



**UNIVERSIDAD DE SANTIAGO DE CHILE
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA INFORMÁTICA**

Estructura de Datos y Análisis de Algoritmos

Manual de Usuario

Alumno: Israel Arias Panez.

Profesor: Mario Inostroza.
Ayudante: Esteban Silva.

Santiago - Chile
2-2020

MANUAL DE USUARIO

3.1 INTRODUCCIÓN

El programa que se encuentra a su disposición le permite encontrar el camino más corto posible desde el planeta Tierra hasta el planeta Pizza Planet, asegurando que se podrá llegar sin problemas de quedar varado por falta de combustible. En este manual de usuario se le indicará las instrucciones de uso del programa, entradas, condiciones para que el programa funcione, entre otros.

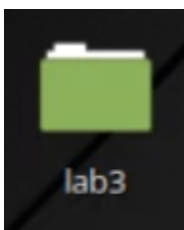
El programa se encuentra escrito en el lenguaje C, cuyo código se encuentra a su disposición.

3.2 CÓMO COMPILAR Y EJECUTAR

En esta sección se detallará, paso a paso como compilar y ejecutar el programa en dos sistemas operativos distintos, Linux y Windows.

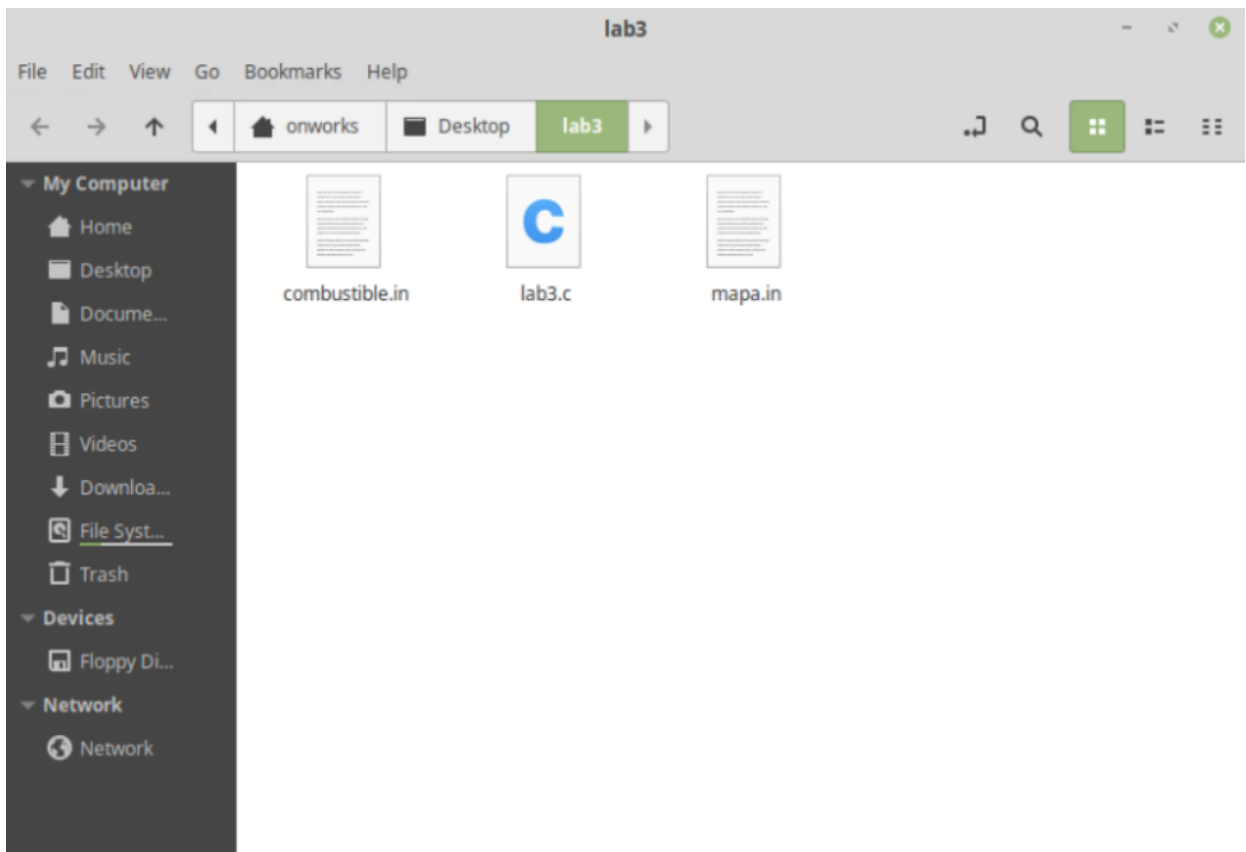
3.2.1 Compilación en Linux

En primer lugar, se debe verificar que se tengan todos los archivos necesarios en una carpeta



Se tomará como ejemplo en este caso la carpeta lab3, la carpeta puede tener el nombre que guste.

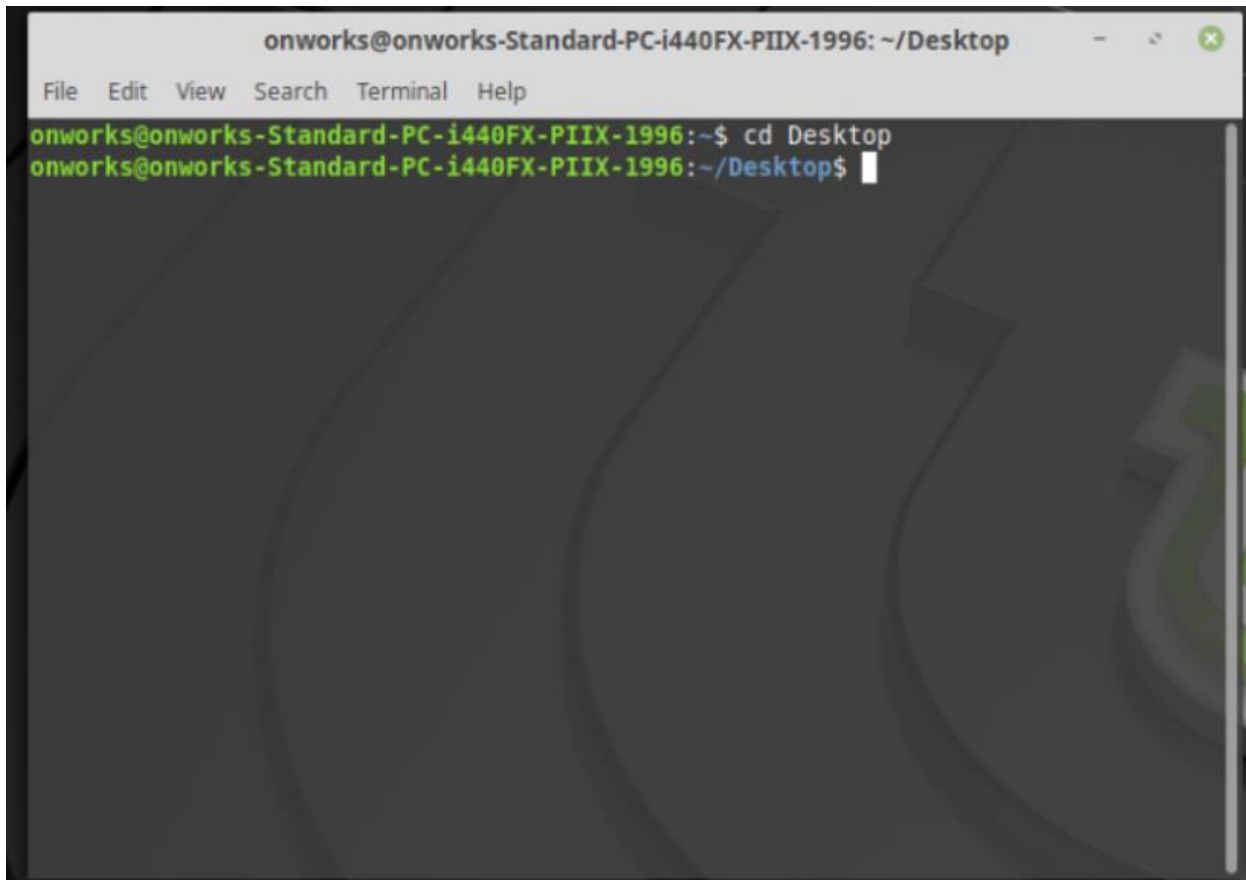
Se debe verificar que se tenga el programa lab3.c y las entradas necesarias para la ejecución del programa, en este caso son: combustible.in y mapa.in



