

Tarea #3

Inyerman Alexander Xap Chin
20210097

Escuela de Mecánica Eléctrica,
Facultad de Ingeniería, Universidad
de San Carlos de Guatemala.

A. Resumen

Primero creamos un nuevo archivo tipo c en nuestro Visual Studio Code y lo bautizamos como true procedemos a crear nuestro código luego lo compilamos y corremos para observar que todo funcione como esperamos luego de ver que todo esté bien procedemos a guardarlo en la carpeta que tenemos vinculada al repositorio.

Hacemos lo mismo con el código variables y calculadora.

Luego utilizamos el comando

```
$ get add *
```

Para agregar todos los códigos nuevos y luego con el código

```
$ git commit -m "primer commit"
```

Y

```
$git push -u origin main
```

Terminamos de subirlo correctamente al repositorio.

B. Objetivos

- Aprender a utilizar la librería stdbool.h
- Aprender como declarar una variable e imprimirla.
- Aprender a escanear un numero para poder asignarle su valor a una variable.
- Crear una calculadora básica con lo aprendido.

C. Marco Teórico

Stdbool.h:

Se utiliza para poder utilizar en C las variables de tipo booleanas, ósea poder utilizar true o false en el código.

Con esta librería lo que se consigue es la eliminación de la necesidad de 0 y 1 para poder controlar el código. Este archivo de cabecera es muy simple y solo contiene la definición formal del tipo:

Macros:

Bool--- Definición del tipo.

True--- Sustituye a 1.

False--- Sustituye a 0.

Tipos de Datos en c:

C ofrece tres tipos de datos básicos:

- Números enteros definidos con la palabra clave int
- Letras o caracteres definidos con la palabra clave char
- Números reales o en coma flotante definidos con las palabras claves float o double

Enteros:

Se definen con "int" y admiten de forma opcional dos prefijos modificadores:

- "short" y "long": Modifica el tamaño en bits del entero. Existen por tanto tres tipos de enteros: "int", "short int" (que se puede abreviar como "short"), y "long int" (que se puede abreviar como "long").
- El lenguaje C no define tamaños fijos para sus tipos de datos básicos. Lo único que garantiza es que un short int tiene un tamaño **menor o igual** que un int y este a su vez un tamaño menor o igual a un long int. Esta característica del lenguaje ha complicado la creación de programas que sean compatibles entre varias plataformas.
- "unsigned": define un número natural (mayor o igual a cero).

Letras y cadenas

Las variables de tipo letra se declaran como "char". Para referirse a una letra se rodea de comillas simples: 'M'. Como las letras se representan internamente como números, el lenguaje C permite realizar operaciones aritméticas como 'M' + 25.

Las cadenas de texto o strings son simplemente tablas de "char". Las funciones de biblioteca para manipular estas cadenas asumen

que el último byte tiene valor cero. Las cadenas de texto se escriben en el programa rodeadas de dobles comillas y contienen el valor cero al final.

Números reales

Los números reales se definen con "float" o "double". La diferencia entre ambas es la precisión que ofrece su representación interna. Hay un número infinito de reales, pero se representan con un número finito de bits. A mayor número de bits, mayor número de reales se representan, y, por tanto, mayor precisión. Los reales definidos con "double" tienen un tamaño doble a los definidos con "float". Al igual que en el caso de los enteros, el tamaño de estas representaciones varía de una plataforma a otra.

Algunas plataformas ofrecen números reales con tamaño mayor al "double" que se definen como "long double". Los tamaños típicos para los tipos "float", "double" y "long double" son 4, 8 y 12 bytes respectivamente.

float a = 3.5;

double b = -5.4e-12;

long double c = 3.54e320;

D. Marco Practico

Creamos nuestros códigos y los compilamos para ver que se compilen correctamente luego los corremos y los probamos.

```
C:\WINDOWS\system32\cmd. x + v
c:\Codigos769>gcc true.c -o true
```

