsa) se ejecuta la instrucción que se encuentra después del símbolo ':'.

La estructura condicional (también llamado operador ternario) permite realizar una comparación rápida. Su sintaxis es la siguiente:

Condicional

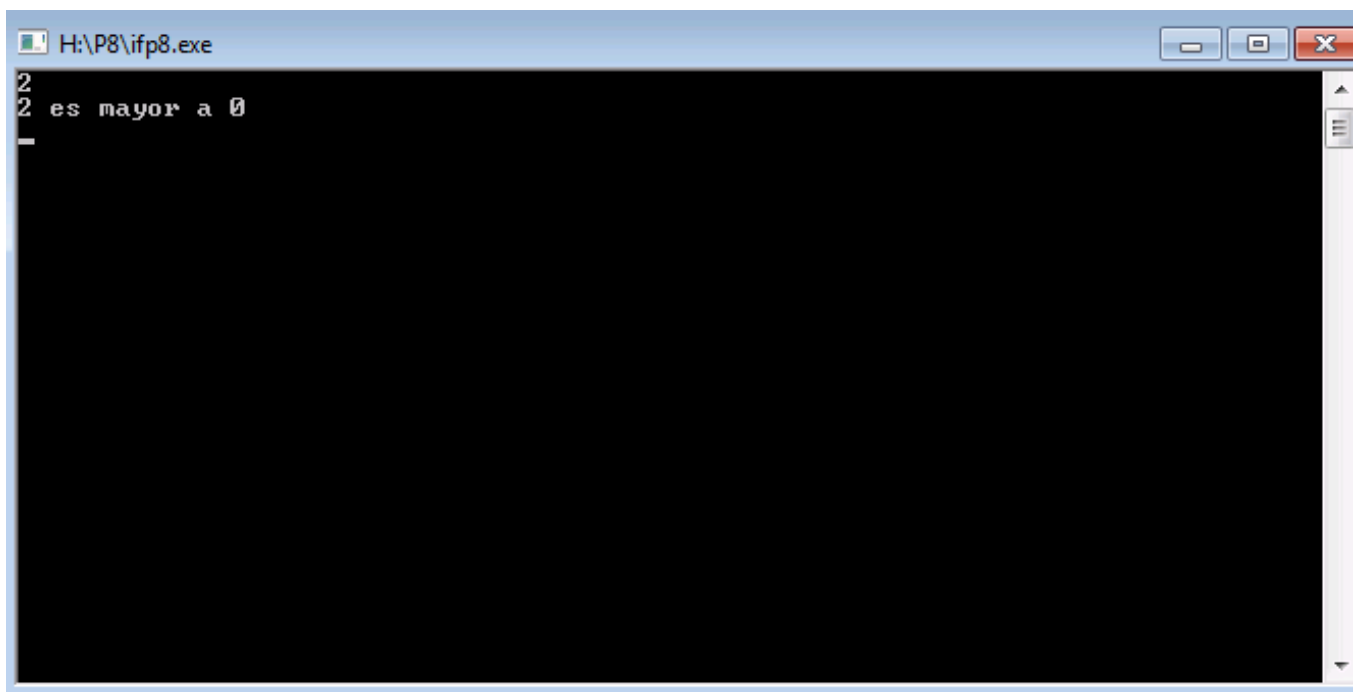
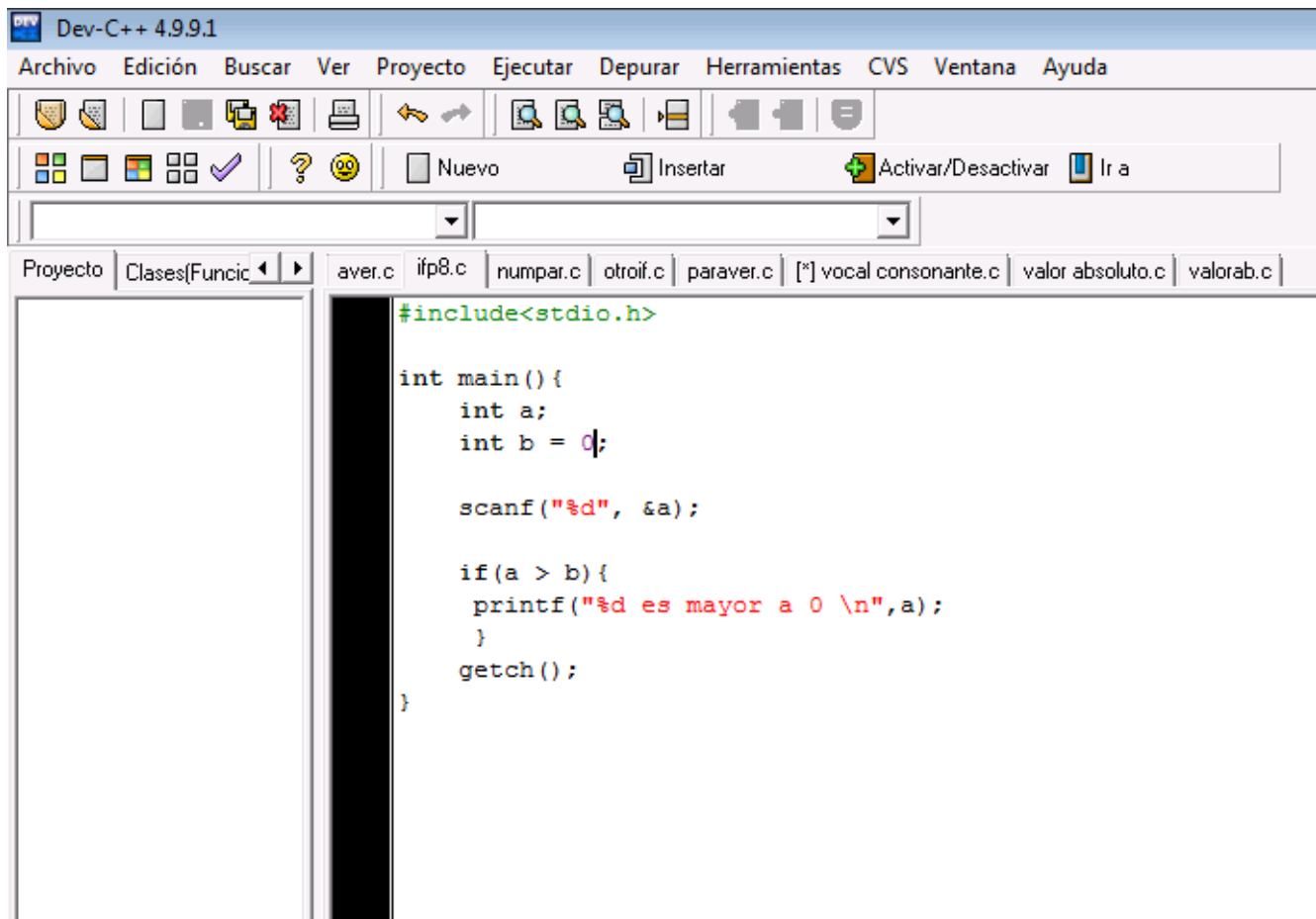
Condición ? SiSeCumple : SiNoSeCumple ;

