

Actividad 2 - Aplicación 1.

Lenguajes de programación 1

Ingeniería en Desarrollo de Software

Tutor: Elizabeth Guevara Roa

Alumno: Fernando Pedraza Garate

Fecha: 08 de Julio del 2022

Índice

Etapa 1 – Investigación en lenguaje C++

- Introducción. pág. 3
- Investigación. pág. 4-7
 - ¿Qué es C++?
 - ¿Crees que es muy útil utilizar este lenguaje?, ¿por qué?
 - ¿Qué son los objetos en C++?
 - ¿Qué son las clases en C++?
 - ¿Cómo se utiliza cada uno? Escribe ejemplos

Etapa 2 – Aplicación para cálculo de RFC

- Introducción. pág. 8
- Codificación. pág. 9-11
- Prueba del sistema. pág. 12-27
- Conclusión. pág. 28
- Referencias. pág. 29

Introducción

Origen de C++

Fue diseñado a mediados de los años 80 por el danés **Bjarne Stroustrup**. Su intención fue la de extender el lenguaje de programación C para que tuviese los mecanismos necesarios para manipular objetos. Por lo tanto C++ contiene los paradigmas de la programación estructurada y orientada a objetos, por lo que se le conoce como un lenguaje de programación multiparadigma.

En futuras actividades se crearán distintos programas en lenguaje C++, por lo que es necesario tener un amplio conocimiento sobre el mismo para la realización de futuros sistemas informáticos que nos ayudarán a resolver problemas.

Investigación.

¿Qué es C++?

A “C++ “ primero se le conoció como “C con clases”. Luego se cambió a “C++” que significa “incremento de C”, dando a entender que se trata de una extensión del lenguaje de programación C.

¿Crees que es muy útil utilizar este lenguaje?, ¿por qué?

Si, por que en la actualidad las aplicaciones del lenguaje C++ son muy extensas. Podemos nombrar que navegadores WEB, Sistemas operativos, Bases de datos, bibliotecas, aplicaciones gráficas, nubes, videojuegos, compiladores, y más están escritos, o tienen bastante de su estructura, programada en C ++.

¿Qué son los objetos en C++?

Un objeto es una instancia de una clase, es decir una entidad que se construye a partir de las descripciones consignadas en una clase (datos y funciones). Por tanto, un objeto se puede entender como una "variable" que se declara del tipo de dato de cierta clase. Un objeto es como tal la entidad tangible que permite acceder a los datos y funciones modeladas al interior de la clase. Dicho de otro modo, un *objeto de función*, o *functor*, es cualquier tipo que implementa

operator()). Este operador se conoce como el *operador de llamada* o a veces el *operador de la aplicación*. La biblioteca estándar de C++ usa objetos de función principalmente como criterios de ordenación para los contenedores y en algoritmos.

Los objetos de función proporcionan dos ventajas principales en comparación con una llamada de función sencilla. La primera es que un objeto de función puede contener el estado. La segunda es que un objeto de función es un tipo y, por tanto, puede usarse como un parámetro de plantilla.

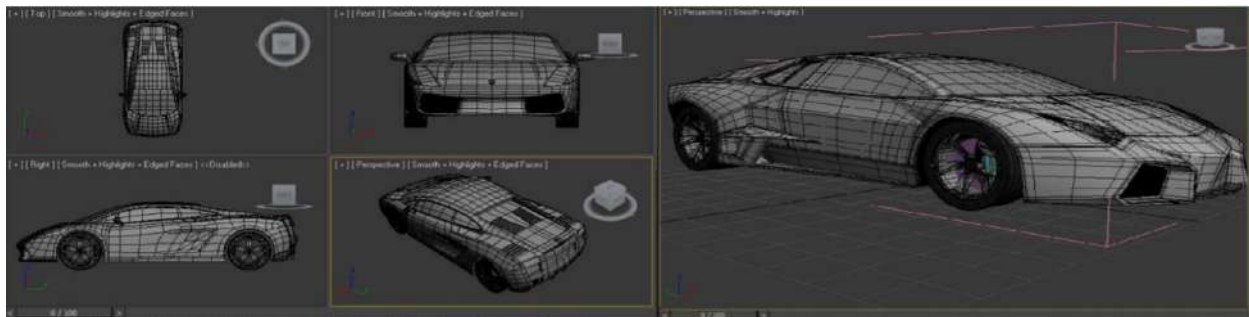
¿Qué son las clases en C++?

Una clase es en general un modelo, receta o plantilla que define el estado y comportamiento de cierto tipo de objetos. Una clase puede pensarse como una colección de variables (atributos o propiedades) y funciones (métodos) que permiten representar un conjunto de datos y especificar las operaciones o procedimientos que permiten manipular tales datos. Se puede inclusive entender una clase como un tipo de dato personalizado, similar a las estructuras (structs), donde cada programador define los miembros que va a tener su tipo de dato. De hecho, los tipos de dato nativos de C++ son en realidad clases.

¿Cómo se utiliza cada uno? Escribe ejemplos

Una analogía para entender las clases y los objetos puede ser una fábrica ensambladora de carros. Hay un modelo o diseño (clase) específico de un auto, pero este modelo en si no es un carro, es solo una descripción de que características y funcionalidades deben tener los carros que sean de ese modelo. Los carros ensamblados en la fábrica de acuerdo a dicho modelo serían los objetos, es decir entidades tangibles que se construyeron a partir de las descripciones y especificaciones consignadas en el diseño o modelo (o sea la clase).

Clase:



Objetos:



Para declarar una clase en C++ se utiliza la palabra reservada `class`, se da un nombre a la clase y luego entre llaves se declaran los miembros de la clase. Las clases no pueden declararse al interior de funciones, ya que son una definición de un tipo de dato creado por el usuario (programador). En general, las clases se declaran en bibliotecas (librerías) individuales cuyo nombre es usualmente el mismo nombre de la clase.

```
class MiClase
{
    //Aquí van los miembros de la clase: Variables y funciones
}; //NO olvidar el ;
```

Los objetos, tal como se había mencionado con anterioridad, son variables (instancias) del tipo de dato definido por una clase. Por tanto, los objetos se pueden declarar al interior o por fuera de funciones, tal y como una variable local o global respectivamente. Pueden ser declarados como miembros de otras clases, es decir al interior de otras clases. Luego, para declarar un objeto primero se utiliza el nombre de la clase a la que pertenece el objeto seguido de un nombre para el objeto y de una lista opcional de inicialización entre paréntesis. Dicha lista se verá más adelante.

```
MiClase objetoGlobal; //Declaración de un objeto global de la clase MiClase
```

```
int main()
{
    MiClase objetoLocal; //Declaración de un objeto local de la clase
    MiClase
}
```

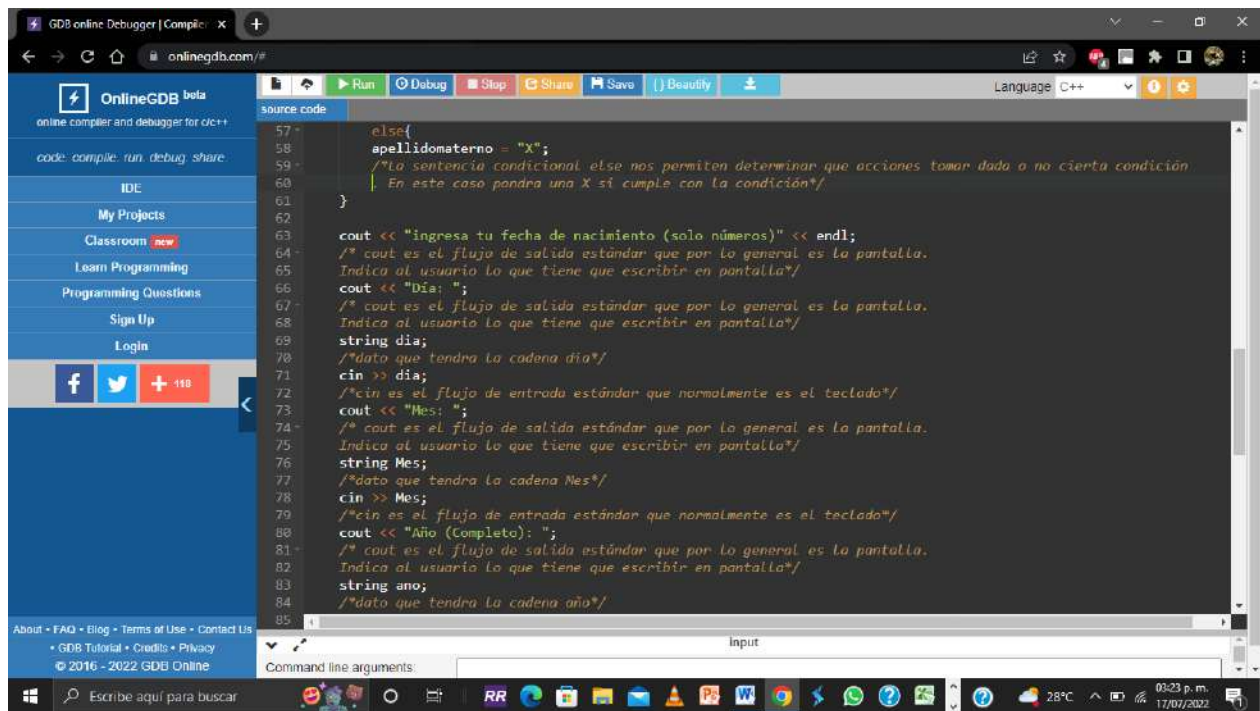
Introducción.

Se necesita crear un programa que nos permita calcular el RFC de los nuevos empleados de Coppel. Este debe generarse a partir de la captura de nombre, apellido paterno, apellido materno y fecha de nacimiento.

Codificación.

```
1  /*Programa para obtener el RFC sin homoclave*/
2  /*Fernando Pedraza Garate*/
3  /*Lenguajes de programación 1*/
4  /*Tutor: Elizabeth Guevara Roa*/
5
6  #include <iostream>
7  /*componente de la biblioteca standar STD*/
8  #include <stdio.h>
9  /* es el archivo de cabecera que contiene
10 las definiciones de las macros, las constantes, las declaraciones
11 de funciones de la biblioteca estándar */
12 #include <string.h>
13 /*string.h es un archivo de la Biblioteca estándar del lenguaje de programación*/
14 using std::string;
15 /* La clase String almacena los caracteres como una secuencia de bytes con la
16 funcionalidad de permitir el acceso al carácter de un solo byte.*/
17 using namespace std;
18 /*Permite tener un alcance directo a todas las funciones declaradas en un espacio de nombres "std"*/
19 string nombre, apellidopaterno, apellidomaterno, dia, Mes, ano, respuesta;
20 /*nombres de las cadenas*/
21 int main()
22 /*representa el punto de inicio de la ejecución.*/
23 {
24     cout << "Calculo de RFC sin homoclave" << endl;
25     /*Muestra el título del programa*/
26     cout << "Introduce tu nombre: ";
27     /* cout es el flujo de salida estándar que por lo general es la pantalla.
28 Indica al usuario lo que tiene que escribir en pantalla*/
29     string nombre;
30     /*nombre que tendrá la cadena nombre*/
31 }
```

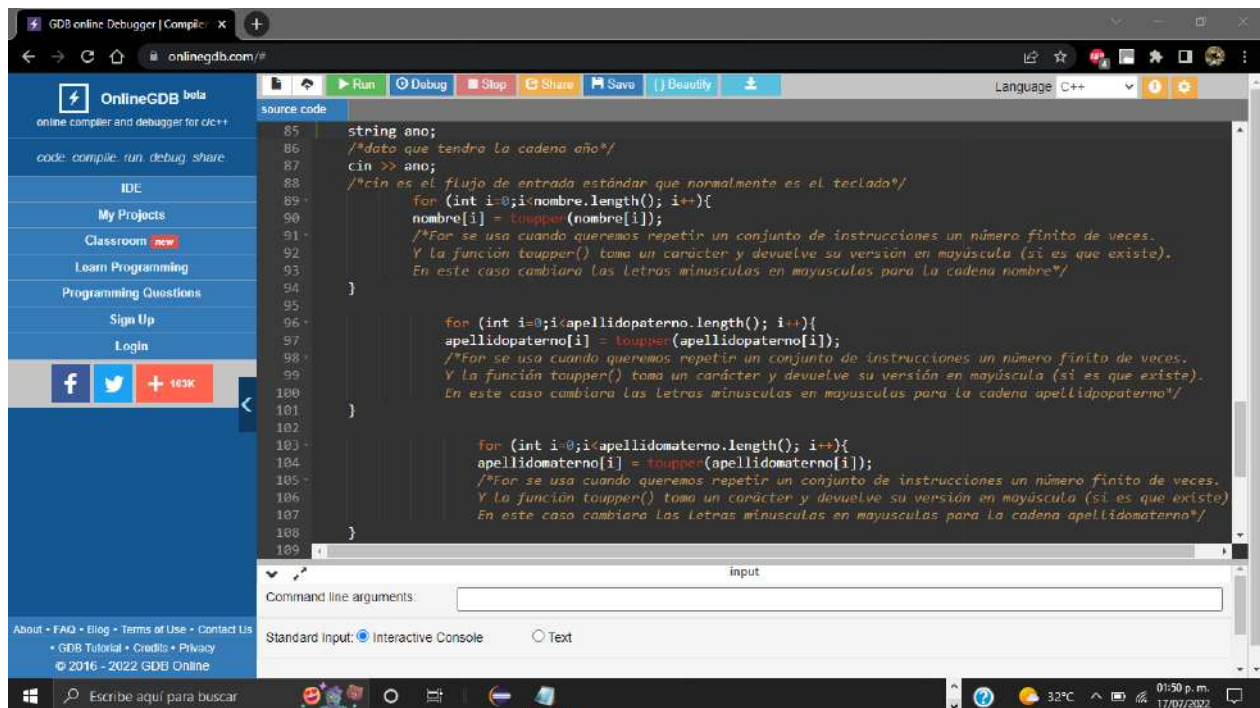
```
31     cin >> nombre;
32     /*cin es el flujo de entrada estándar que normalmente es el teclado*/
33     cout << "Introduce tu apellido paterno: ";
34     /* cout es el flujo de salida estándar que por lo general es la pantalla.
35 Indica al usuario lo que tiene que escribir en pantalla*/
36     string apellidopaterno;
37     /*nombre que tendrá la cadena apellidopaterno*/
38     cin >> apellidopaterno;
39     /*cin es el flujo de entrada estándar que normalmente es el teclado*/
40     cout << "Si tienes apellido materno, escribe una S, si es que no, escribe una N: ";
41     /* cout es el flujo de salida estándar que por lo general es la pantalla.
42 Indica al usuario lo que tiene que escribir en pantalla*/
43     cin >> respuesta;
44     /*cin es el flujo de entrada estándar que normalmente es el teclado*/
45
46     if(respuesta=="S"){
47         /*La sentencia condicional if se usa para tomar decisiones, esta evalúa
48 básicamente una operación lógica, es decir una expresión que da como
49 resultado falso o verdadero, y ejecuta la pieza de código siguiente siempre
50 y cuando el resultado sea verdadero.*/
51         cout << "Introduce tu apellido materno: ";
52         /* cout es el flujo de salida estándar que por lo general es la pantalla.
53 Indica al usuario lo que tiene que escribir en pantalla*/
54         cin >> apellidomaterno;
55         /*cin es el flujo de entrada estándar que normalmente es el teclado*/
56     }
57 }
```