Luego los subimos a nuestro repositorio.

```
c:\Codigos769>git add .
c:\Codigos769>git commit -m "primer commit"
[main 5d63fe1] primer commit
10 files changed, 65 insertions(+)
create mode 100644 Tarea1_InyermanXap_202100297.pdf
create mode 100644 Tarea2_InyermanXap_202100297.pdf
create mode 100644 c
create mode 100644 calculadora.c
create mode 100644 calculadora.exe
create mode 100644 numerol.c
create mode 100644 true.c
create mode 100644 true.exe
create mode 100644 variables.c
create mode 100644 variables.exe
c:\Codigos769>git push -u origin main
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 16 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (12/12), 597.05 KiB | 12.44 MiB/s, done.
Total 12 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/InyermanXap/RepositorioCodigos.git
  9d51ffe..5d63fe1  main -> main
branch 'main' set up to track 'origin/main'.
c:\Codigos769>
```

E. Código

Primer código en C probando los comandos true y false.

```
C true.c x C variables.c C calculadora.c
C:\Codigos769> C true.c
1 #include <stdlib.h>
2 /*
3  En C, el tipo de datos bool no es un tipo integrado, es una macro definida en el archivo de
4  */
5
6
7 int main() {
8     bool x = true;
9     bool y = false;
10    system("clear");
11    printf("x: %d\n", x);
12    printf("y: %d\n", y);
13    return 0;
14 }
```

Segundo código en C probando las diferentes variables.

```
C true.c C variables.c x C calculadora.c
C:\Codigos769> C variables.c
1 #include <stdio.h>
2 #include <stdbool.h>
3 #include <stdlib.h>
4 /*
5  En C, el tipo de datos bool no es un tipo integrado, es una macro definida en el archivo de
6  */
7
8 int main() {
9     int i = 11;
10    int j = 12;
11    char c = 'A';
12    float n = 40.791512;
13    system("clear");
14    printf("i: %d, j: %d, c: %c, n: %f\n", i, j, c, n);
15    return 0;
16 }
```

Tercer código en C creando calculadora básica.

```
C true.c C variables.c C calculadora.c x
C:\Codigos769> C calculadora.c
1 #include <stdio.h>
2 int main() {
3     double num1, num2;
4
5     printf("Ingrese el primer número: no un caracter ");
6     scanf("%lf", &num1);
7
8     printf("Ingrese el segundo número: no un caracter ");
9     scanf("%lf", &num2);
10
11    double suma = num1 + num2;
12    double resta = num1 - num2;
13    double multiplicacion = num1 * num2;
14
15    printf("\nResultados:\n");
16    printf("Suma: %.4lf\n", suma);
17    printf("Resta: %.4lf\n", resta);
18    printf("Multiplicación: %.4lf\n", multiplicacion);
19
20    if (num2 == 0) {
21        printf("Error: No se puede dividir entre cero. \n");
22    }
23    else {
24        double division = num1 / num2;
25        printf("División: %.4lf\n", division);
26    }
27    return 0;
28 }
```

F. Resultados

Primer código corriendo dando los valores 1 para true y 0 para false.

```
C:\WINDOWS\system32\cmd. x +
1
0
c:\Codigos769>
```

Segundo código corriendo dando como los valores de las variables como las solicitamos.

```
C:\WINDOWS\system32\cmd. x + v
b 12 A 40.792
c:\Codigos769>
```

Tercer código corriendo dando como resultado las 4 operaciones con los dos números que le proporcionamos.

```
c:\Codigos769>calculadora
Ingrese el primer número: no un caracter 5
Ingrese el segundo número: no un caracter 5

Resultados:
Suma: 10.0000
Resta: 0.0000
Multiplicación: 25.0000
División: 1.0000

c:\Codigos769>
```

G. Conclusiones

- Aprendimos que la librería `stdbool.h` sirve para poder utilizar los valores de `true` y `false` en nuestro código.
- Aprendimos a declarar las variables enteras y la diferencia entre `double`, `char` y `int`.
- Aprendimos a poder solicitar y escanear un valor y asignarlo a una variable.
- Creamos una calculadora básica con lo aprendido.

H. Referencias

<https://programavideojuegos.blogspot.com/2014/02/la-libreria-stdbool.html>

[https://www.it.uc3m.es/pbasanta/asn-g/course notes/data types es.html#data types real numbers](https://www.it.uc3m.es/pbasanta/asn-g/course%20notes/data%20types%20es.html#data%20types%20real%20numbers)