A continuación, se debe abrir la terminal, para esto basta con hacer click en la terminal abajo en la barra de herramientas.

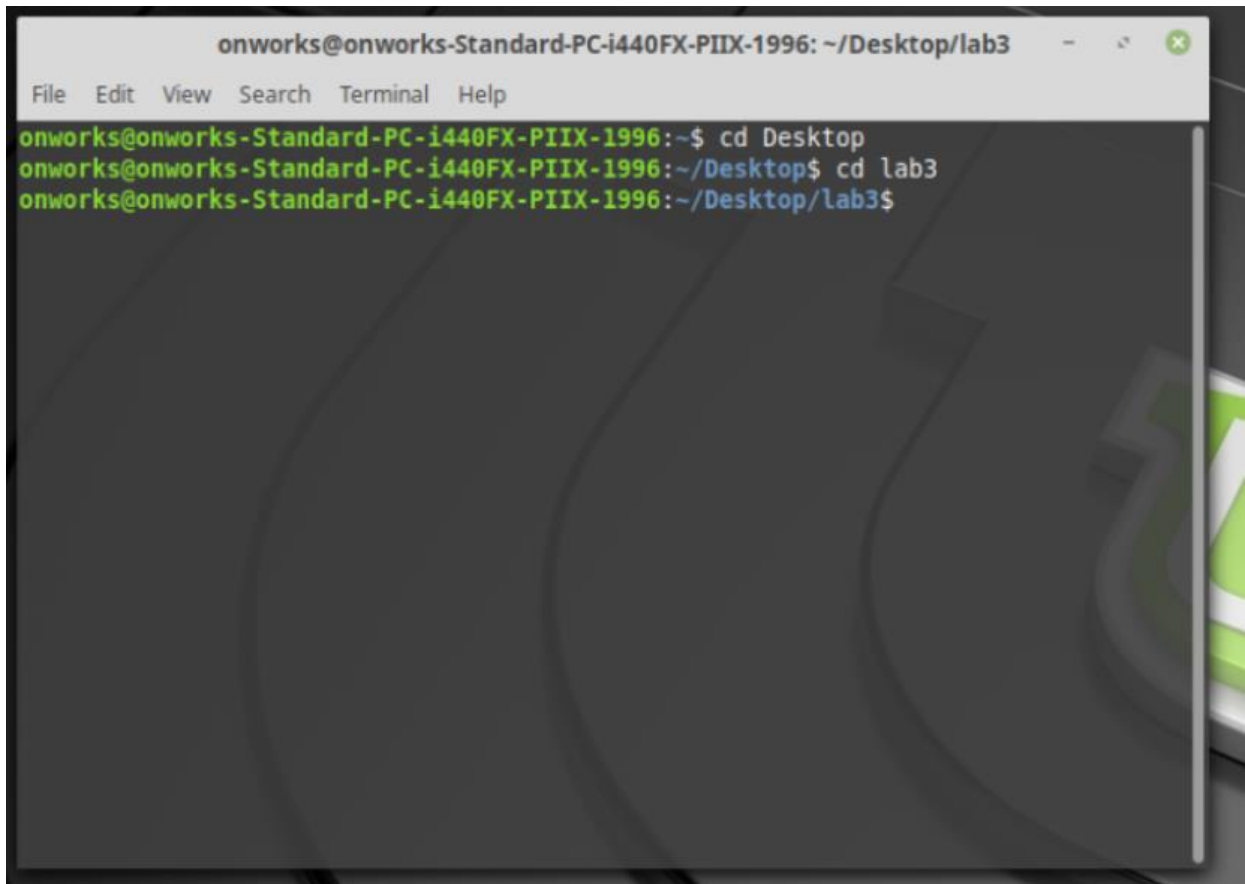


Se debería abrir la terminal, en la cual necesitamos ir al directorio o carpeta en el cual se tiene el programa, en este caso se encuentra en lab3, entonces se hace uso del comando cd Desktop, para ir a la carpeta desktop