The screenshot shows the OnlineGDB website interface. On the left is a sidebar with navigation links: "OnlineGDB beta", "code: compile, run, debug, share", "IDE", "My Projects", "Classroom new", "Learn Programming", "Programming Questions", "Sign Up", and "Login". Below these are social media icons for Facebook, Twitter, and a "+198" button. The main area displays C++ source code in a dark-themed editor. The code includes comments in Spanish and uses `cout` and `cin` to handle input and output. The code is as follows:

```
57-         else{
58-             apellidomaterno = "X";
59-             /*La sentencia condicional else nos permiten determinar que acciones tomar dada a no cierta condición
60-             | En este caso pondra una X si cumple con la condición*/
61-         }
62-
63-         cout << "ingresa tu fecha de nacimiento (solo números)" << endl;
64-         /* cout es el flujo de salida estándar que por lo general es la pantalla.
65-         Indica al usuario lo que tiene que escribir en pantalla*/
66-         cout << "Día: ";
67-         /* cout es el flujo de salida estándar que por lo general es la pantalla.
68-         Indica al usuario lo que tiene que escribir en pantalla*/
69-         string dia;
70-         /*dato que tendra la cadena día*/
71-         cin >> dia;
72-         /*cin es el flujo de entrada estándar que normalmente es el teclado*/
73-         cout << "Mes: ";
74-         /* cout es el flujo de salida estándar que por lo general es la pantalla.
75-         Indica al usuario lo que tiene que escribir en pantalla*/
76-         string Mes;
77-         /*dato que tendra la cadena Mes*/
78-         cin >> Mes;
79-         /*cin es el flujo de entrada estándar que normalmente es el teclado*/
80-         cout << "Año (Completo): ";
81-         /* cout es el flujo de salida estándar que por lo general es la pantalla.
82-         Indica al usuario lo que tiene que escribir en pantalla*/
83-         string ano;
84-         /*dato que tendra la cadena año*/
85-
```

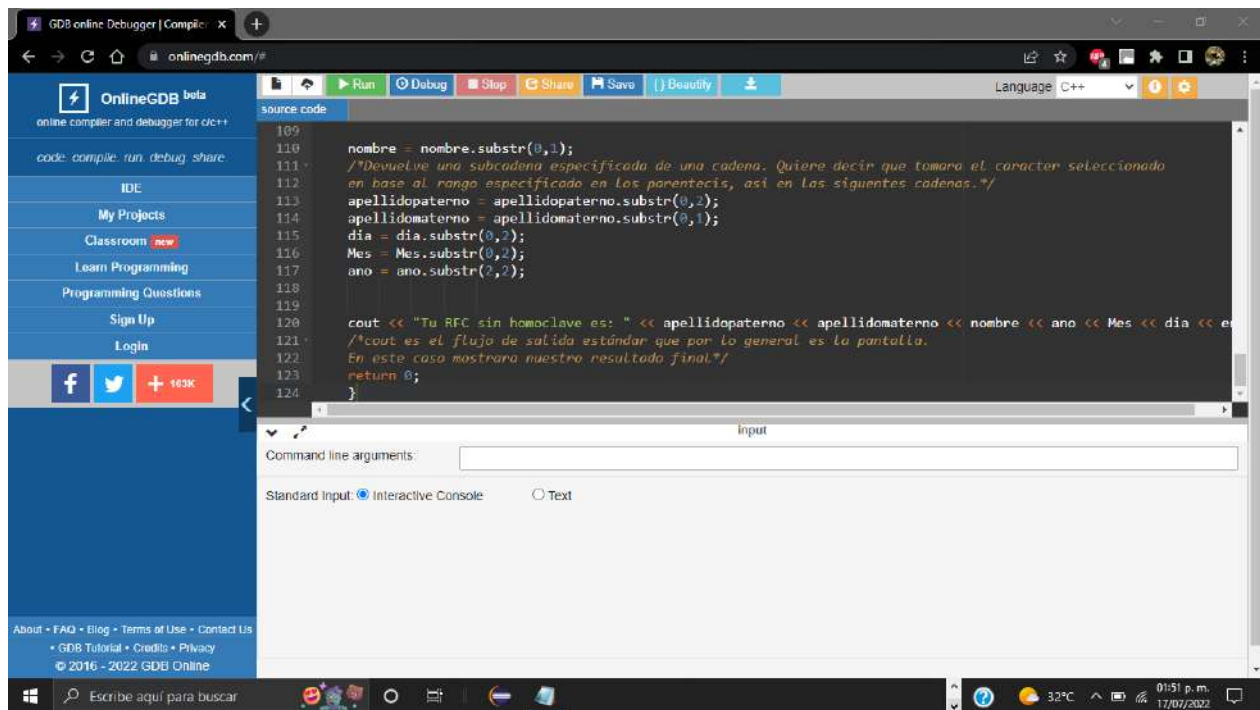
At the bottom, there is an "input" field and a "Command line arguments:" field.



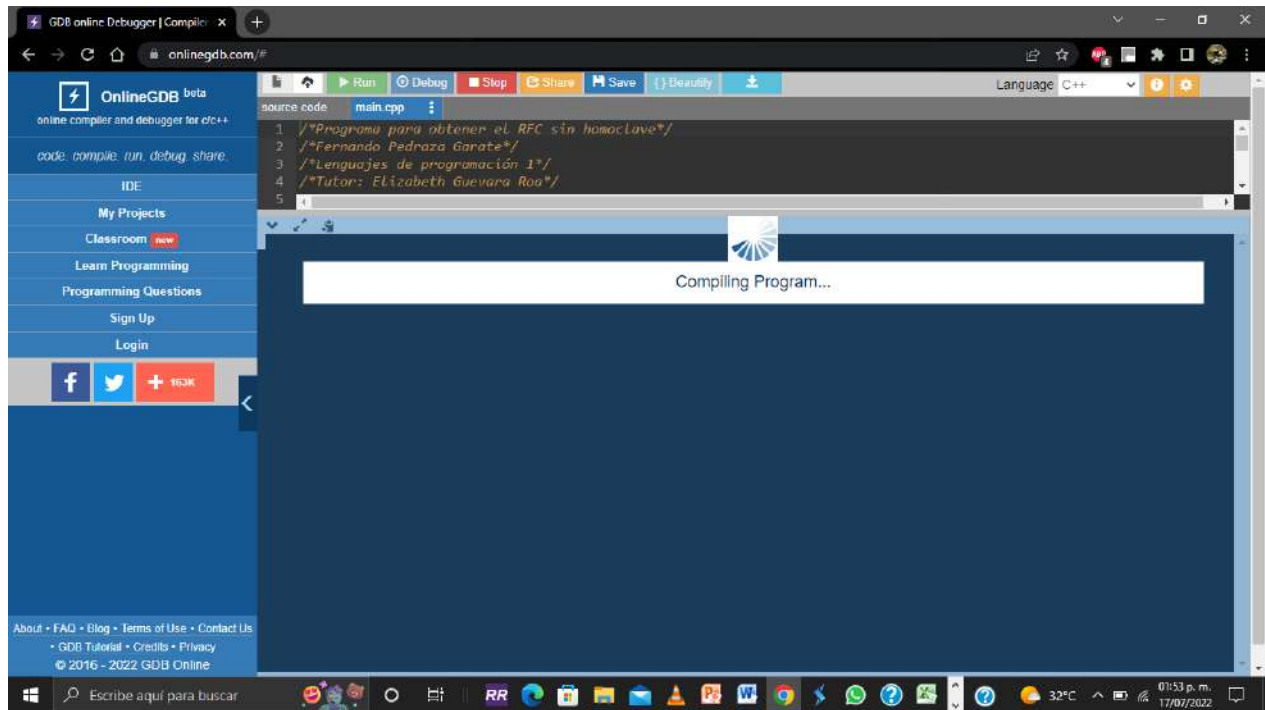
The screenshot shows the OnlineGDB website interface with the same sidebar as the first image. The main area displays C++ source code in a dark-themed editor. The code includes comments in Spanish and uses the `toupper` function to convert a string to uppercase. The code is as follows:

```
85-         string ano;
86-         /*dato que tendra la cadena año*/
87-         cin >> ano;
88-         /*cin es el flujo de entrada estándar que normalmente es el teclado*/
89-         for (int i=0;i<nombre.length(); i++){
90-             nombre[i] = toupper(nombre[i]);
91-             /*For se usa cuando queremos repetir un conjunto de instrucciones un número finito de veces.
92-             Y la función toupper() toma un carácter y devuelve su versión en mayúscula (si es que existe).
93-             En este caso cambiara las letras minúsculas en mayúsculas para la cadena nombre*/
94-         }
95-
96-         for (int i=0;i<apellidopaterno.length(); i++){
97-             apellidopaterno[i] = toupper(apellidopaterno[i]);
98-             /*For se usa cuando queremos repetir un conjunto de instrucciones un número finito de veces.
99-             Y la función toupper() toma un carácter y devuelve su versión en mayúscula (si es que existe).
100-            En este caso cambiara las letras minúsculas en mayúsculas para la cadena apellidopaterno*/
101-         }
102-
103-         for (int i=0;i<apellidomaterno.length(); i++){
104-             apellidomaterno[i] = toupper(apellidomaterno[i]);
105-             /*For se usa cuando queremos repetir un conjunto de instrucciones un número finito de veces.
106-            Y la función toupper() toma un carácter y devuelve su versión en mayúscula (si es que existe).
107-            En este caso cambiara las letras minúsculas en mayúsculas para la cadena apellidomaterno*/
108-         }
109-
```

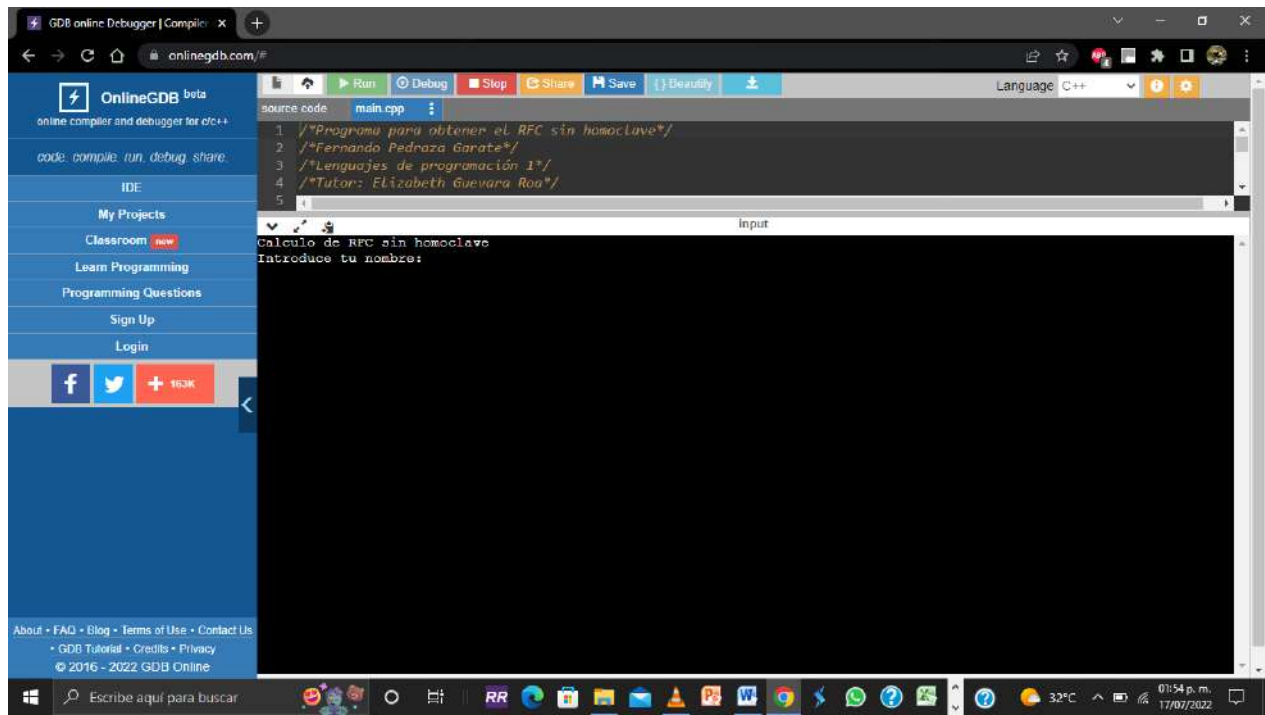
At the bottom, there is an "input" field, a "Command line arguments:" field, and radio buttons for "Standard Input" (selected), "Interactive Console", and "Text".