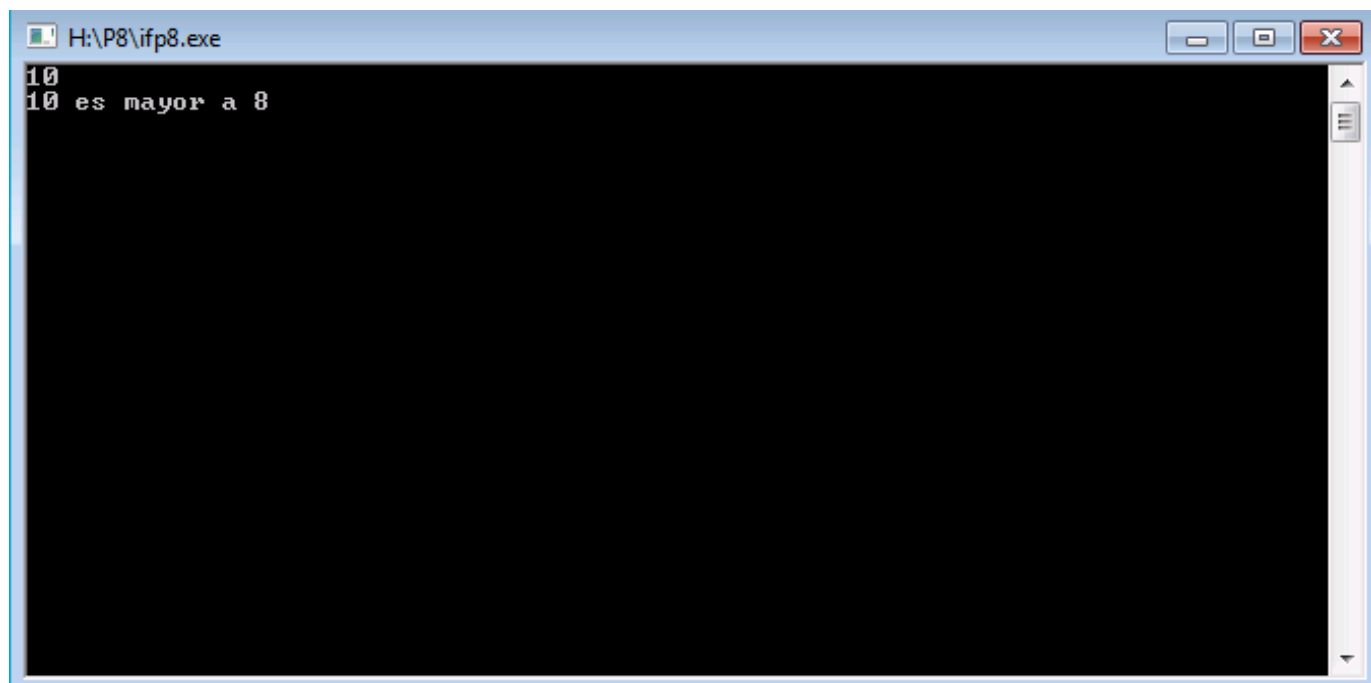
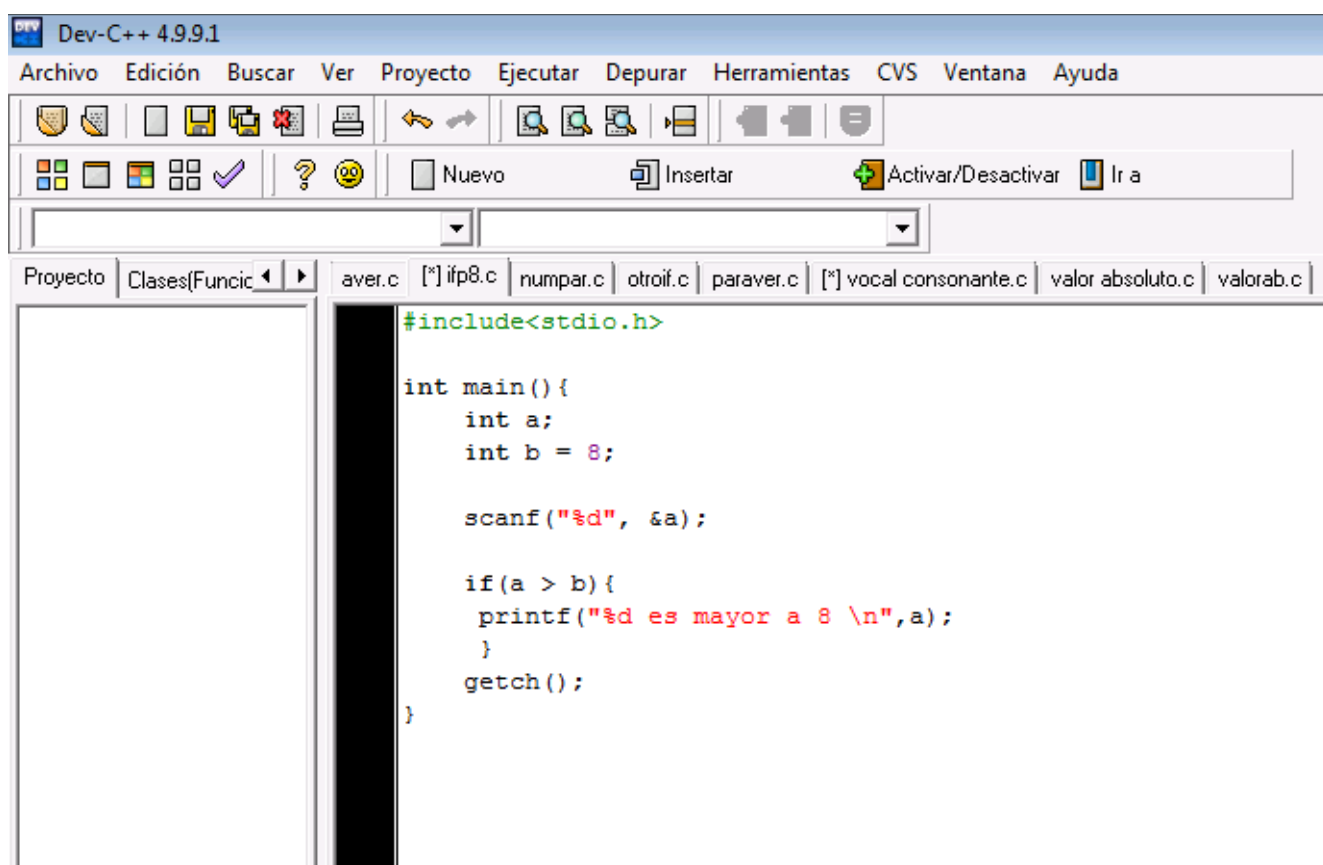
```
int a = 3;  
int b = 4;
```