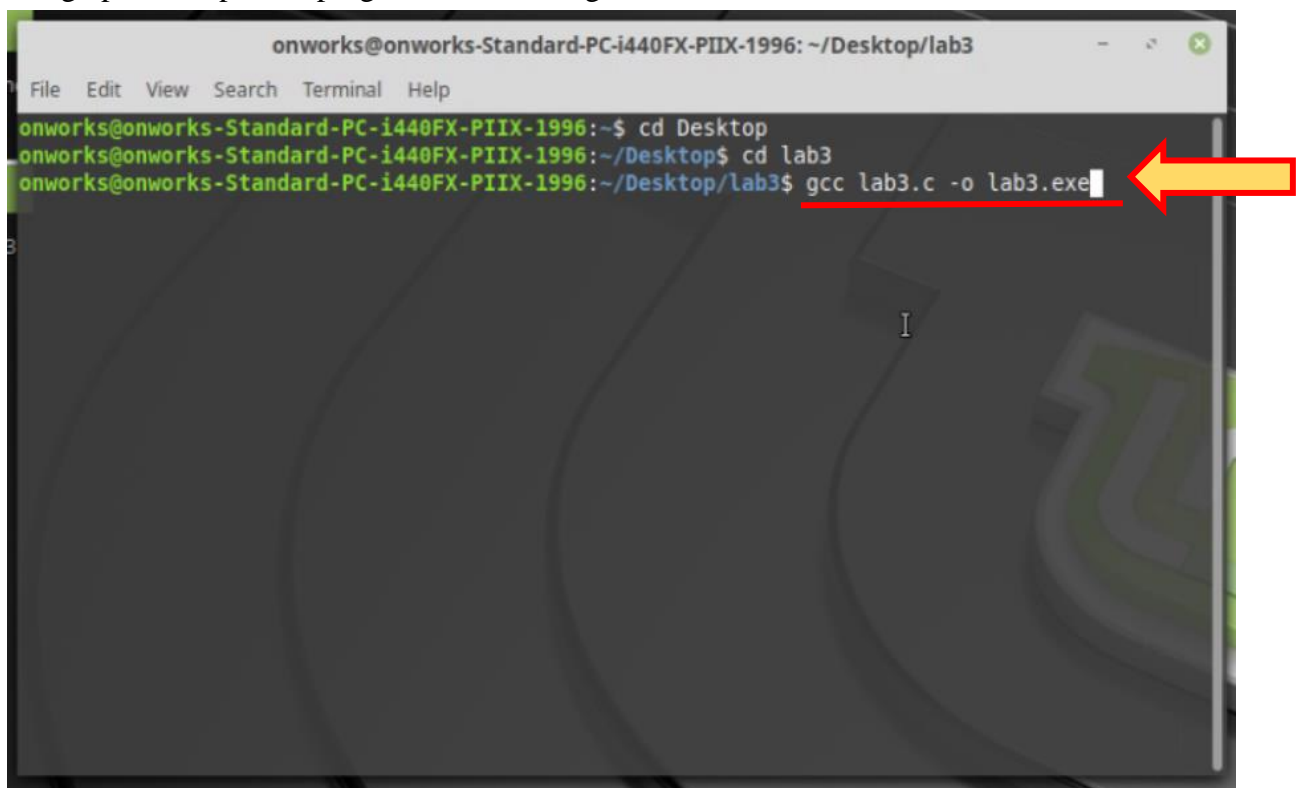
```
onworks@onworks-Standard-PC-i440FX-PIIX-1996: ~/Desktop
File Edit View Search Terminal Help
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ cd Desktop
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/Desktop$
```

Luego haciendo uso del mismo comando se accede a la carpeta lab3.



```
onworks@onworks-Standard-PC-i440FX-PIIX-1996: ~/Desktop/lab3
File Edit View Search Terminal Help
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ cd Desktop
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/Desktop$ cd lab3
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/Desktop/lab3$
```

Luego para compilar el programa se usa el siguiente comando



```
onworks@onworks-Standard-PC-i440FX-PIIX-1996: ~/Desktop/lab3
File Edit View Search Terminal Help
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ cd Desktop
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/Desktop$ cd lab3
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/Desktop/lab3$ gcc lab3.c -o lab3.exe
```

El comando es `gcc lab3.c -o lab3.exe`

// `lab3.c` es el nombre del programa y `lab3.exe` es el nombre que tendrá el ejecutable del programa, no es necesario que el ejecutable se llame `lab3.exe`, puede ser cualquier nombre a elección, sin embargo, si es necesario ingresar `lab3.c` ya que ese es el nombre del programa en su archivo.c

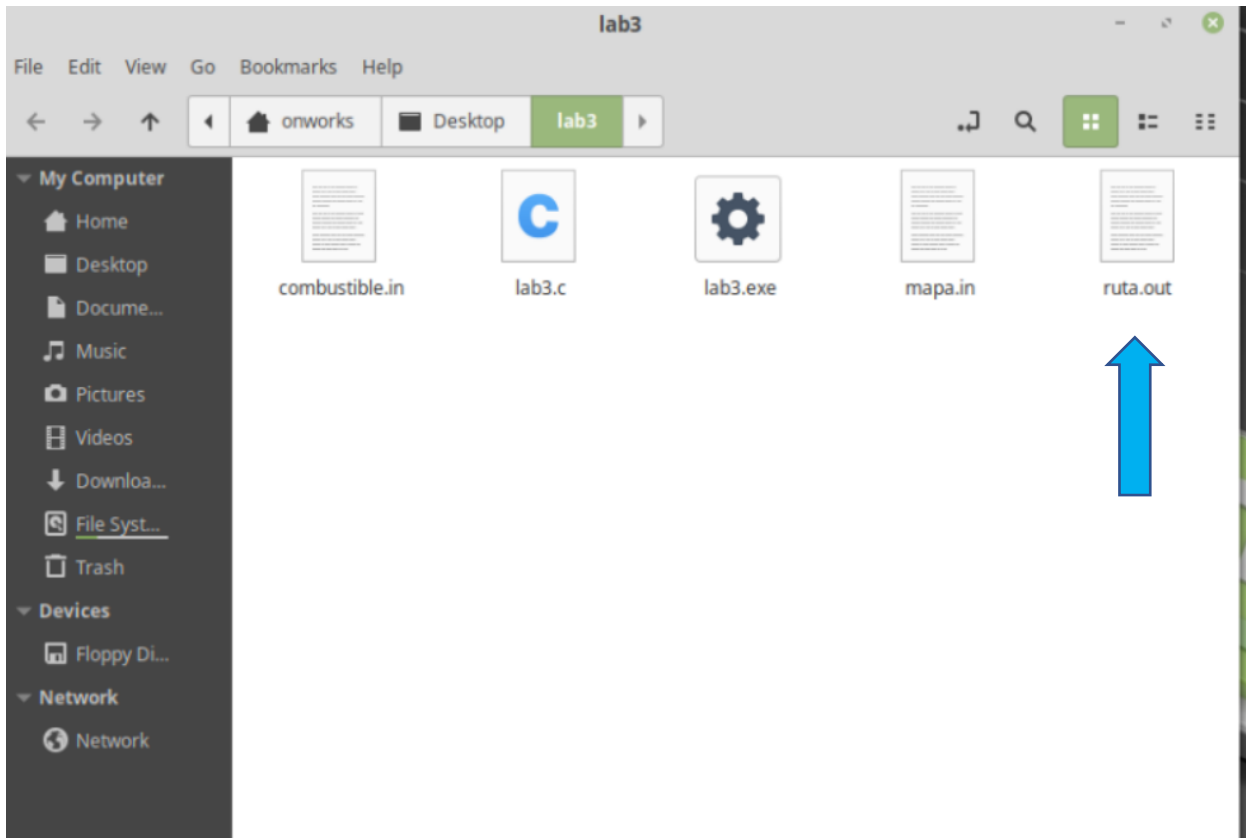
Ahora solo queda ejecutar el archivo .exe creado en la compilación anterior, para ello se usa el comando `./lab3.exe` es importante destacar que es necesario el punto y el slash antes de nombrar al ejecutable, si no, no funcionara. Además para esta ocasión se deben ingresar los nombres de los dos archivos de entrada, entonces en este caso el comando para la compilación corresponde a `./lab3.exe mapa.in combustible.in`