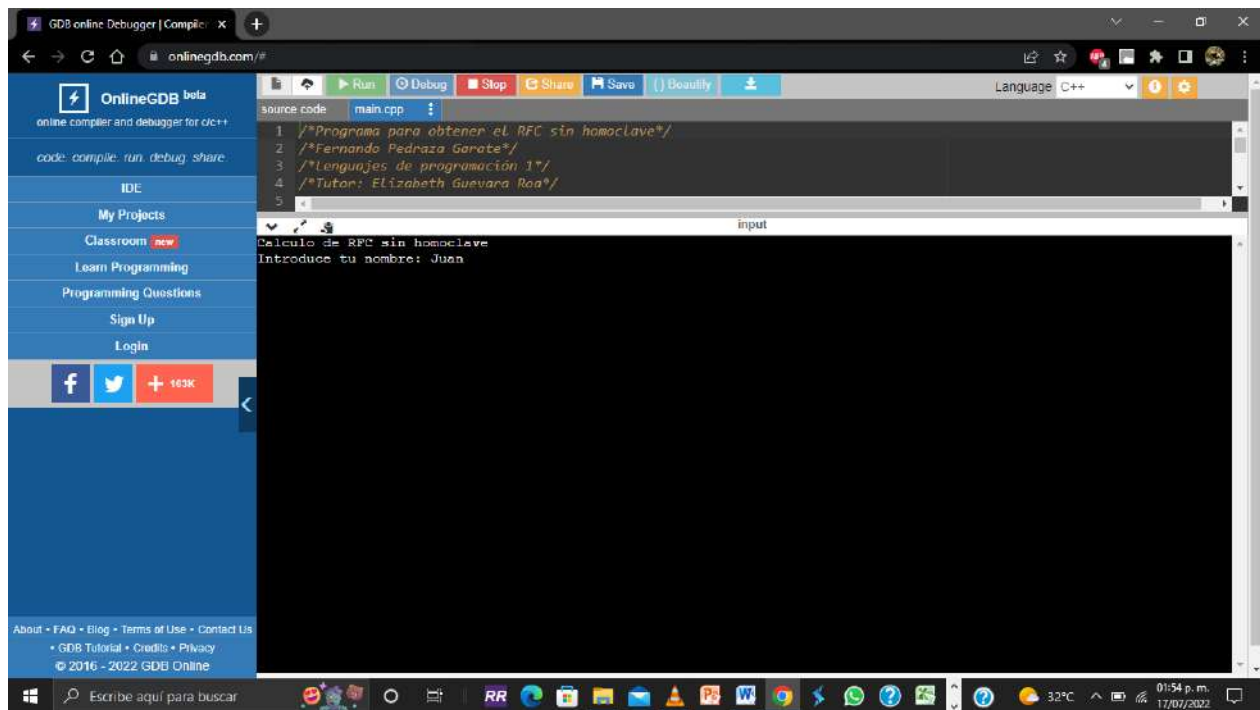
Prueba del sistema.



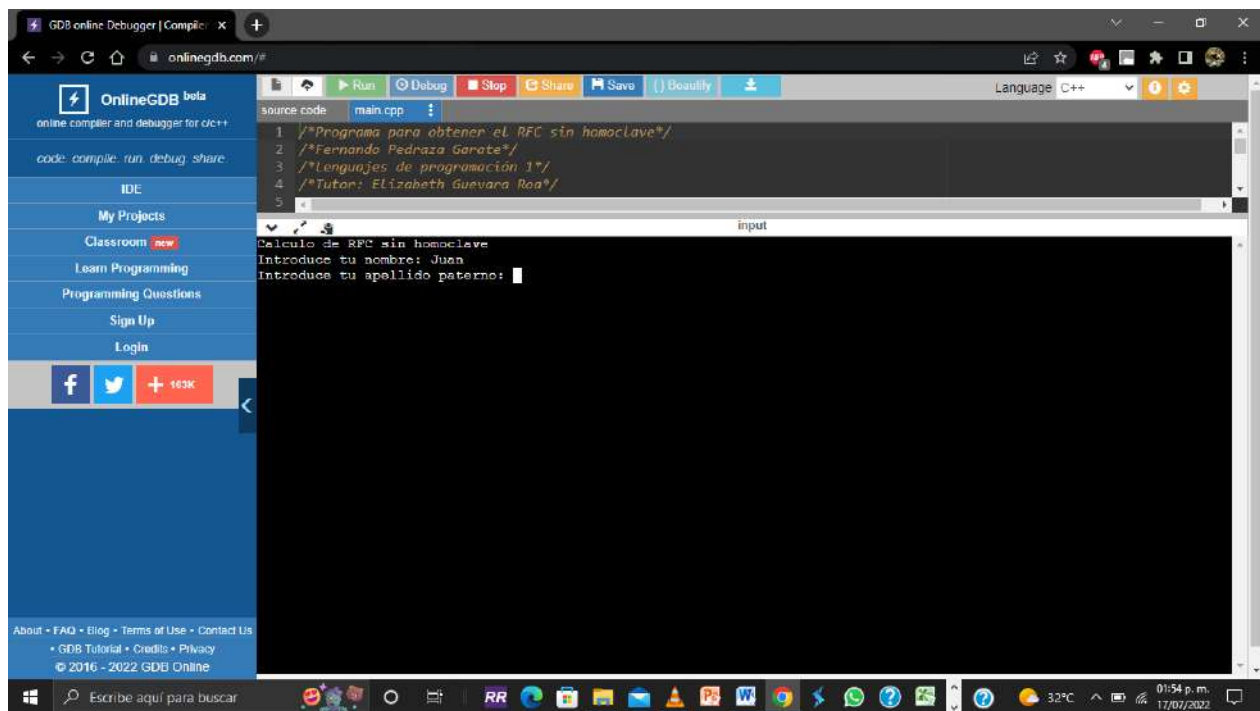
1.- Inicio del programa.



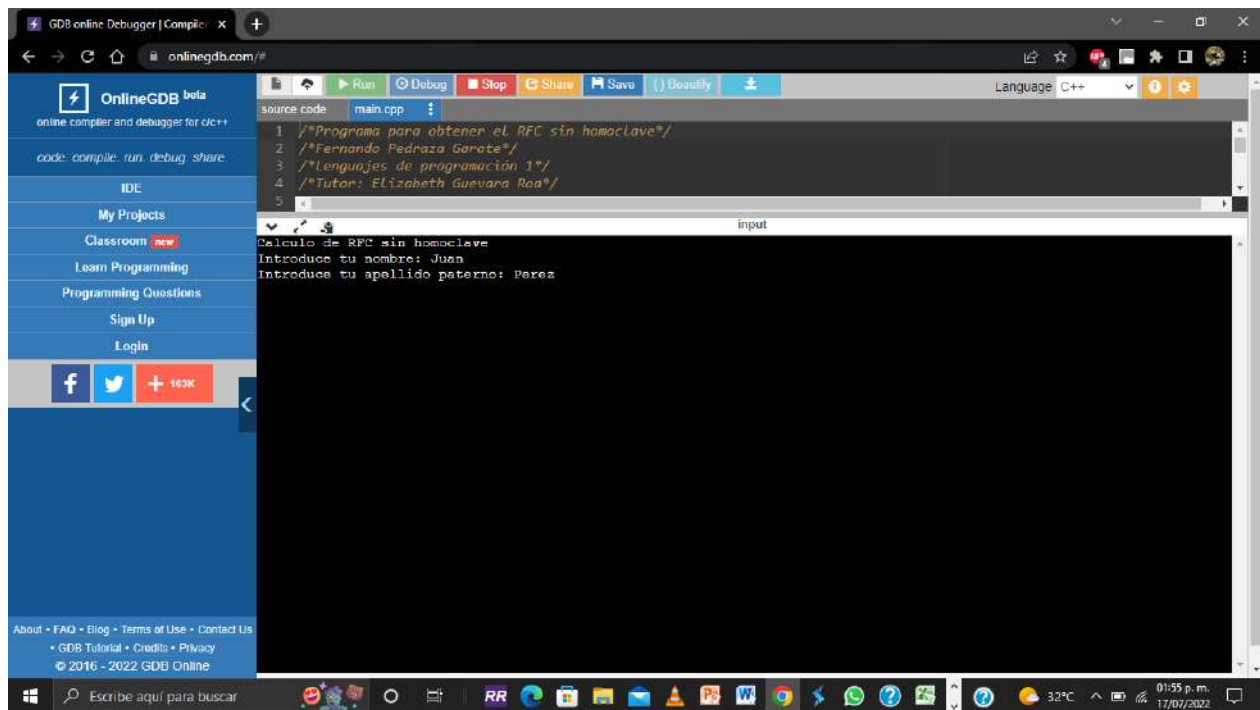
2.- Nos solicita ingresar el nombre en pantalla.



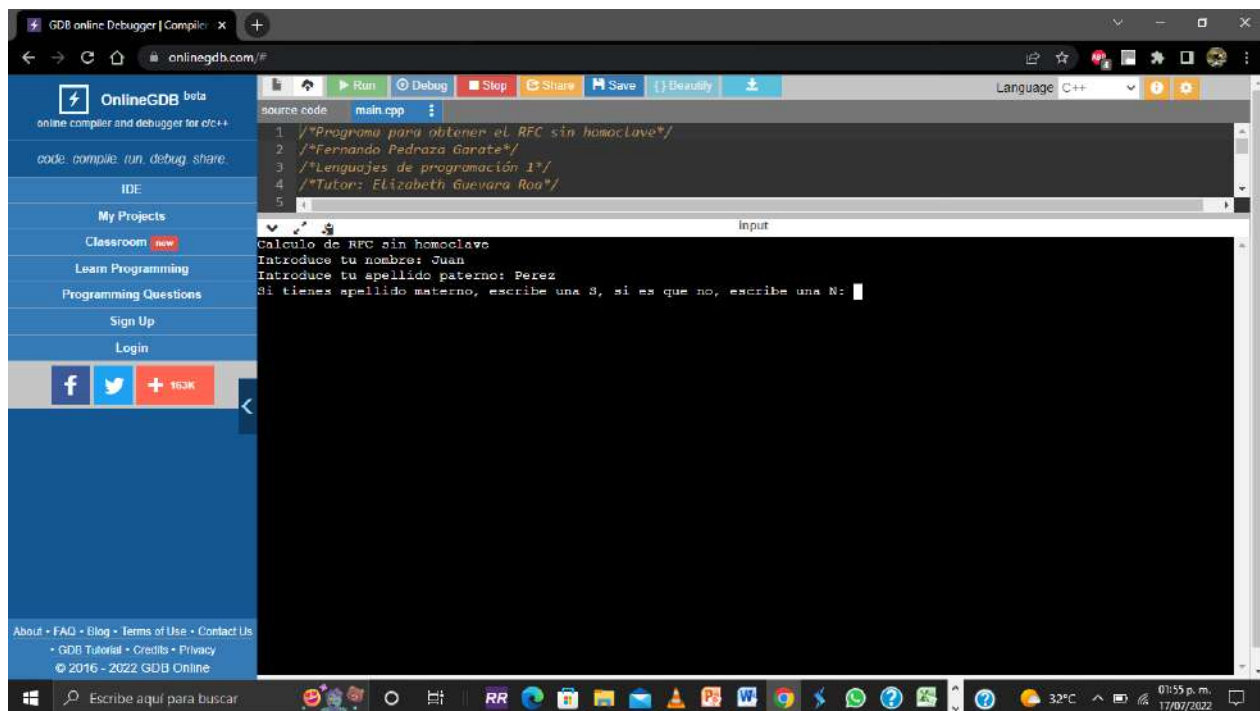
3.- Se escribe el nombre.



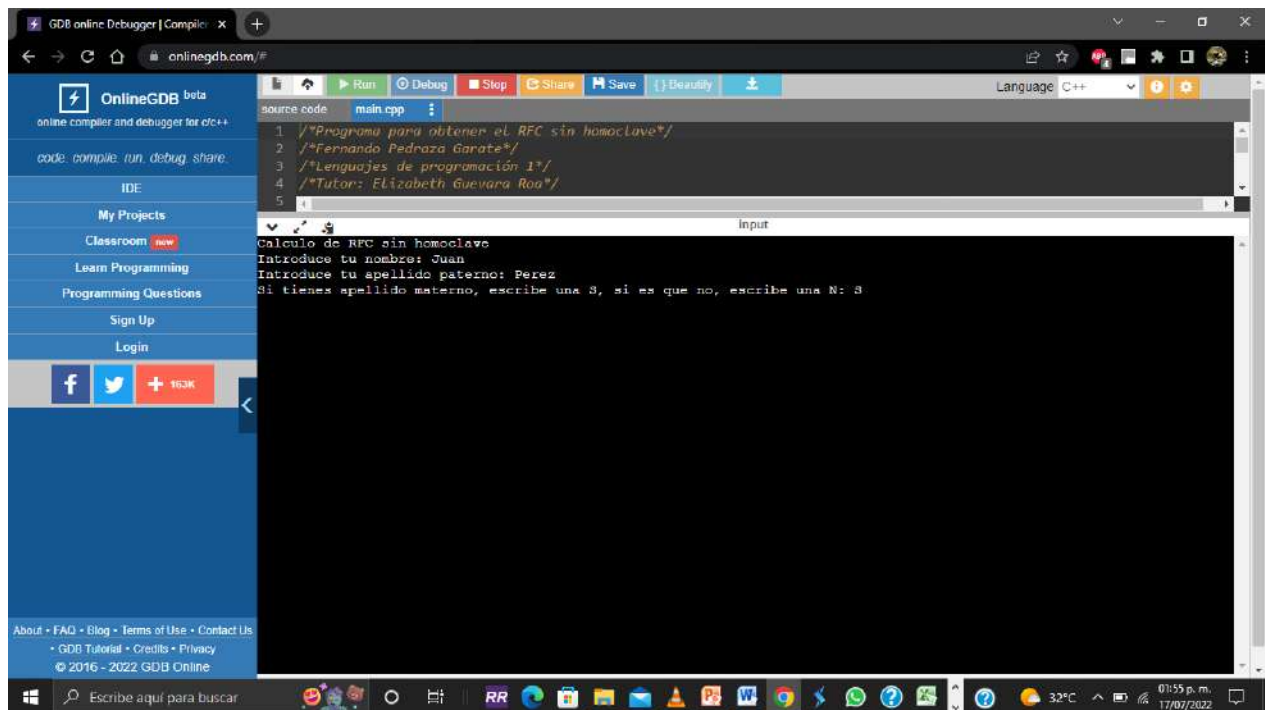
4.- Nos solicita ingresar el apellido paterno.