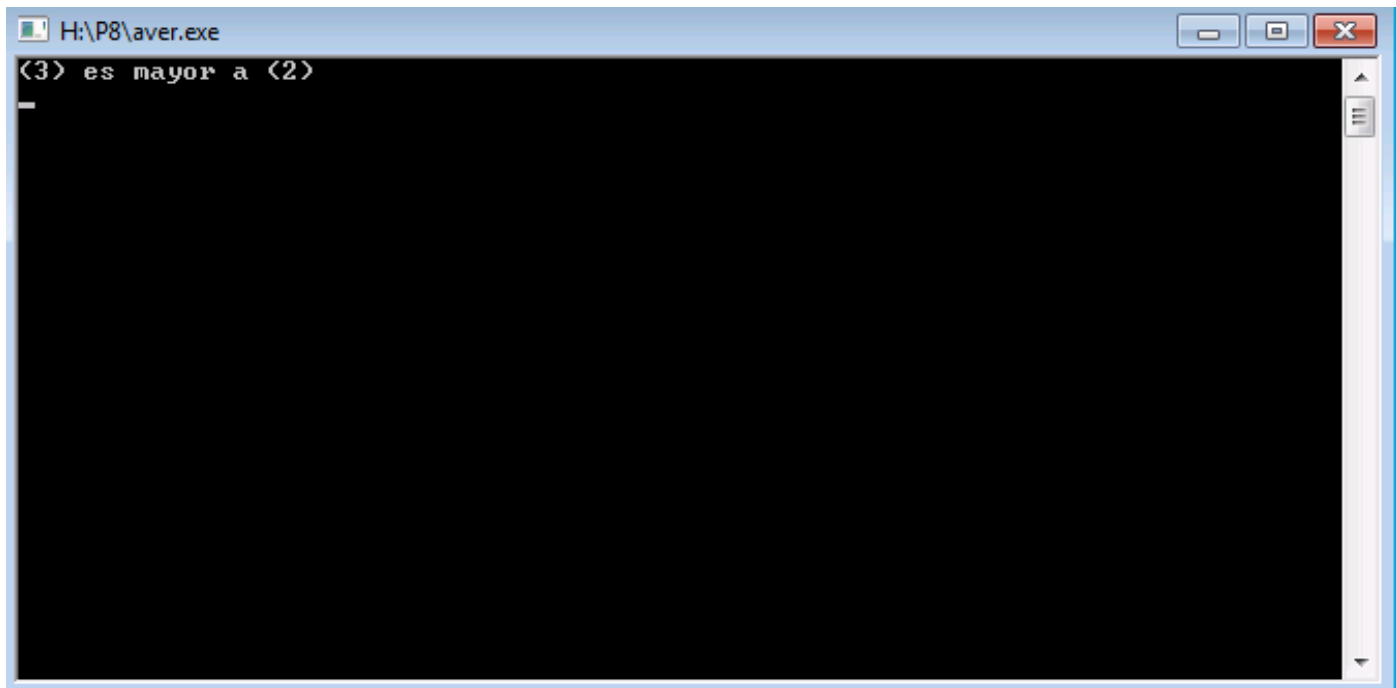
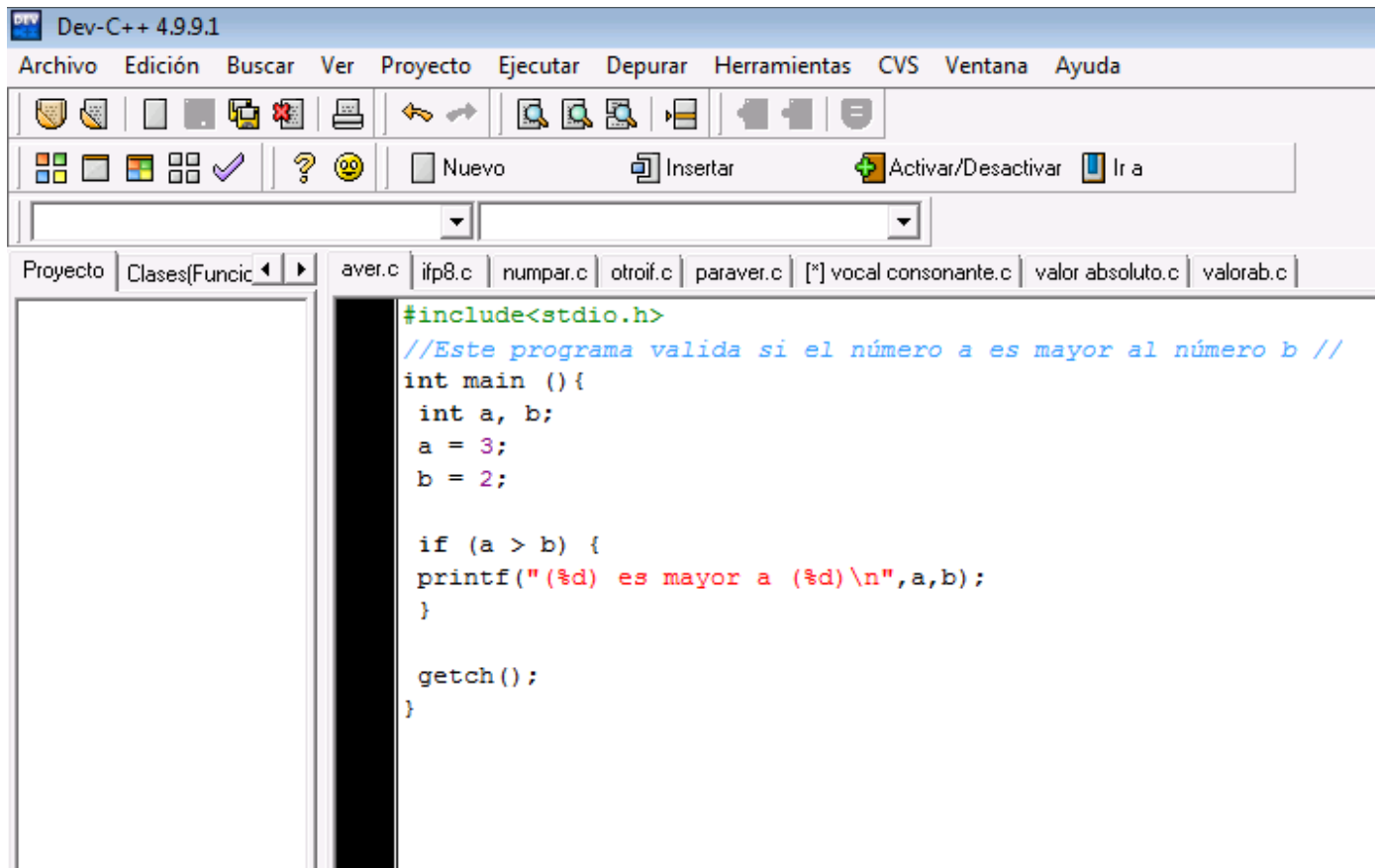
```
mayor = a > b ? a : b ;
```