The screenshot shows a terminal window titled "onworks@onworks-Standard-PC-i440FX-PIIX-1996: ~/Desktop/lab3". The terminal has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The command history shows the following sequence of commands and output:

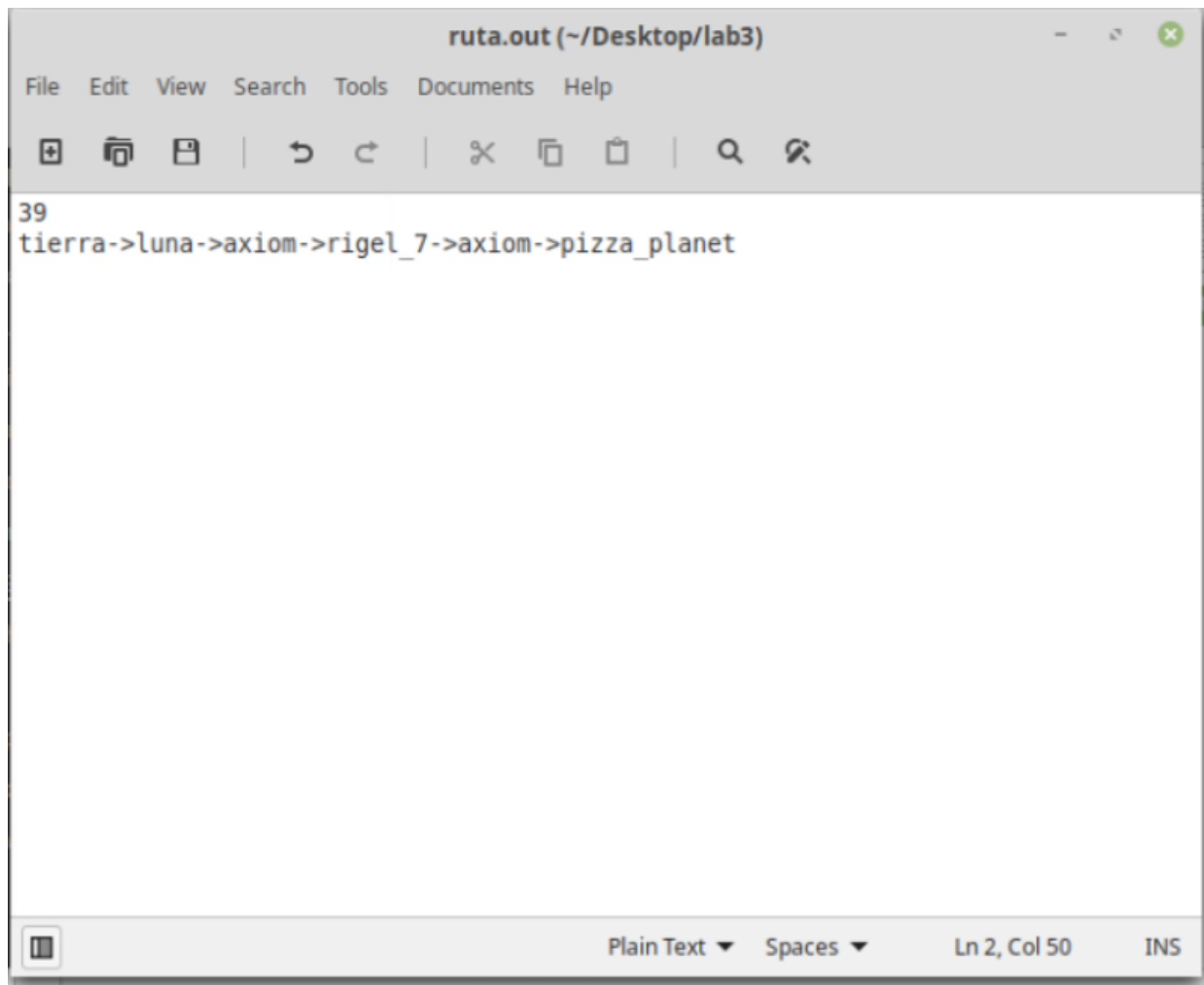
```
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~$ cd Desktop
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/Desktop/lab3$ ./lab3.exe mapa.in combustible.in
Archivo generado!
onworks@onworks-Standard-PC-i440FX-PIIX-1996:~/Desktop/lab3$
```

A red arrow points to the command `./lab3.exe mapa.in combustible.in` in the terminal.



Es posible observar que apareció el archivo ruta.out , ese archivo fue generado de acuerdo con las entradas de mapa.in y combustible.in y contiene la ruta más corta desde la Tierra hasta Pizza Planet

Al abrir el archivo será posible visualizar el resultado de la ejecución del programa.



The image shows a screenshot of a text editor window titled "ruta.out (~/Desktop/lab3)". The window has a menu bar with "File", "Edit", "View", "Search", "Tools", "Documents", and "Help". Below the menu bar is a toolbar with icons for file operations (new, open, save, print), editing (undo, redo, cut, copy, paste), and search. The main text area contains two lines of text: "39" on the first line and "tierra->luna->axiom->rigel_7->axiom->pizza_planet" on the second line. The status bar at the bottom indicates "Plain Text", "Spaces", "Ln 2, Col 50", and "INS".

```
39
tierra->luna->axiom->rigel_7->axiom->pizza_planet
```