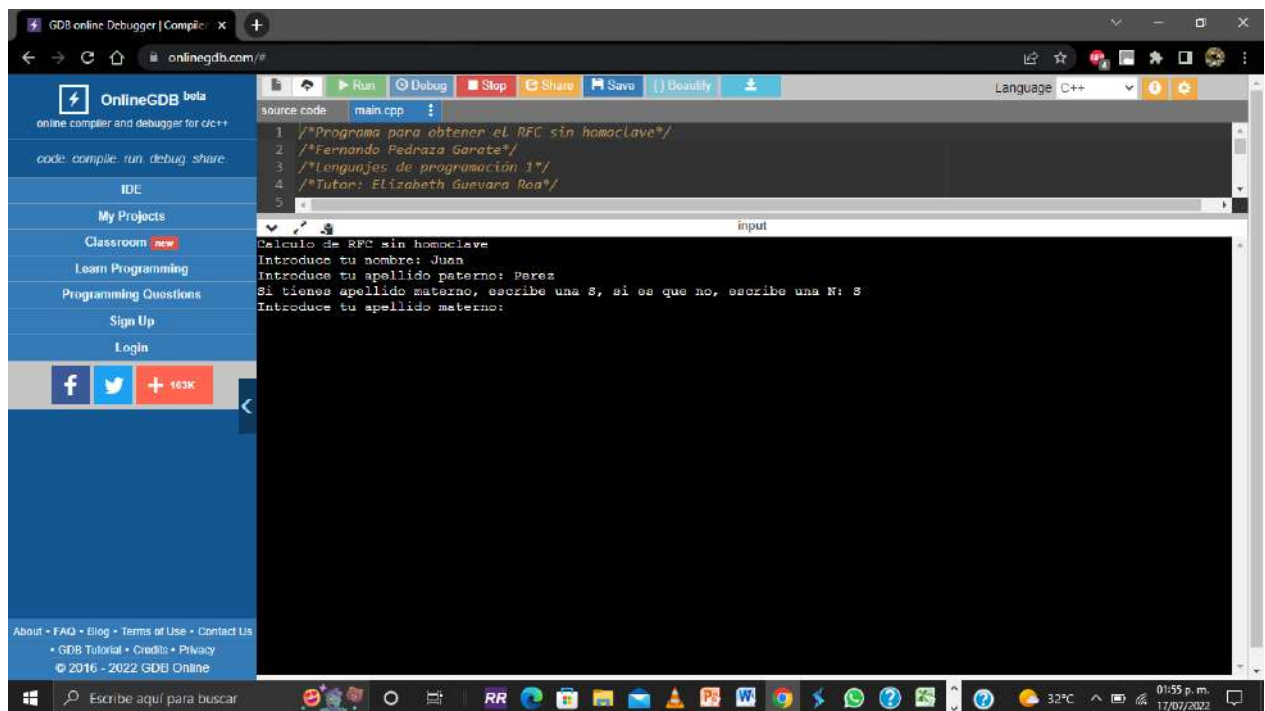
5.- Se escribe el apellido paterno.



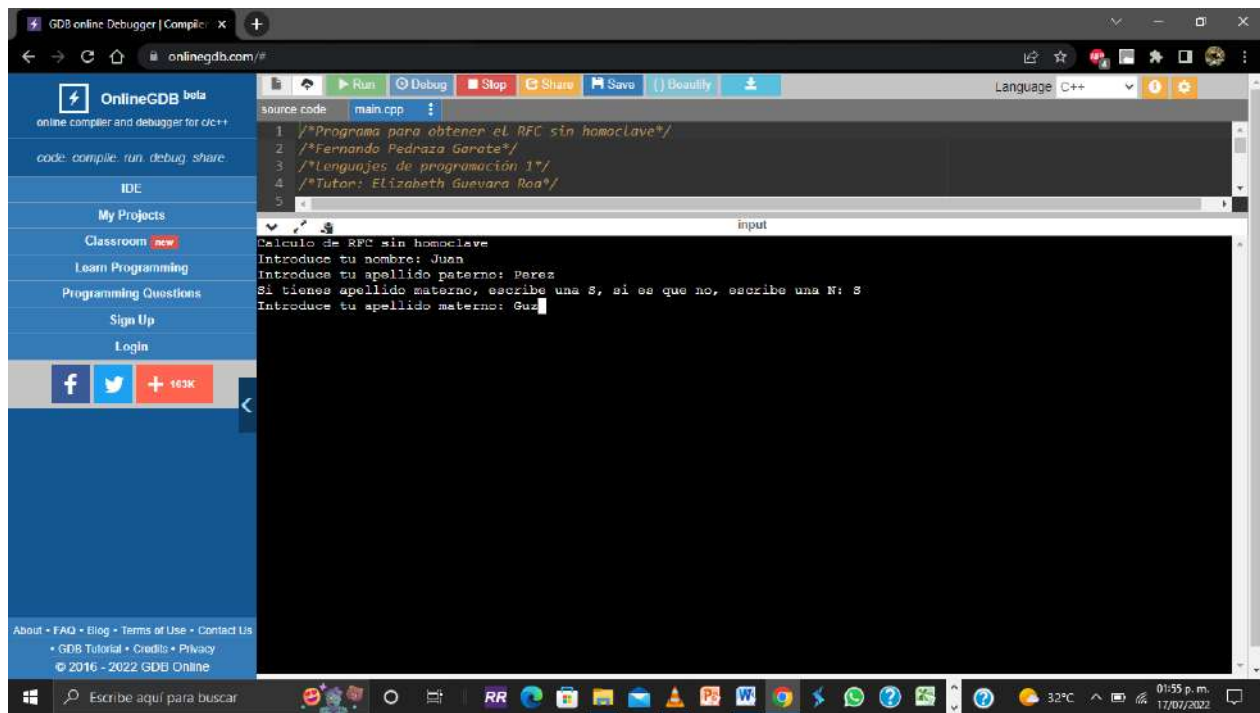
6.- Indica que hacer si se tiene apellido materno o no.



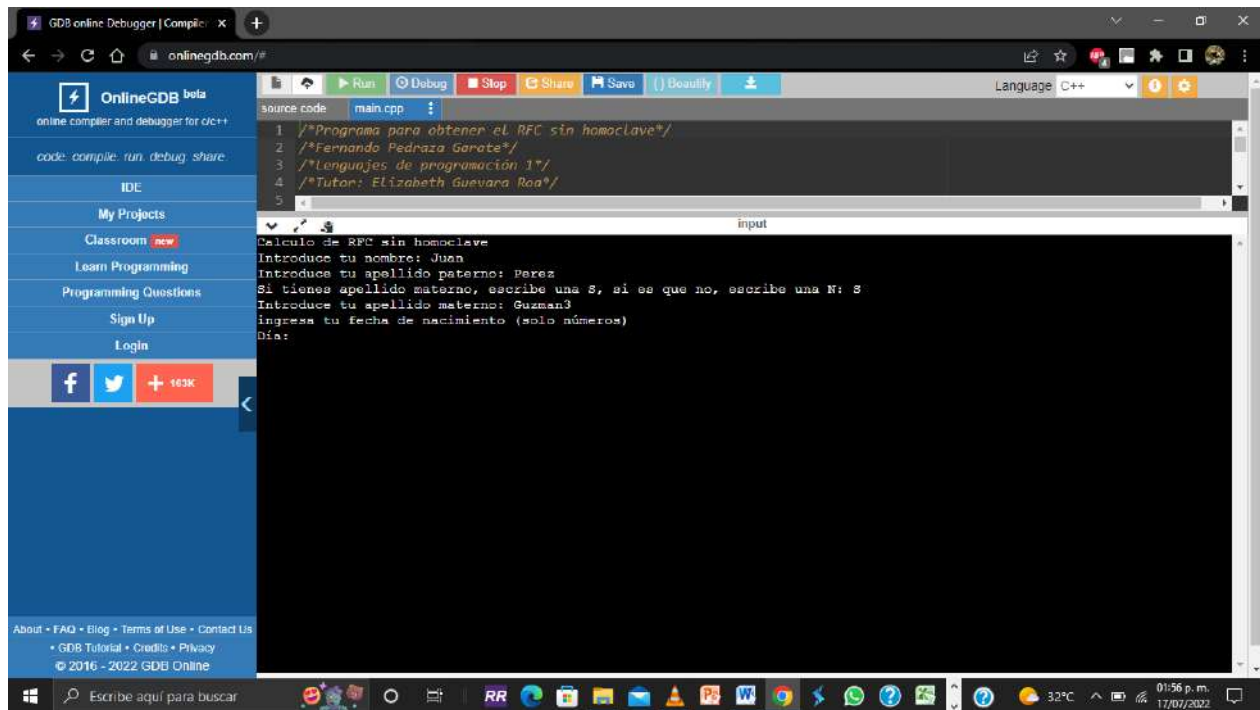
7.- En este caso si se tiene y escribimos una “S”.



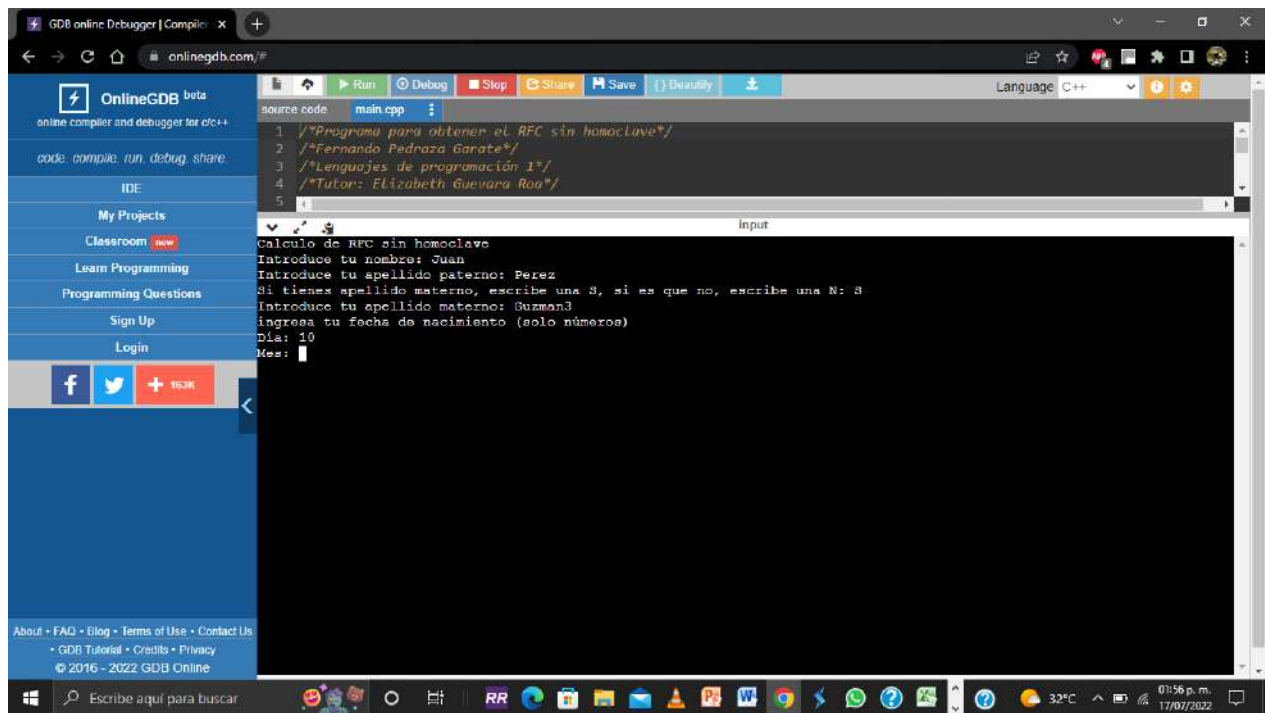
8.- Nos solicita ingresar el apellido materno.



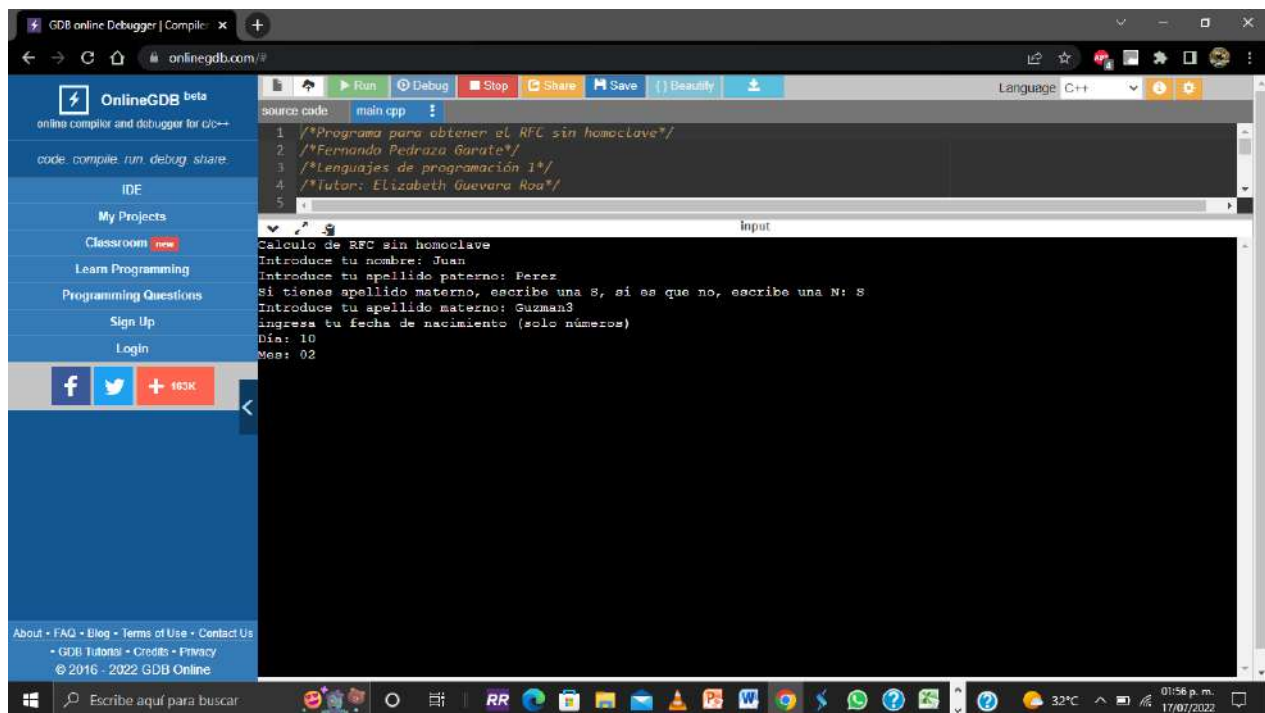
9.- Se escribe el apellido materno.