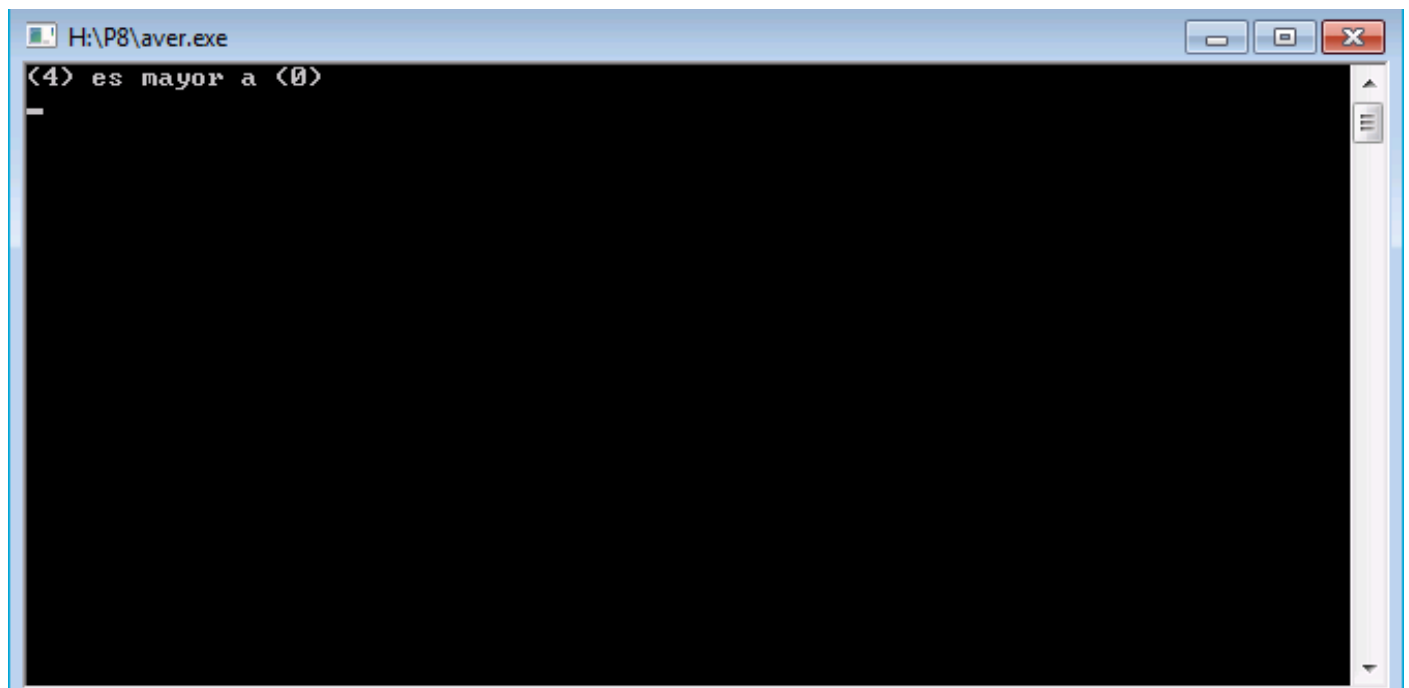
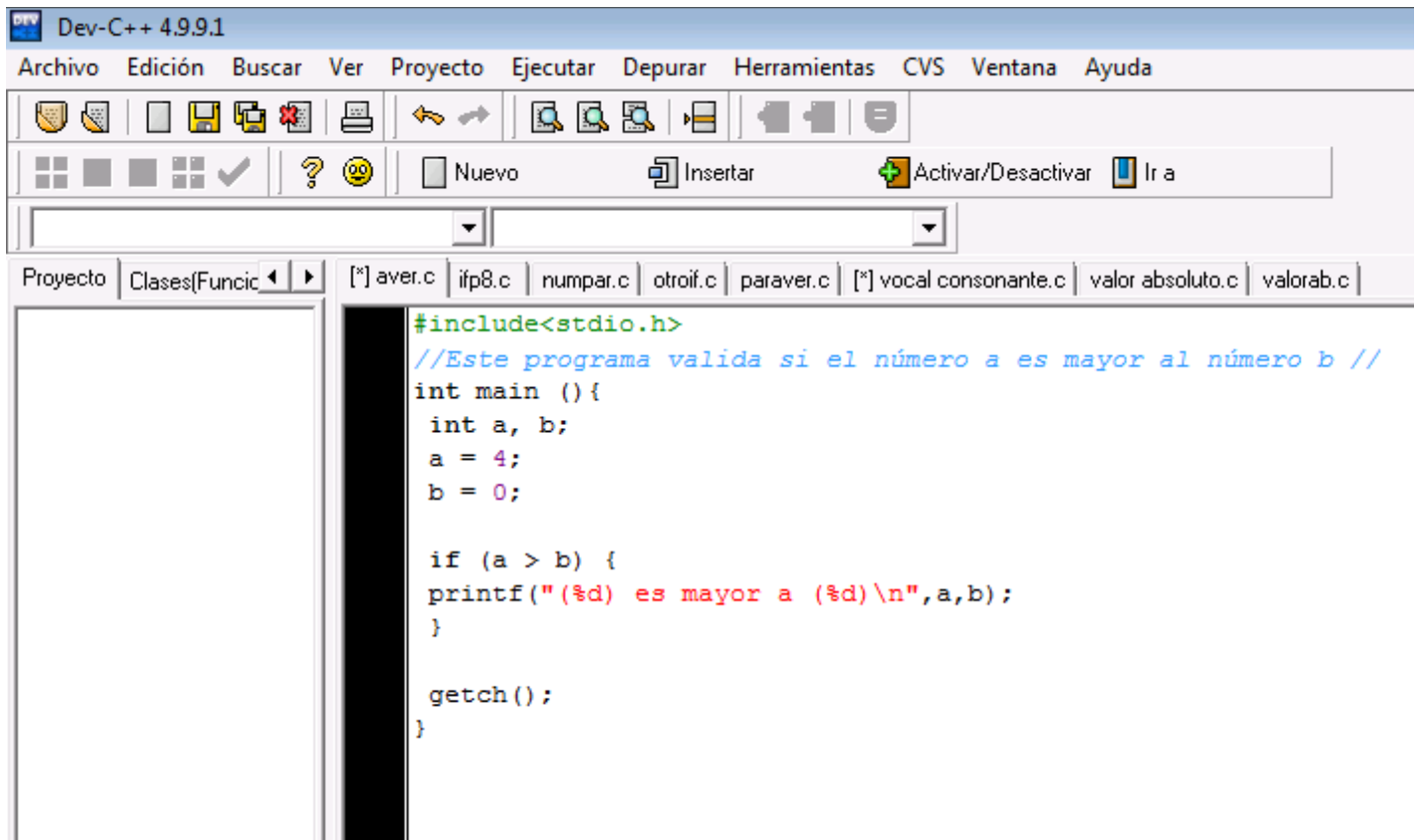
Actividades