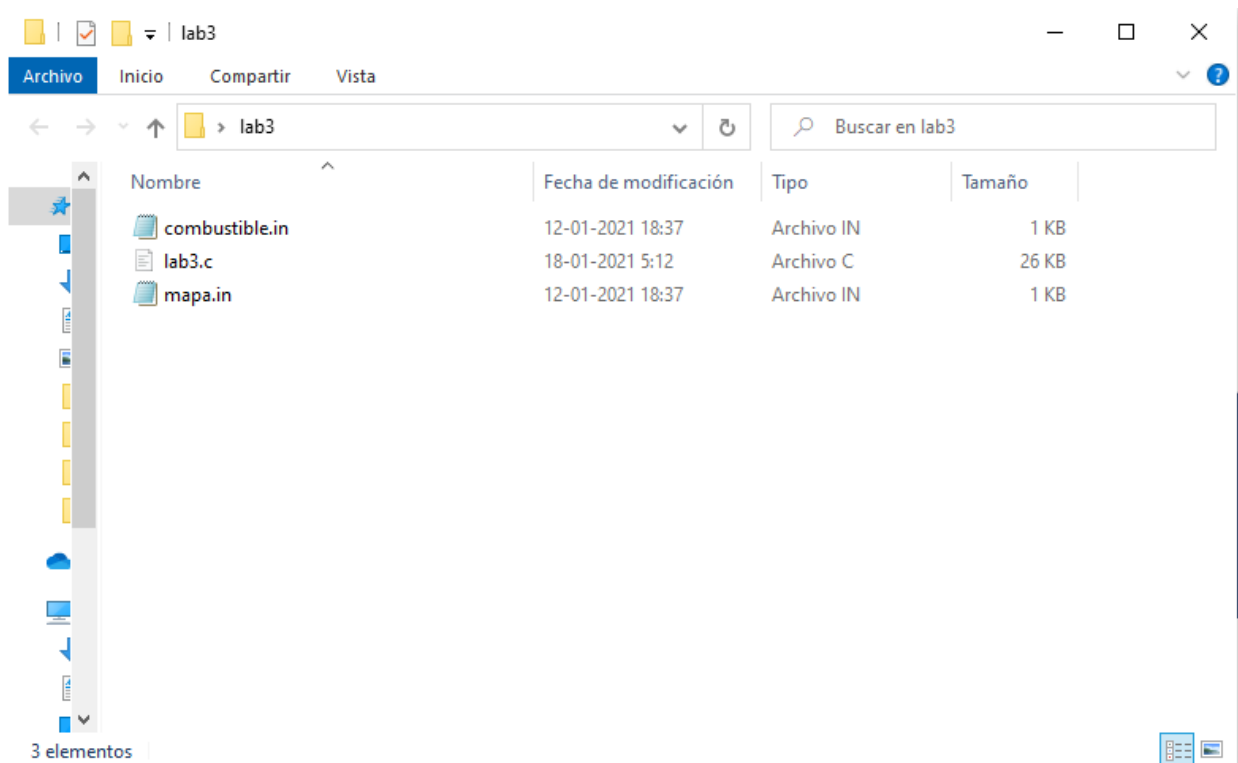
La primera línea indica la cantidad de tiempo que tomara el recorrido, para a continuación en la segunda línea mostrar el recorrido que se debe seguir.

3.2.2 Compilación en Windows

En primer lugar, se debe verificar que se tengan todos los archivos necesarios en una carpeta

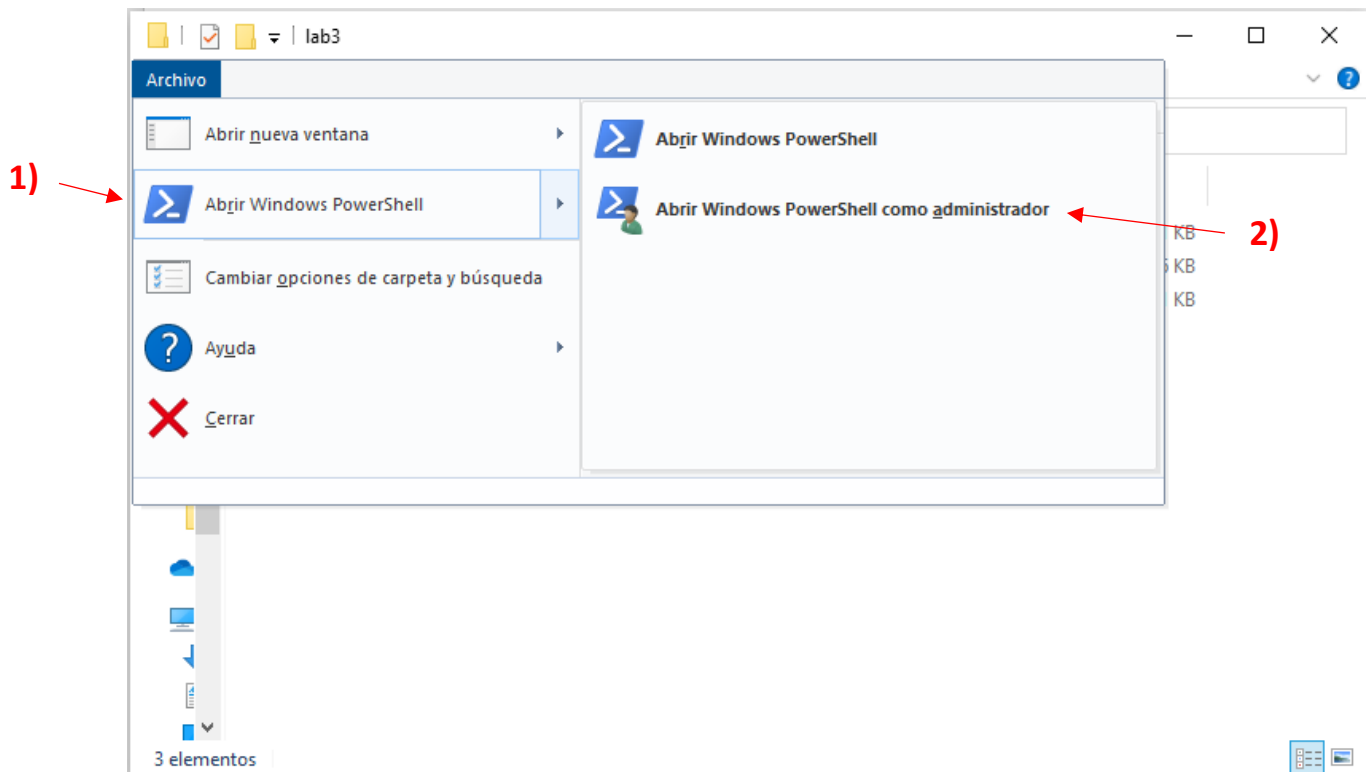


Se tomará como ejemplo en este caso la carpeta lab3, la carpeta puede tener el nombre que guste.