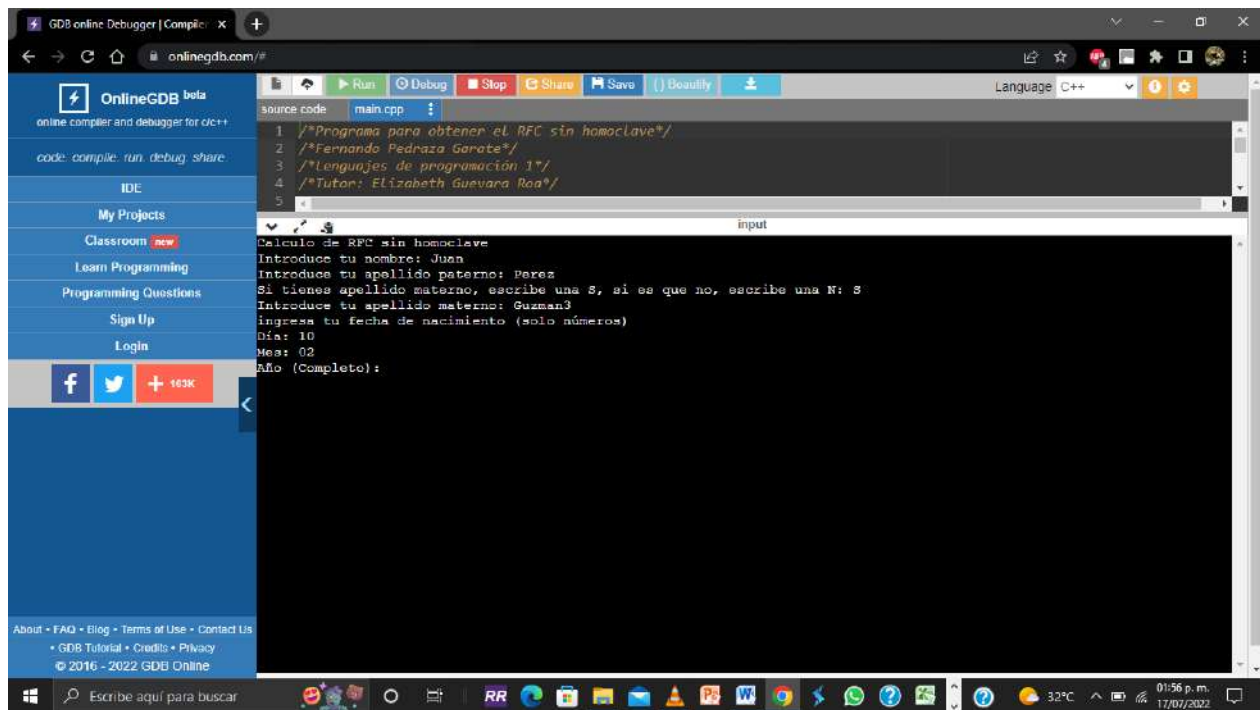
10.- Nos solicita ingresar la fecha del día de nacimiento.



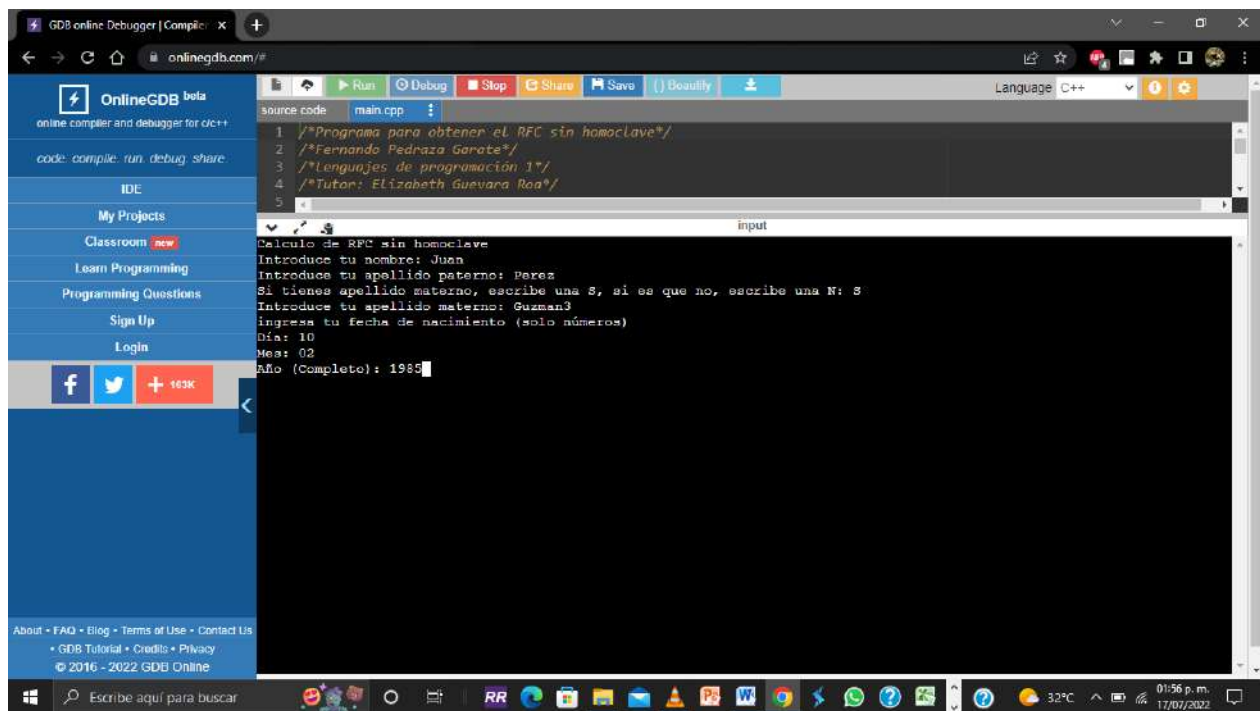
11.- Nos solicita ingresar el mes de nacimiento.



12.- Se ingresa el mes de nacimiento.



13.- Nos solicita el año de nacimiento.



14.- Se ingresa el año de nacimiento.

The screenshot displays the OnlineGDB web interface. On the left is a sidebar with navigation links: OnlineGDB beta, code, compile, run, debug, share, IDE, My Projects, Classroom, Learn Programming, Programming Questions, Sign Up, and Login. The main area shows a C++ source code file named main.cpp with the following content:

```
1 /*Programa para obtener el RFC sin homoclave*/
2 /*Fernando Pedraza Garate*/
3 /*Lenguajes de programación 1*/
4 /*Tutor: Elizabeth Guevara Roa*/
5
```

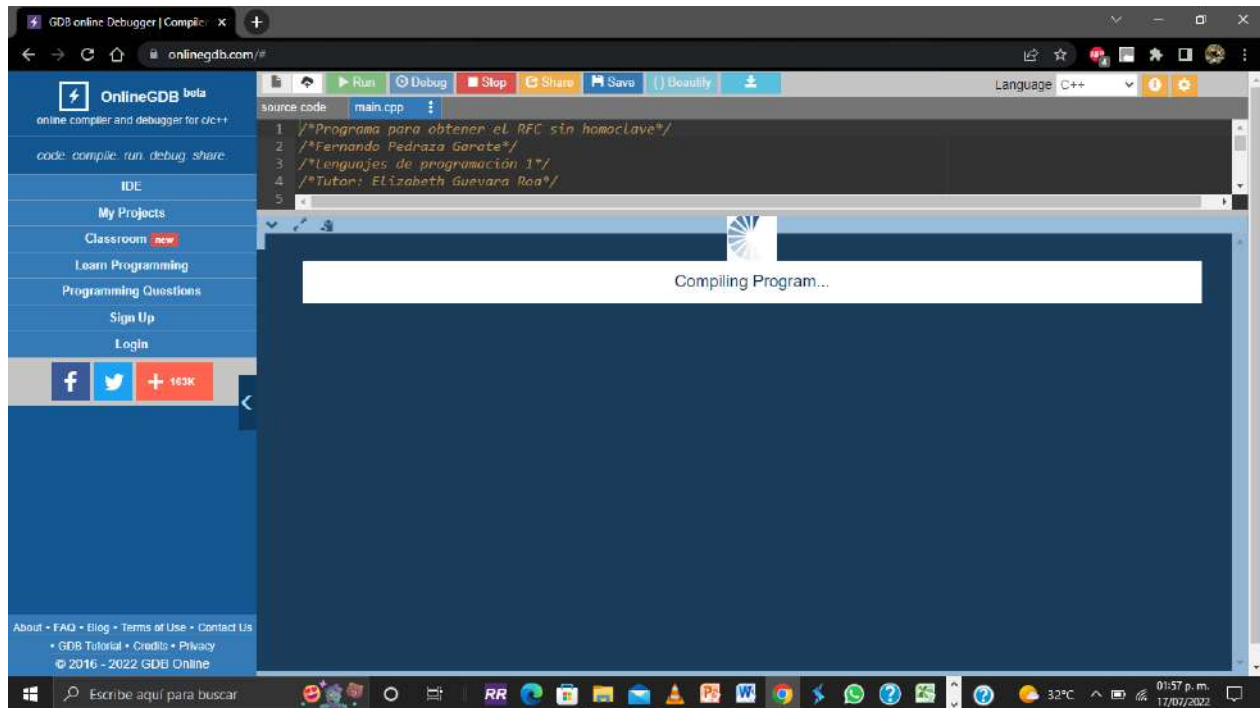
Below the code editor is an input field and a console window. The console shows the program's execution with the following input and output:

```
Calculo de RFC sin homoclave
Introduce tu nombre: Juan
Introduce tu apellido paterno: Perez
Si tienes apellido materno, escribe una S, si es que no, escribe una N: S
Introduce tu apellido materno: Guzman3
Ingresa tu fecha de nacimiento (solo números)
Dia: 10
Mes: 02
Año (Completo): 1985
Tu RFC sin homoclave es: PEGJ850210
...Program finished with exit code 0
Press ENTER to exit console.
```

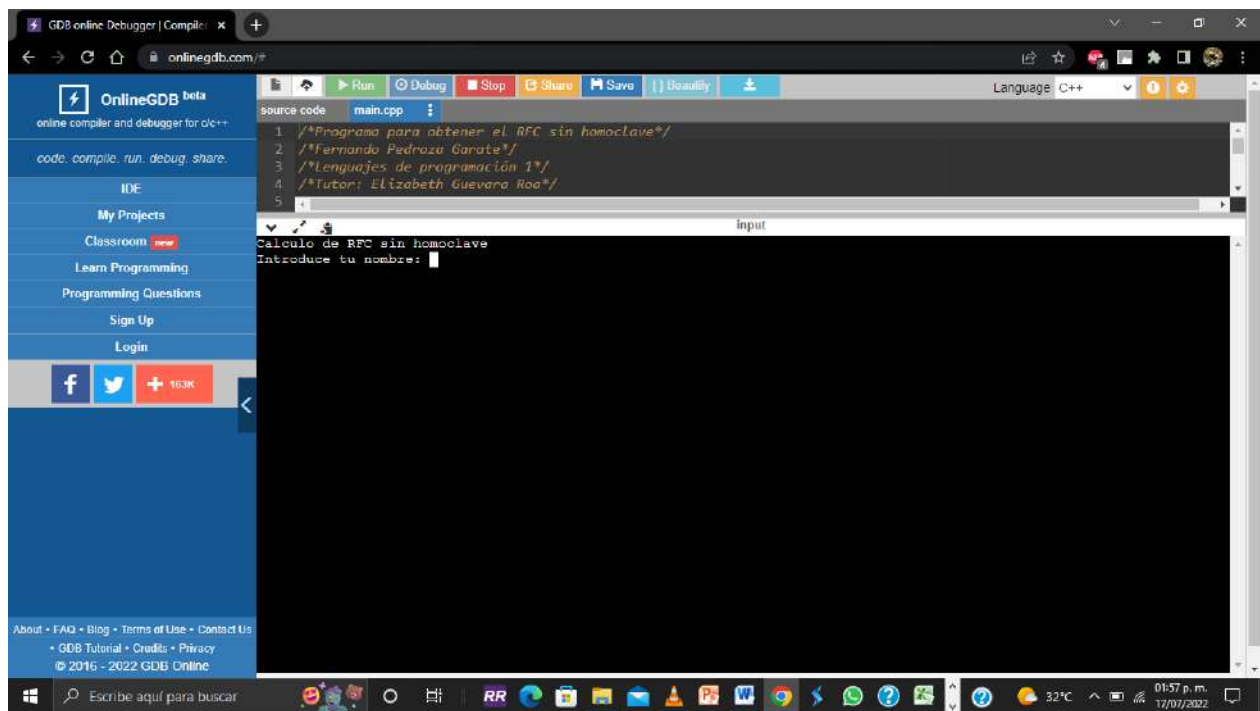
The bottom of the image shows a Windows taskbar with various application icons and system information: 32°C, 01:57 p.m., 17/07/2022.

15.- Nos muestra el resultado en base a los datos otorgados.