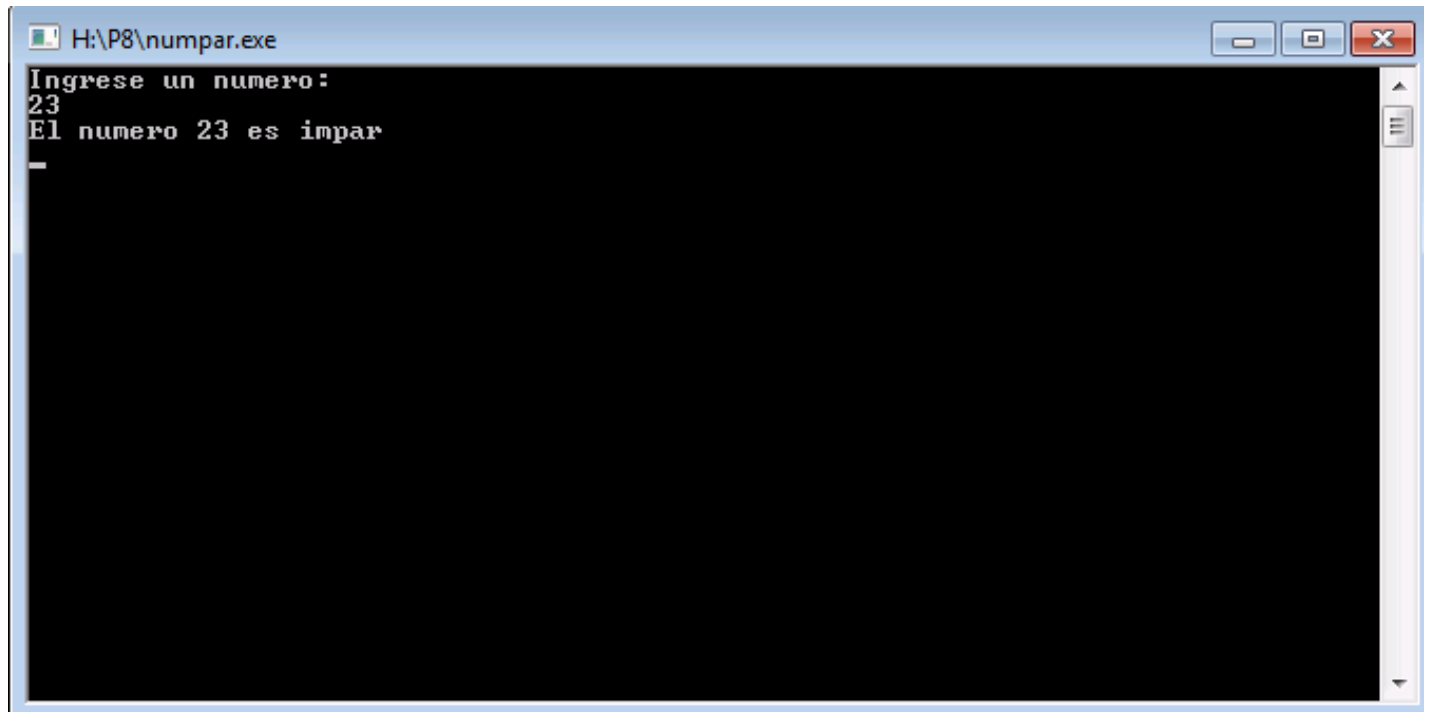
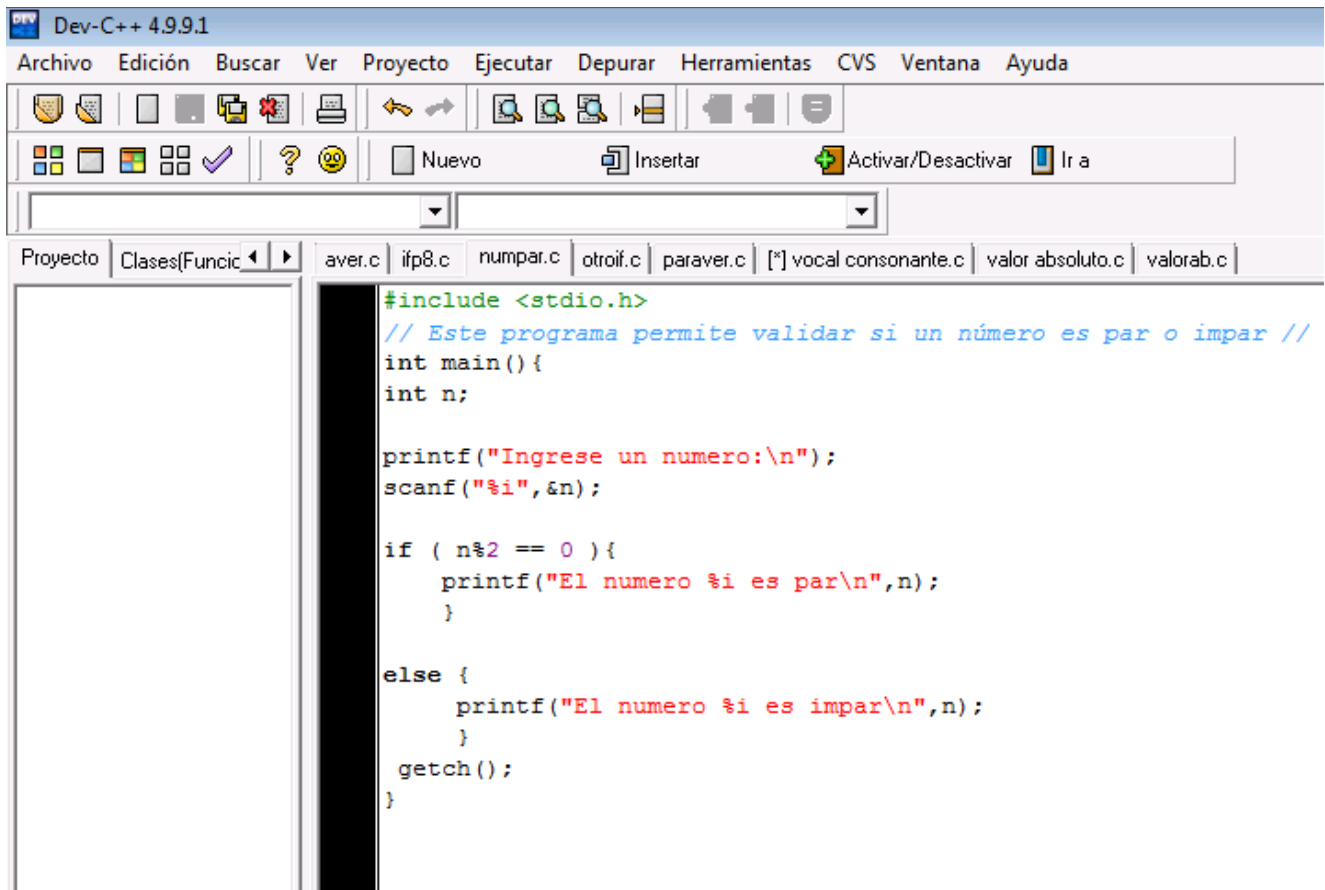
- Experimentar al utilizar diferentes números dentro del condicional del "if" (asegúrense de usar el 0 entre sus elecciones).
- Hacer un programa que lea un número e indique si es par o non.