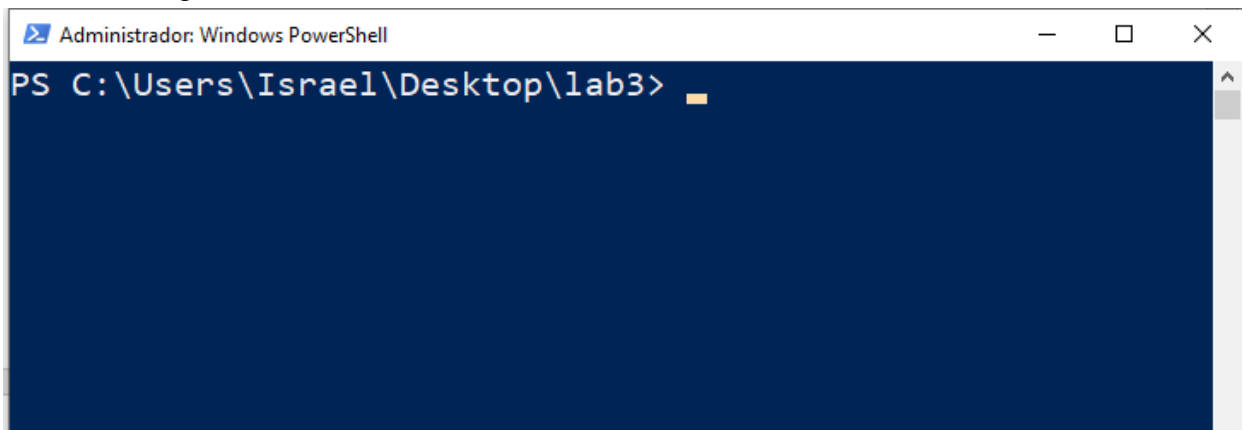


Se debe verificar que se tenga el programa lab3.c y las entradas necesarias para la ejecución del programa, en este caso son: combustible.in y mapa.in

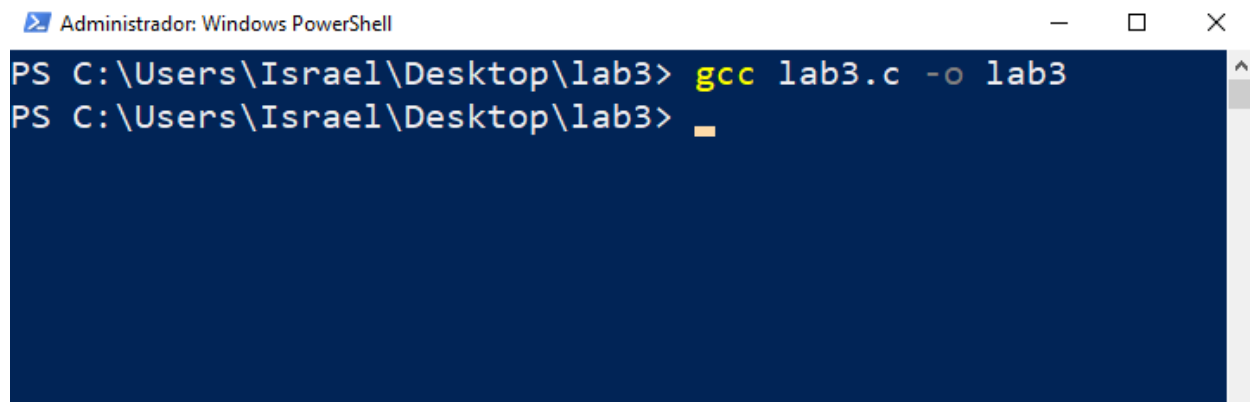
A continuación, se debe presionar en archivo en la esquina superior izquierda, luego abrir Windows PowerShell, **es importante, debe abrirse como administrador, si no se abre como administrador es posible que el programa no compile o no funcione de manera correcta.**



Se abrirá la siguiente ventana:



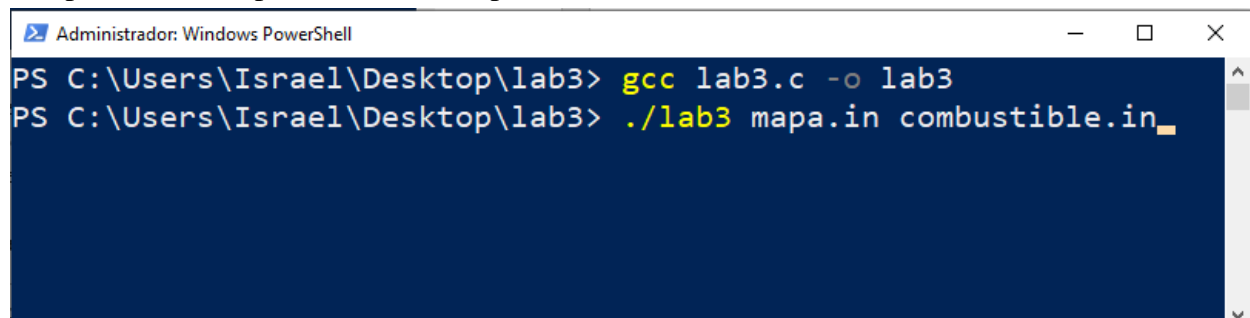
Ahora para compilar el programa se debe usar el comando `gcc lab3.c -o lab3`



```
Administrador: Windows PowerShell
PS C:\Users\Israel\Desktop\lab3> gcc lab3.c -o lab3
PS C:\Users\Israel\Desktop\lab3> 
```

// lab3.c es el nombre del programa y lab3 es el nombre que tendrá el ejecutable del programa, no es necesario que el ejecutable se llame lab3, puede ser cualquier nombre a elección, sin embargo, si es necesario ingresar lab3.c ya que ese es el nombre del programa en su archivo.c

Ahora solo queda ejecutar el archivo .exe creado en la compilación anterior, para ello se usa el comando `./lab3` es importante destacar que es necesario el punto y el slash antes de nombrar al ejecutable, si no, no funcionara. Además para esta ocasión se deben ingresar los nombres de los dos archivos de entrada, entonces en este caso el comando para la compilación corresponde a `./lab3 mapa.in combustible.in`



```
Administrador: Windows PowerShell
PS C:\Users\Israel\Desktop\lab3> gcc lab3.c -o lab3
PS C:\Users\Israel\Desktop\lab3> ./lab3 mapa.in combustible.in_
```