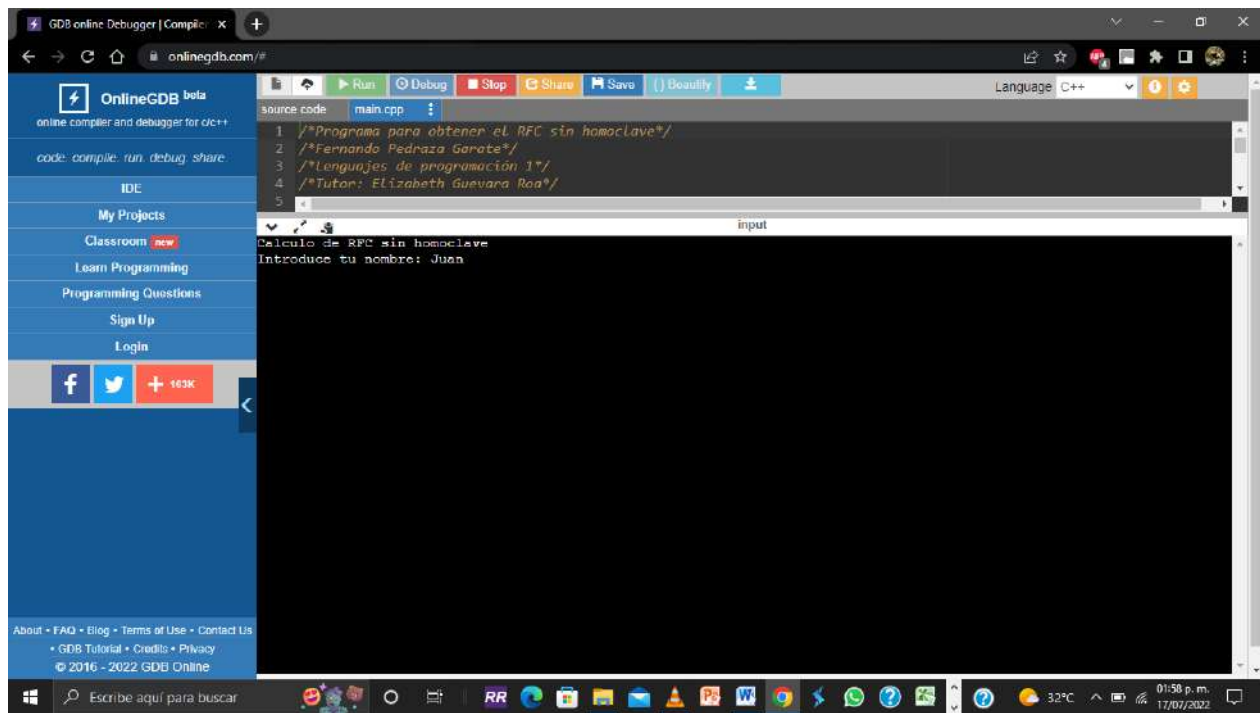
Ejecución del programa en caso de no contar con apellido materno.



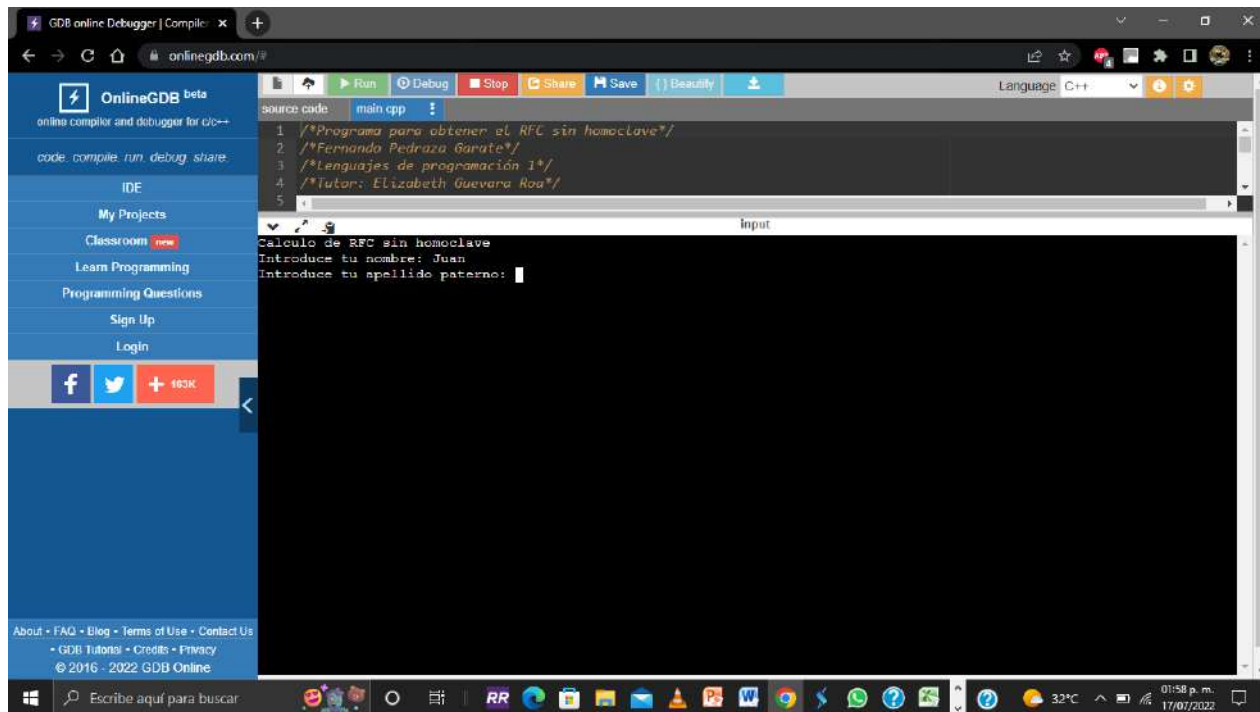
1.- Inicio del programa.



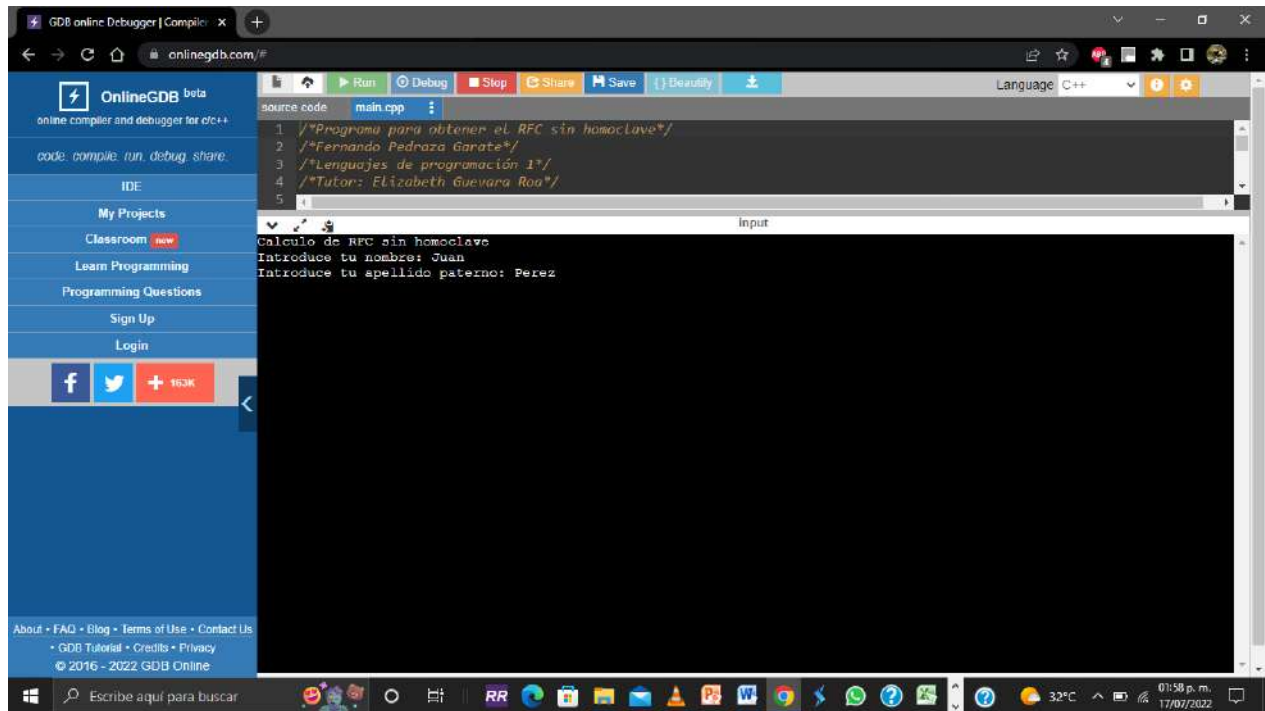
2.- Nos solicita ingresar el nombre en pantalla.



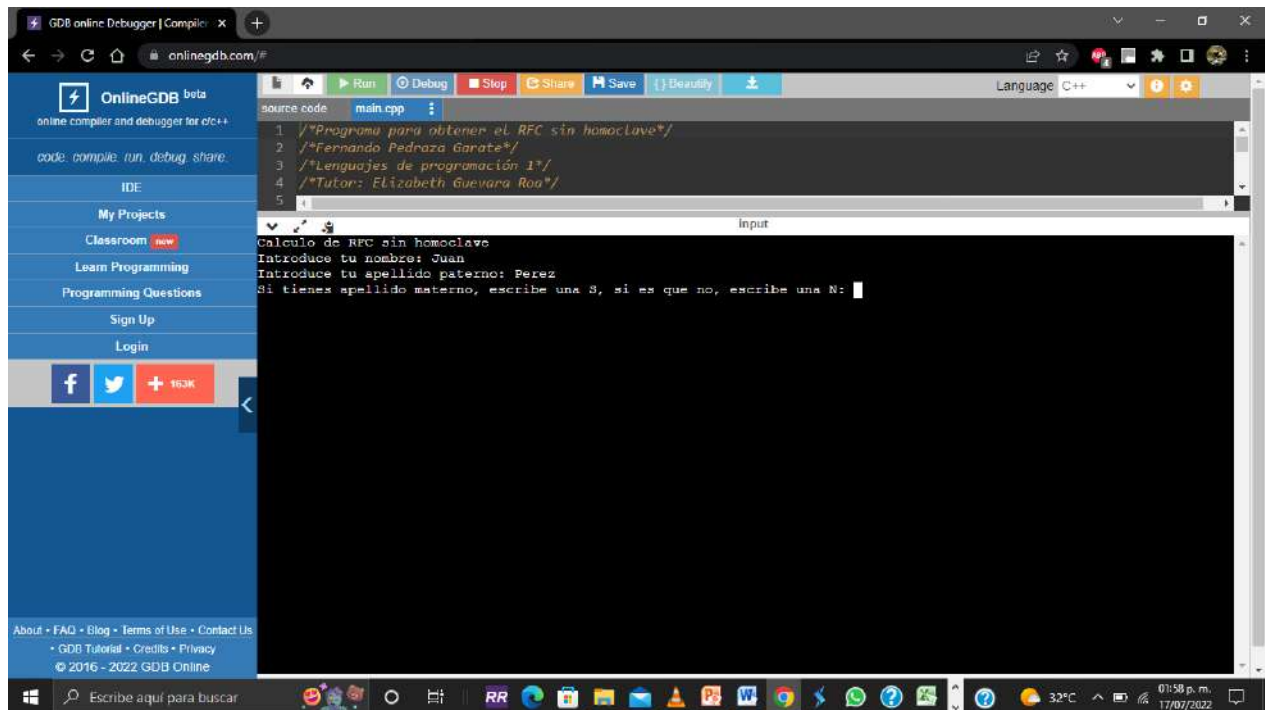
3.- Se escribe el nombre.



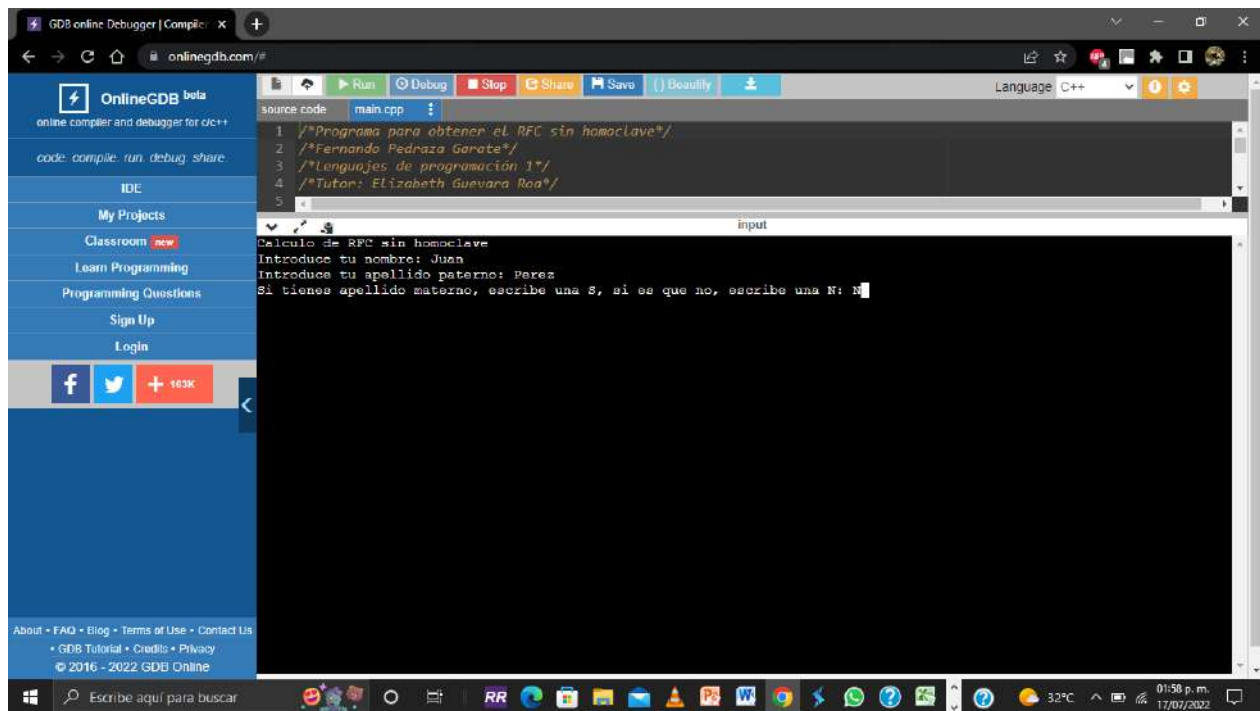
4.- Nos solicita ingresar el apellido paterno.