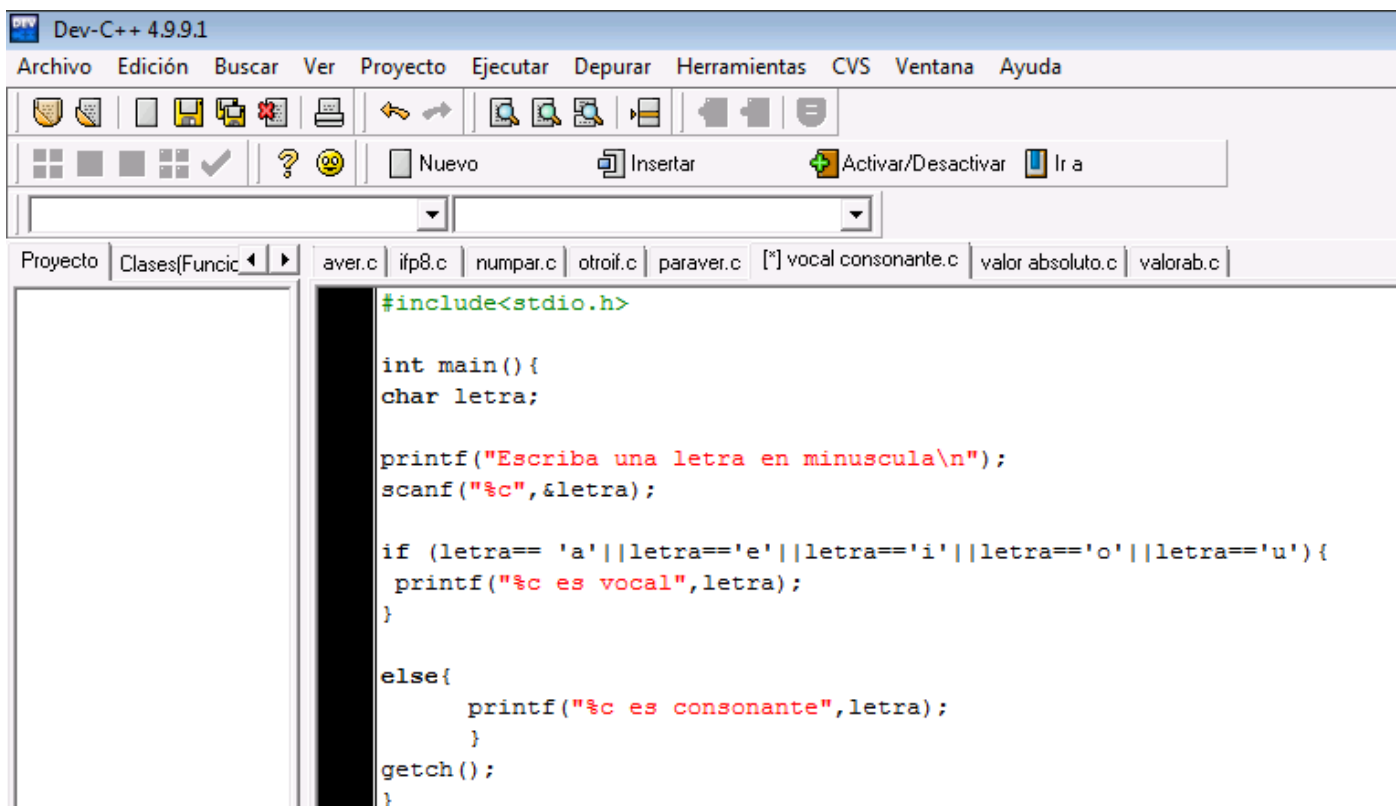






Actividades

- Hacer un programa que lea una letra e indique si es vocal o consonante.