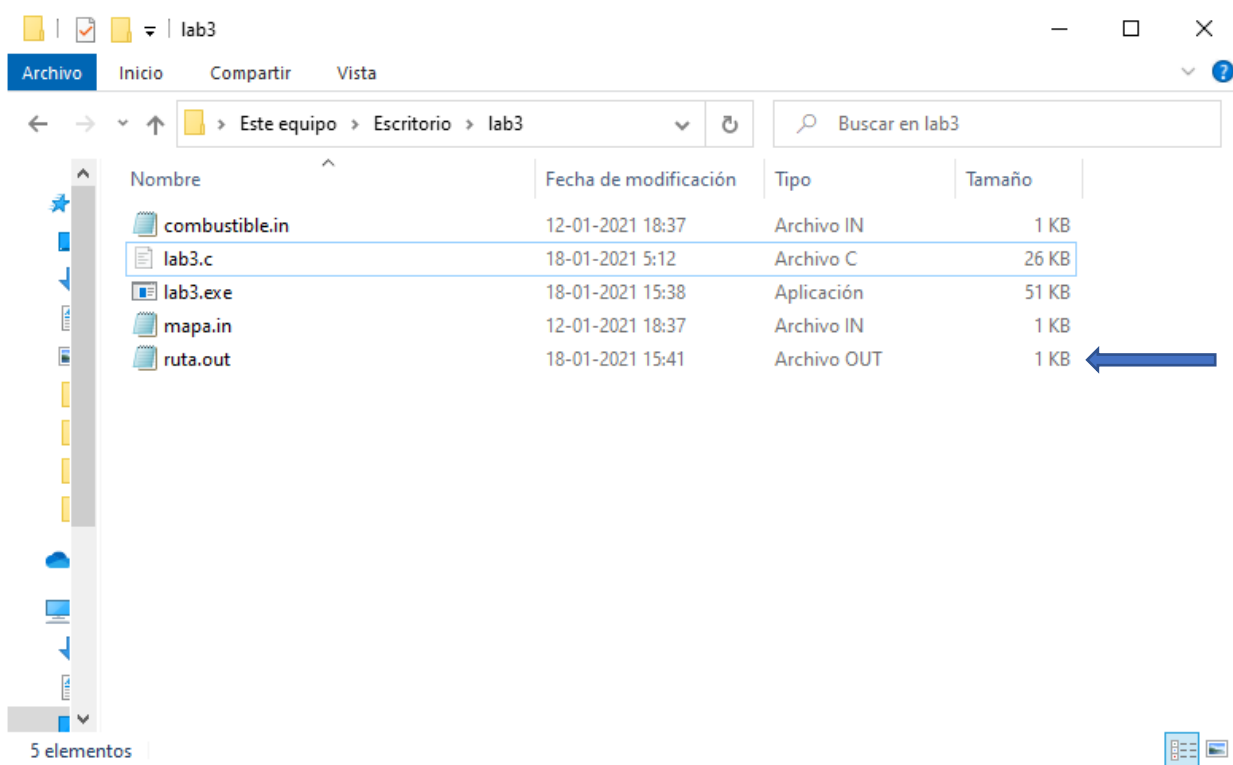
```
PS C:\Users\Israel\Desktop\lab3> gcc lab3.c -o lab3
PS C:\Users\Israel\Desktop\lab3> ./lab3 mapa.in combustible.in
tierra (0), pandora (1), tiempo: 16, gas: 6
tierra (0), luna (2), tiempo: 10, gas: 3
pandora (1), tierra (0), tiempo: 10, gas: 4
pandora (1), ego (3), tiempo: 8, gas: 4
pandora (1), rigel_7 (4), tiempo: 12, gas: 5
luna (2), tierra (0), tiempo: 5, gas: 2
luna (2), ego (3), tiempo: 15, gas: 5
luna (2), axiom (5), tiempo: 15, gas: 5
ego (3), pandora (1), tiempo: 10, gas: 10
ego (3), luna (2), tiempo: 10, gas: 5
ego (3), rigel_7 (4), tiempo: 10, gas: 5
rigel_7 (4), pandora (1), tiempo: 10, gas: 10
rigel_7 (4), ego (3), tiempo: 10, gas: 5
rigel_7 (4), axiom (5), tiempo: 5, gas: 2
rigel_7 (4), pizza_planet (6), tiempo: 10, gas: 10
axiom (5), luna (2), tiempo: 8, gas: 4
axiom (5), rigel_7 (4), tiempo: 5, gas: 2
axiom (5), pizza_planet (6), tiempo: 4, gas: 3
pizza_planet (6), rigel_7 (4), tiempo: 10, gas: 10
pizza_planet (6), axiom (5), tiempo: 4, gas: 3
```

Matriz:

```
( ) -1/-1 gasolinera?:0 | (pandora) 16/6 gasolinera?:0 | (luna) 10/3 gasolinera?:0 | ( ) -1/-1 gasolinera?:0 | ( ) -1/-1 gasolinera?:0 | ( ) -1/-1 gasolinera?:0 |
( ) -1/-1 gasolinera?:0 |
(tierra) 10/4 gasolinera?:0 | ( ) -1/-1 gasolinera?:0 | ( ) -1/-1 gasolinera?:0 | (ego) 8/4 gasolinera?:1 | (rigel_7) 12/5 gasolinera?:1 | ( ) -1/-1 gasolinera?:
0 | ( ) -1/-1 gasolinera?:0 |
(tierra) 5/2 gasolinera?:0 | ( ) -1/-1 gasolinera?:0 | ( ) -1/-1 gasolinera?:0 | (ego) 15/5 gasolinera?:1 | ( ) -1/-1 gasolinera?:0 | (axiom) 15/5 gasolinera?:0
| ( ) -1/-1 gasolinera?:0 |
( ) -1/-1 gasolinera?:0 | (pandora) 10/10 gasolinera?:0 | (luna) 10/5 gasolinera?:0 | ( ) -1/-1 gasolinera?:0 | (rigel_7) 10/5 gasolinera?:1 | ( ) -1/-1 gasoline
ra?:0 | ( ) -1/-1 gasolinera?:0 |
( ) -1/-1 gasolinera?:0 | (pandora) 10/10 gasolinera?:0 | ( ) -1/-1 gasolinera?:0 | (ego) 10/5 gasolinera?:1 | ( ) -1/-1 gasolinera?:0 | (axiom) 5/2 gasolinera?:
0 | (pizza_planet) 10/10 gasolinera?:0 |
( ) -1/-1 gasolinera?:0 | ( ) -1/-1 gasolinera?:0 | (luna) 8/4 gasolinera?:0 | ( ) -1/-1 gasolinera?:0 | (rigel_7) 5/2 gasolinera?:1 | ( ) -1/-1 gasolinera?:0 | (
pizza_planet) 4/3 gasolinera?:0 |
( ) -1/-1 gasolinera?:0 | ( ) -1/-1 gasolinera?:0 | ( ) -1/-1 gasolinera?:0 | ( ) -1/-1 gasolinera?:0 | (rigel_7) 10/10 gasolinera?:1 | (axiom) 4/3 gasolinera?:0
| ( ) -1/-1 gasolinera?:0 |
```

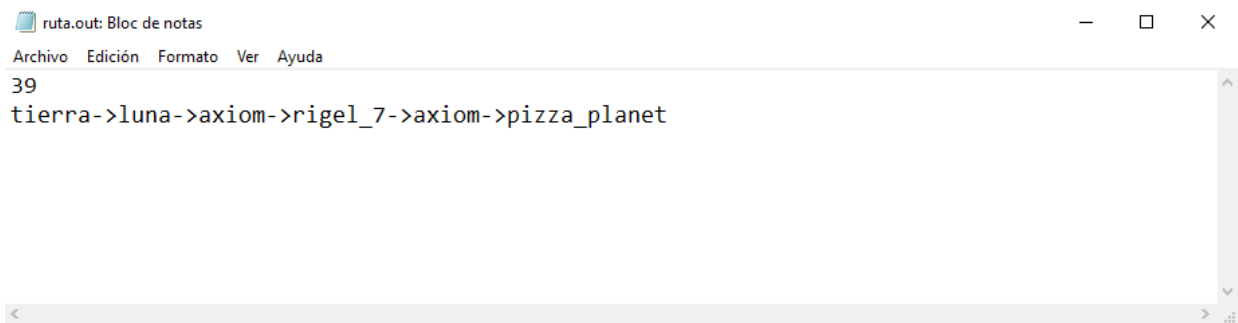
Archivo generado!