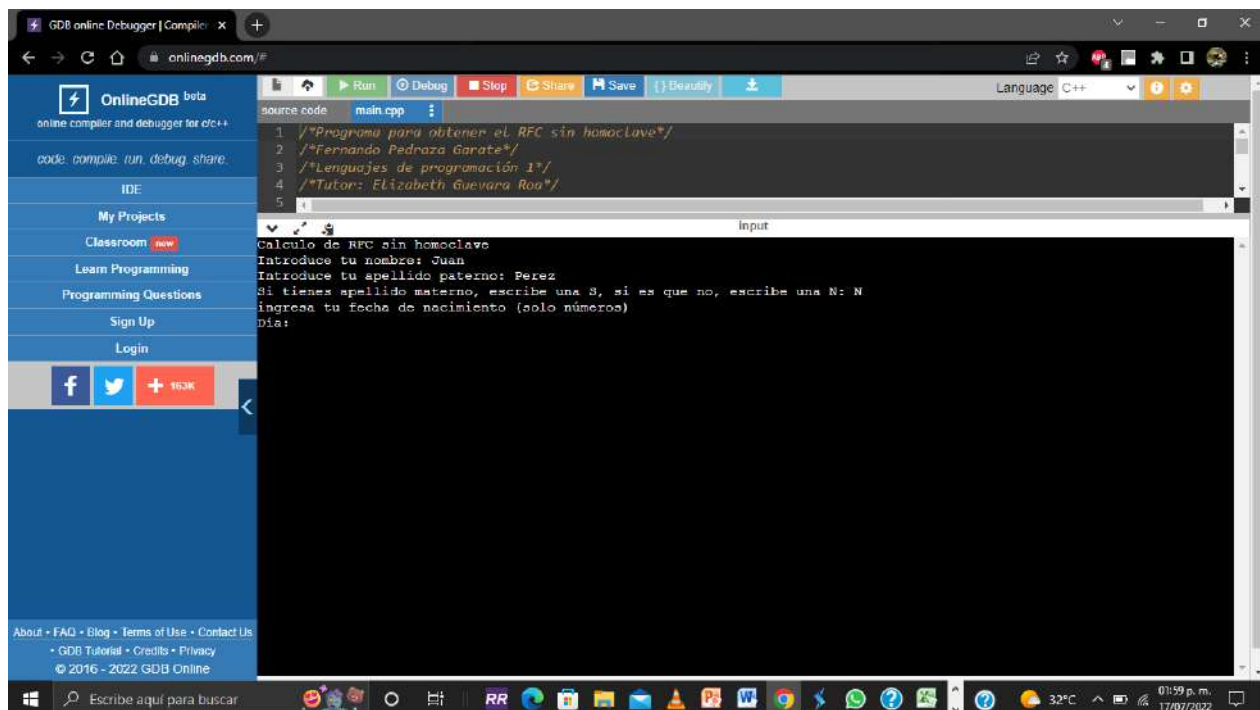
5.- Se escribe el apellido paterno.



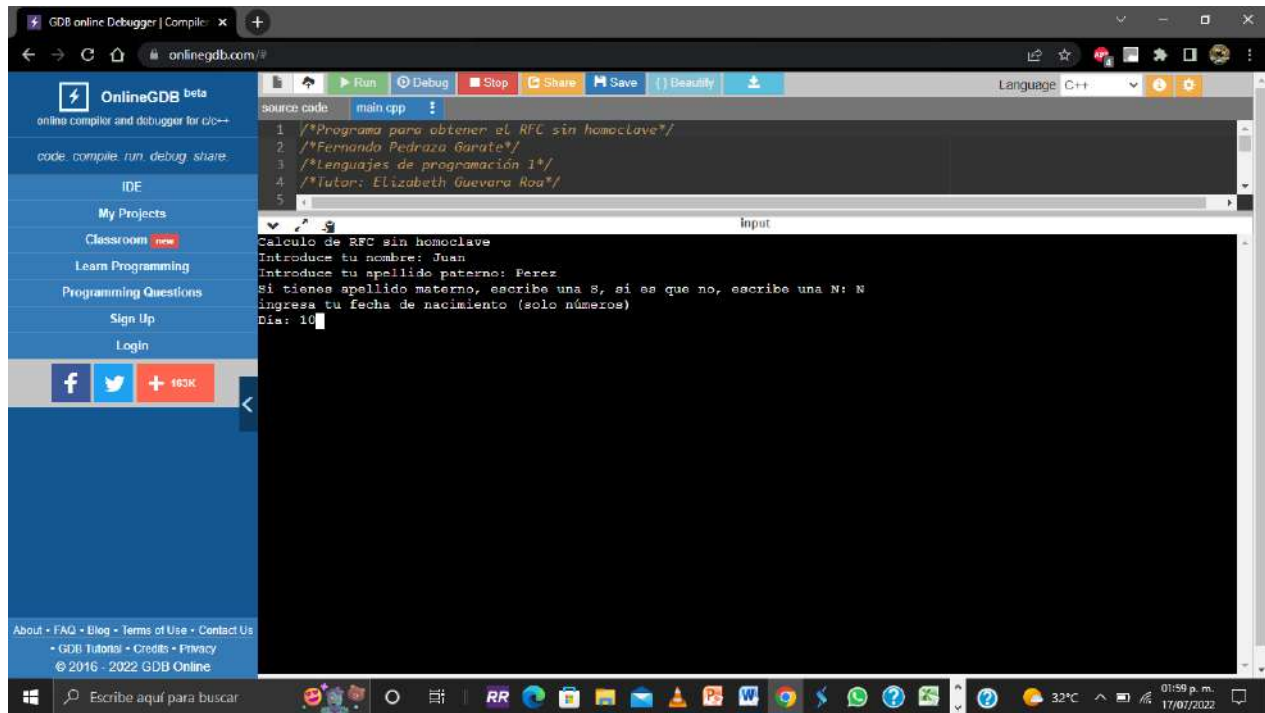
6.- Indica que hacer si se tiene apellido materno o no.



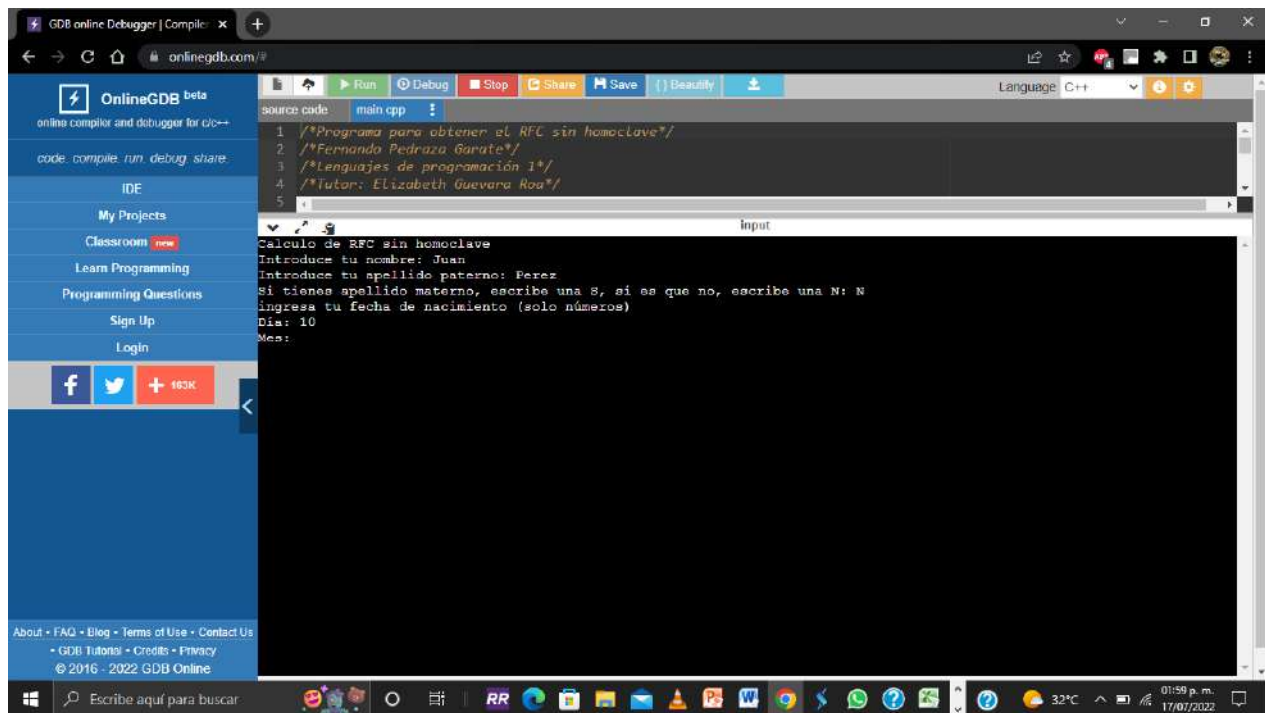
7.- En este caso no se tiene y escribimos una “N”.



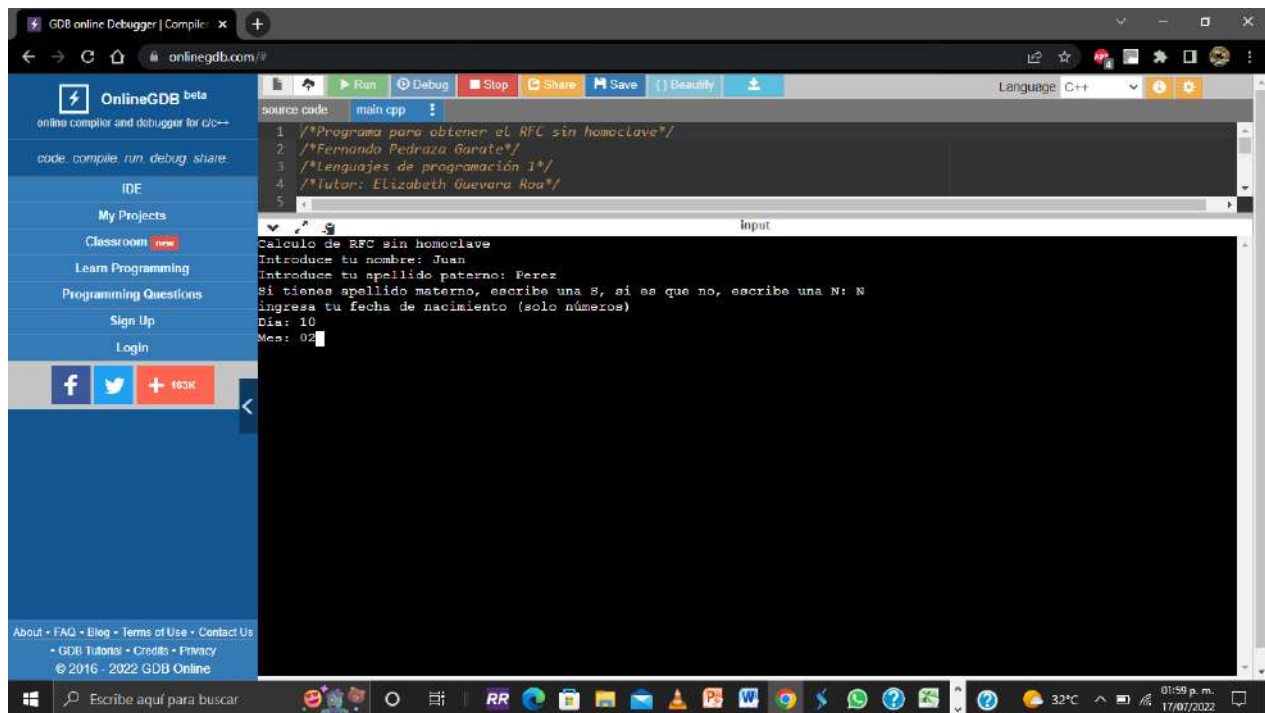
8.- Nos solicita ingresar la fecha del día de nacimiento.



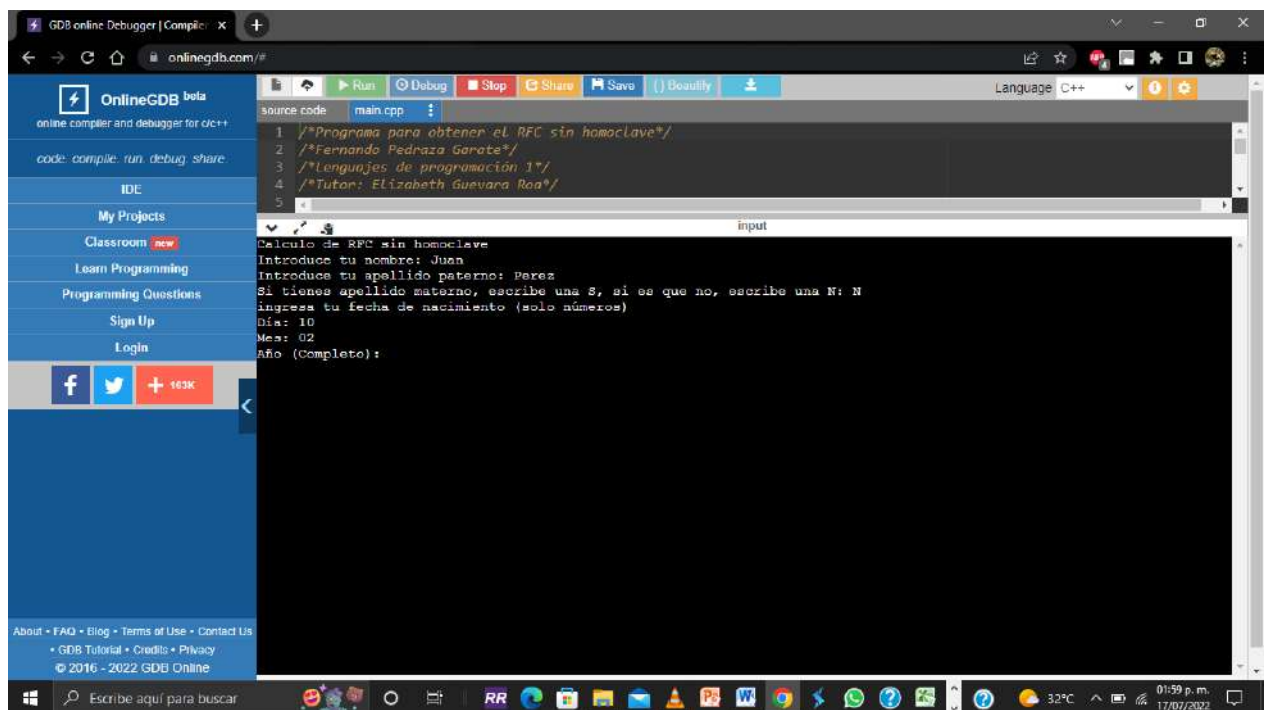
9.- Se ingresa en pantalla la fecha de nacimiento.