The screenshot shows the Dev-C++ 4.9.9.1 IDE interface. The menu bar includes Archivo, Edición, Buscar, Ver, Proyecto, Ejecutar, Depurar, Herramientas, CVS, Ventana, and Ayuda. The toolbar contains various icons for file operations and execution. The project explorer on the left shows a project named 'Proyecto' with several files listed: aver.c, ifp8.c, numpar.c, otroif.c, paraver.c, '[*] vocal consonante.c', valor absoluto.c, and valorab.c. The main editor window displays the following C code:

```
#include<stdio.h>

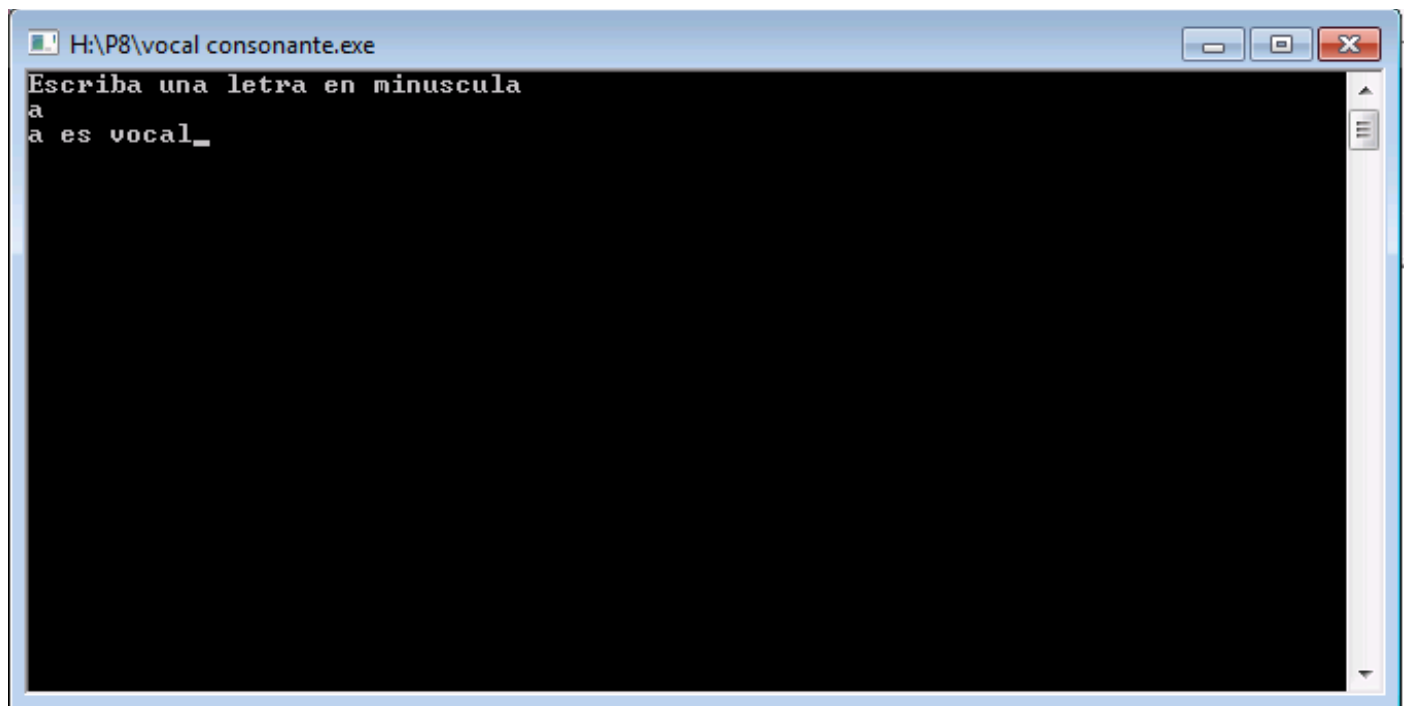
int main(){
char letra;

printf("Escriba una letra en minuscúla\n");
scanf("%c",&letra);

if (letra== 'a' || letra=='e' || letra=='i' || letra=='o' || letra=='u') {
printf("%c es vocal",letra);
}

else{
printf("%c es consonante",letra);
}

getch();
}
```




```
H:\P8\vocal consonante.exe
Escriba una letra en minuscula
a
a es vocal_
```

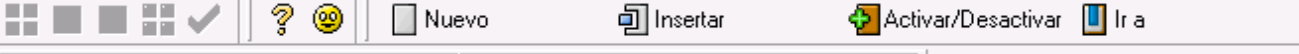
Actividades

- Usar condicional para hacer un programa que obtenga el valor absoluto de la diferencia entre dos números.

Dev-C++ 4.9.9.1

Archivo Edición Buscar Ver Proyecto Ejecutar Depurar Herramientas CVS Ventana Ayuda



 Nuevo Insertar Activar/Desactivar Ir a

Proyecto Clases(Funcion) aver.c ifp8.c numpar.c otroif.c paraver.c [*] vocal consonante.c valor absoluto.c valorab.c

```
#include<stdio.h>

int main(){
int a,b,c;

printf("En base a la formula del valor absoluto |a-b|\n");
printf("Ingrese primer numero:\n");
scanf("%i",&a);
printf("Ingrese segundo numero:\n");
scanf("%i",&b);

c = a-b > 0 ? a-b : b-a ;
printf("La diferencia en valor absoluto es %i",c);
getch();

}
```

H:\P8\valorab.exe

```
En base a la formula del valor absoluto |a-b|
Ingrese primer numero:
23
Ingrese segundo numero:
56
La diferencia en valor absoluto es 33
```

Conclusión

Las estructuras de control nos ayudan a diseñar programas con C, es importante conocer el formato con el que se emplean, los empleados en esta práctica fueron IF, SWITCH y el CONDICIONAL. El lenguaje en C tiene muchas restricciones, por lo que hay que ser cuidadoso en los puntos y coma, paréntesis, comillas, apóstrofes y llaves.

En la programación, hay muchas formas de llegar al resultado, es decir, se pueden utilizar diversas estructuras de control para lograr el objetivo. Sin embargo, podemos evitar líneas de código innecesario al identificar cuál es la mejor estructura que debemos utilizar para cierta situación. No es lo mismo programar para 2 casos que para 20 casos posibles.