Ahora el programa se debería haber ejecutado sin problemas, entonces se debe volver a abrir la carpeta lab3, debiese verse así:



Es posible observar que apareció el archivo ruta.out , ese archivo fue generado de acuerdo con las entradas de mapa.in y combustible.in y contiene la ruta más corta desde la Tierra hasta Pizza Planet.

Y al abrir el archivo ruta.out, generado por el programa, el cual se encuentra señalado con una flecha celeste en la imagen anterior, es posible ver el resultado del programa:



```
ruta.out: Bloc de notas
Archivo Edición Formato Ver Ayuda
39
tierra->luna->axiom->rigel_7->axiom->pizza_planet
```

Debiese mostrarse lo que se ve en la imagen, el resultado de la ejecución del programa, donde en la primera línea se señala la cantidad de tiempo usada en recorrer desde la Tierra hasta Pizza Planet y en la segunda línea se entrega el recorrido que se debe seguir para llegar en la menor cantidad de tiempo hasta Pizza Planet.

3.3 FUNCIONALIDADES DEL PROGRAMA

Como se mencionó en la introducción, este programa posee una funcionalidad, la cual es la de encontrar el camino que menos tiempo tome desde la Tierra hasta Pizza Planet, tomando en consideración el combustible que posee la nave y las gasolineras existentes en el grafo entregado como entrada.

Para poder cumplir sus dos funcionalidades, el programa necesita dos entradas, las cuales deben ingresarse mediante archivos de texto plano con los nombres: “mapa.in” y “combustible.in”.

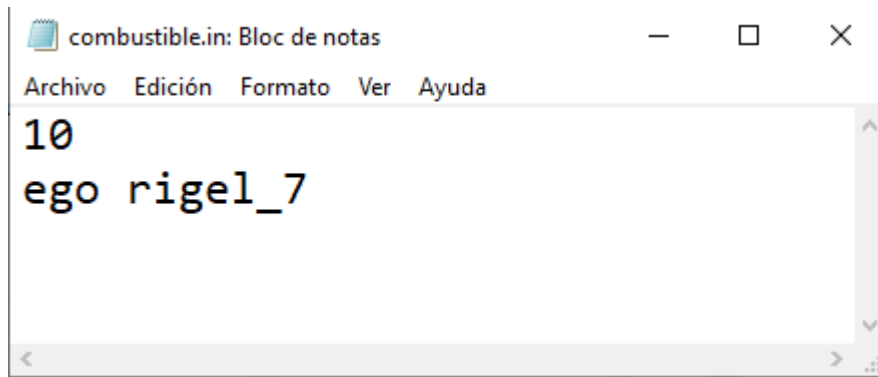
Ambos archivos de entrada llevan un formato que se debe respetar:



```
7
tierra pandora 16 6
tierra luna 10 3
pandora tierra 10 4
pandora ego 8 4
pandora rigel_7 12 5
luna tierra 5 2
luna ego 15 5
luna axiom 15 5
ego pandora 10 10
ego luna 10 5
ego rigel_7 10 5
rigel_7 pandora 10 10
rigel_7 ego 10 5
rigel_7 axiom 5 2
rigel_7 pizza_planet 10 10
axiom luna 8 4
axiom rigel_7 5 2
axiom pizza_planet 4 3
pizza_planet rigel_7 10 10
pizza_planet axiom 4 3
```

“mapa.in” en su primer valor debe indicar la cantidad de planetas o nodos que tiene el grafo para, a continuación, mencionar los nombres de los planetas, junto a la cantidad de tiempo que se gasta y el combustible, por ejemplo en la segunda línea del archivo presentado en la imagen anterior

significa “ir desde Tierra a Pandora toma 16 Unidades de tiempo y 6 de combustible”.



“combustible.in” en su primer valor debe indicar la cantidad de combustible inicial (y máximo) con el cual la nave empieza su recorrido, mientras que en la segunda línea se debe indicar los nombres de los planetas en los cuales se puede abastecer combustible, separados por un espacio.

Usted puede generar cualquier tipo de recorrido mínimo desde el planeta Tierra hasta Pizza Planet, para ello puede modificar los archivos “mapa.in” y “combustible.in”, siempre procurando respetar el formato presentado para cada archivo, o sea cumplir con el valor inicial, espacios, saltos de línea, etc.

En caso de no cumplir los formatos es probable que el programa no logre ejecutarse adecuadamente o no se ejecute directamente.

3.4 POSIBLES ERRORES

El error más probable es el de, al intentar compilar y ejecutar el programa, se debe entregar el nombre de los archivos de entrada, tomando como ejemplo la compilación en Windows, para ejecutar el programa una vez compilado se debe ingresar el comando `./lab3 mapa.in combustible.in`

Donde `mapa.in` y `combustible.in` son las entradas, las cuales en caso de haber ingresado mal su nombre provocara un error en el programa, sin embargo, la solución implementada contempla este posible error, por ende, dará aviso cuál de los dos archivos fue mal ingresado o presenta problemas, también podría darse el caso que alguno de los dos archivos directamente no exista, para ese caso también el programa alertara al usuario de aquello.

Otro posible error del programa es el de que el contenido de alguno de los archivos de entrada no cumpla con el formato presentado anteriormente, lo cual probablemente provoque errores en la ejecución de la solución implementada.

Otro error común puede ser el que se haya intentado ejecutar el programa dándole cero combustible como entrada, o que no exista un camino valido posible para llegar a Pizza Planet, en este caso la solución implementada no tiene como saber reconocer esos casos, por lo que probablemente arroje un error de segmentación o un “Stack Overflow” por haber quedado atrapado en un bucle.

Limitaciones

La principal limitación que tiene el programa es que tiene una memoria fija de 100 caracteres para leer y almacenar el nombre de los planetas en los archivos `mapa.in` y `combustible.in`, en caso de que el nombre tuviera mas de 100 caracteres, el programa no podría ejecutarse ya que no sería capaz de almacenar el nombre de ese planeta, causando un error de segmentación (segmentation fault), esta limitación se da en ambos Sistemas operativos, Windows y Linux. Aunque es posible aumentar esta capacidad indagando en el código fuente y modificando el parámetro de memoria de los arreglos.