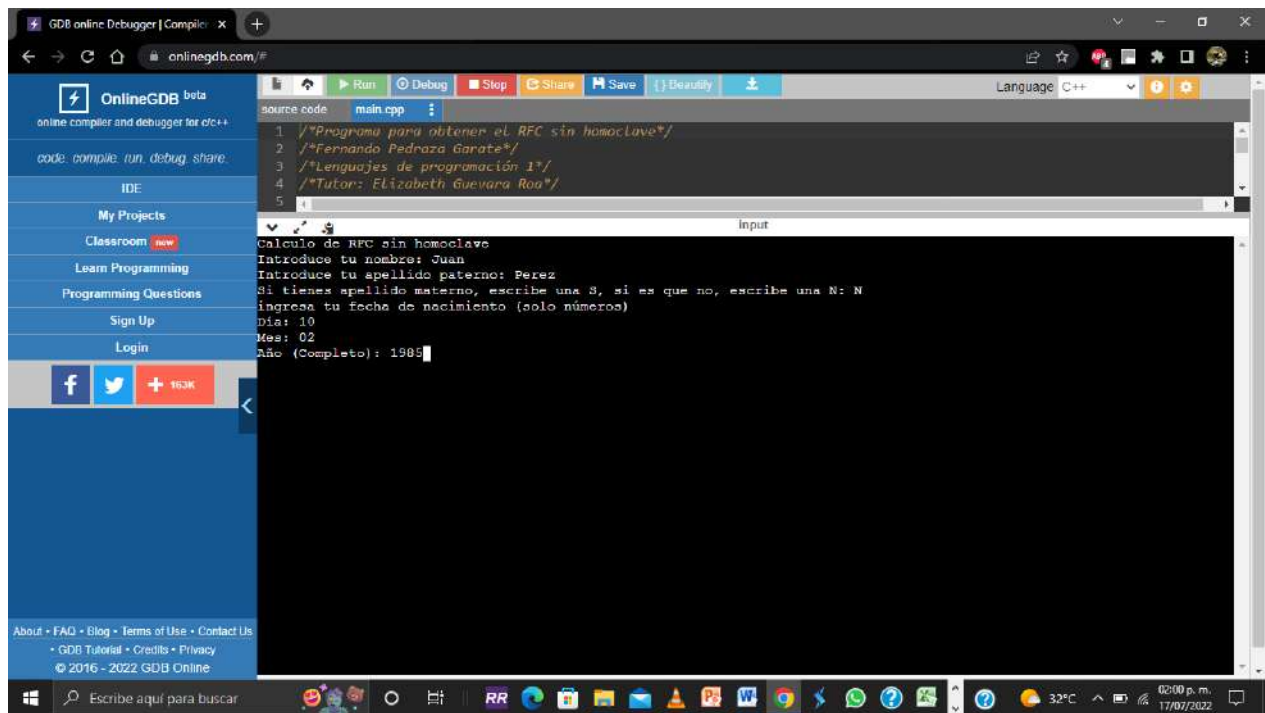
11.- Nos solicita ingresar el mes de nacimiento.



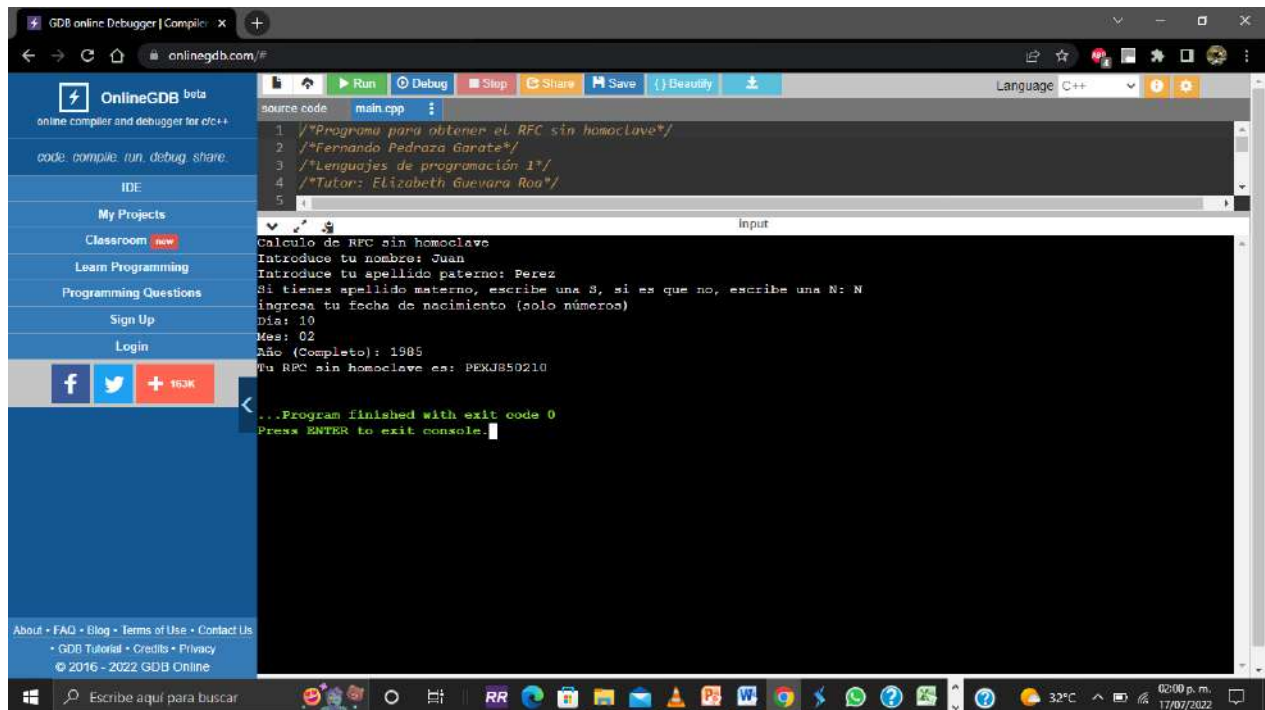
12.- Se ingresa el mes de nacimiento.



13.- Nos solicita ingresar el año de nacimiento.

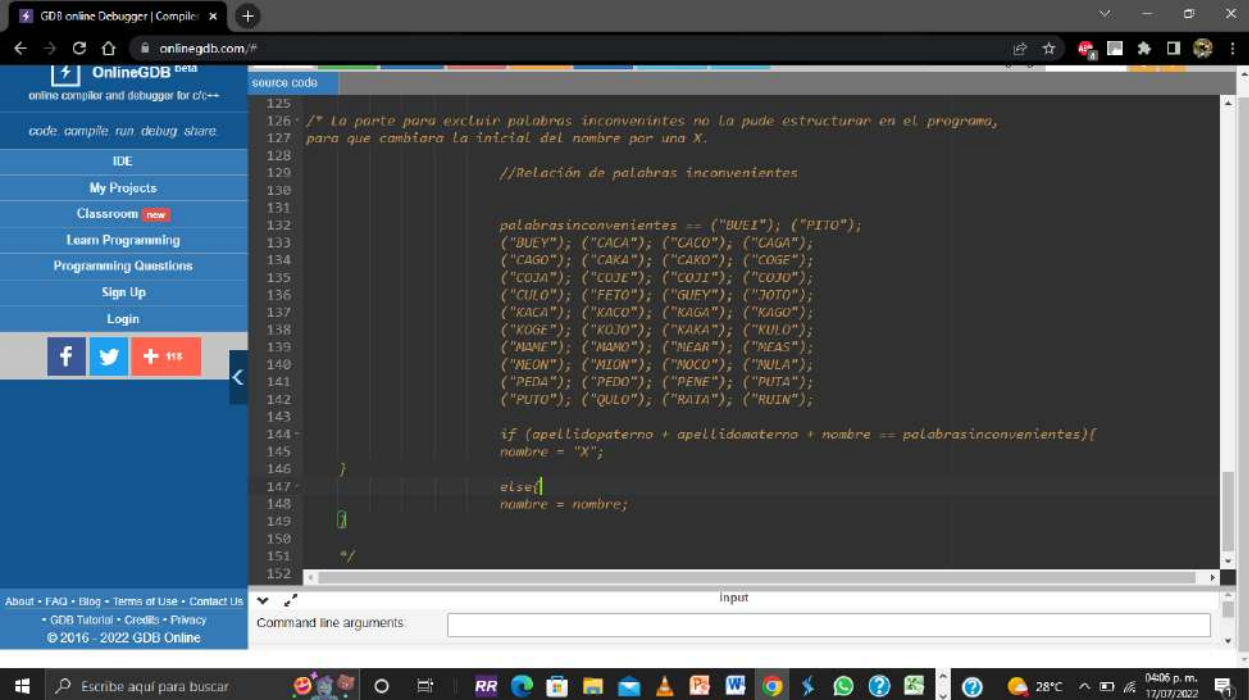


14.- Se ingresa el año de nacimiento.



15.- Nos muestra el resultado con la excepción del apellido materno, sustituyéndolo por un X.

Detalles por mejorar en la ejecución del programa:



The screenshot shows the OnlineGDB web interface. The left sidebar contains navigation links: IDE, My Projects, Classroom (highlighted), Learn Programming, Programming Questions, Sign Up, and Login. Below these are social media icons for Facebook, Twitter, and a generic share icon. The main area displays a C++ source code file with line numbers 125 to 152. The code defines an array of inappropriate words and checks if a concatenated name string contains any of them. If it does, the name is replaced with 'X'. The code is as follows:

```
125
126 /* La parte para excluir palabras inconvenientes no la pude estructurar en el programa,
127 para que cambiara la inicial del nombre por una X.
128
129 //Relación de palabras inconvenientes
130
131 palabrasinconvenientes = {"BUEY", "PITO",
132 ("BUEY", "CACA", "CACO", "CAGA",
133 "CAGO", "CAKA", "CAKO", "COGE",
134 "COJA", "COJE", "COJI", "COJO",
135 "CULO", "FETO", "GUEY", "TOTO",
136 "KACA", "KACO", "KAGA", "KAGO",
137 "KOGA", "KOJO", "KAKA", "KULO",
138 "MANE", "MAHO", "NEAR", "NEAS",
139 "MEON", "MION", "MOCO", "MULA",
140 "PEDA", "PEDO", "PENE", "PUTA",
141 "PUTO", "QULO", "RAIA", "RUIN"};
142
143 if (apellidopaterno + apellidomaterno + nombre == palabrasinconvenientes){
144     nombre = "X";
145 }
146
147 else{
148     nombre = nombre;
149 }
150
151 */
152
```

Below the code editor is an 'Input' field and a 'Command line arguments' field. The Windows taskbar at the bottom shows the time as 04:06 p.m. on 17/07/2022.

Evitar las palabras inconvenientes en el resultado final.

Conclusión.

En conclusión es importante comprender la base del lenguaje de C++ para poder desarrollar y elaborar un buen programa, reconociendo la diferencia entre las clases y los objetos, así como su estructura, para no cometer tener errores al momento de ejecutar los programas, respetando en todo momento las llaves, corchetes, y el punto y coma, ya que son componentes de esta misma.

En cuestión de utilidad es muy diverso al tener oportunidad de aplicarse en diferentes tipos de desarrollo, como ya se mencionó, páginas WEB, video juegos, BD, nubes, etc.

En base a la actividad 2 podemos reafirmar que si no hay una buena estructura de programación se cometerán errores que nos impedirán ejecutarlos en base a las necesidades o requerimientos solicitados. Ocasionando invertir más tiempo de lo planeado.

La utilidad que se le puede dar a las variables es muy extensa por lo que bien aplicadas permitirá obtener buenos resultados.

Referencias.

<https://openwebinars.net/blog/que-es-cpp/>

<https://docs.microsoft.com/es-es/cpp/standard-library/function-objects-in-the-stl?view=msvc-170>

<https://www.codingame.com/playgrounds/50557/clases-y-objetos-en-c-practica-1/clases-y-objetos-en->

<c#:~:text=Una%20clase%20puede%20pensarse%20como,que%20permiten%20manipular%20tales%20datos.>

<https://www.codingame.com/playgrounds/50557/clases-y-objetos-en-c-practica-1/clases-y-objetos-en->

[c#:~:text=Para%20declarar%20una%20clase%20en,por%20el%20usuario%20\(programador\).](c#:~:text=Para%20declarar%20una%20clase%20en,por%20el%20usuario%20(programador).)

<https://cec.cele.unam.mx/include/howToRFC.php>

<https://www.delftstack.com/es/howto/cpp/how-to-convert-string-to-uppercase-cpp/>

<https://www.aprendeaprogramar.com/cursos/verApartado.php?id=16201>

<https://parzibyte.me/blog/2021/04/24/cortar-cadena-cpp/>

<https://www.youtube.com/watch?v=bjqF9qvNhsE>

<https://www.youtube.com/watch?v=D7hbaJhgZBs>

<https://www.youtube.com/watch?v=RvMBDWuXsaw>

<https://cplusplus.com/reference/string/string/replace/>