



Universidad de Santiago de Chile
Facultad de Ingeniería
Departamento de Ingeniería Informática

Estructura de datos y análisis de algoritmos

Manual de usuario

Christofer Rodríguez

Profesor: Mario Inostroza

Ayudante: Estaban Silva

3.1 Introducción

El programa que usted posee actualmente, le permite resolver el problema de “salvación espacial”, para el correcto funcionamiento de este programa se deben seguir las instrucciones indicadas a continuación en este manual.

Salvación espacial consiste en obtener todas las combinaciones de tripulantes que cumplan con el peso máximo permitido por una nave espacial, en caso de que la combinación no cumpla con esta condición, no será considerada como una solución al problema. Las soluciones que cumplan con la condición serán entregadas en un archivo de texto ordenadas de manera decreciente con respecto a la calificación total de su tripulación.

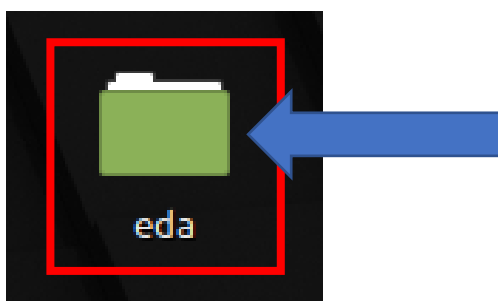
El programa entregado al usuario es el código original creada en el lenguaje de programación C.

3.2 Cómo compilar y ejecutar

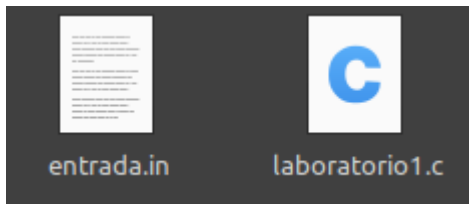
En la siguiente sección se indicará paso a paso la manera de compilar y ejecutar el programa en dos ambientes diferentes:

3.2.1 Compilación para Linux

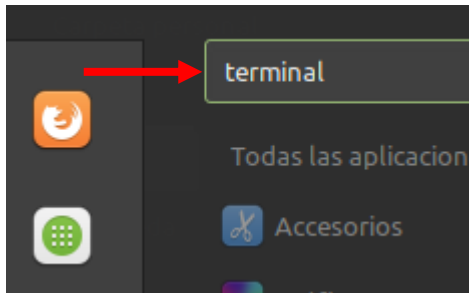
Para poder compilar el archivo, primero debe tener los archivos dentro de una carpeta.



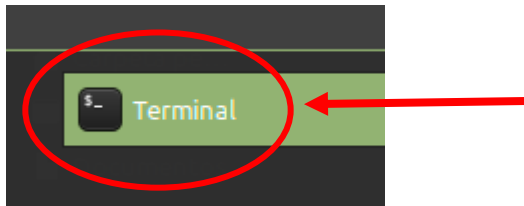
1. Como se observa en la imagen, se remarca una carpeta donde deben estar los archivos del programa, para un fácil acceso, se ha elegido el escritorio a modo de ejemplo.



2. Al abrir la carpeta debe asegurarse de que esta contenga los siguientes 2 archivos.

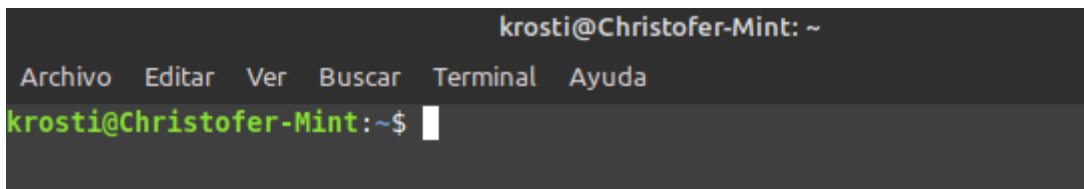


3. Una vez comprobados los archivos, necesitaremos acceder a la terminal, para esto se debe abrir el buscador y escribir la palabra "terminal".



4. Una vez aparecido el icono de la aplicación, debe clickear para abrirla.

Una vez abramos la aplicación aparecerá la siguiente ventana emergente:



5. En esta ventana emergente es la terminal, aquí debe ingresar líneas de comando para poder hacer uso de su programa.

*Es muy importante que luego de escribir una línea de comando en la terminal, se debe presionar la tecla "enter", de esta manera se ejecutará el comando ingresado.

```
kroستي@Christofer-Mint
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
kroستي@Christofer-Mint:~$ cd Escritorio/eda
```

6. Ahora es necesario ingresar a la dirección donde se encuentra nuestro programa, para esto debemos escribir la palabra “cd” “espacio” seguido de la dirección donde se encuentra en este caso es “Escritorio”, “/” inmediatamente seguido del nombre de la carpeta donde se encuentran los archivos, en este ejemplo es “eda”.

Ahora que ya hemos accedido a la carpeta de los archivos desde la terminal, procederemos a compilar el programa ingresando las siguientes líneas de comando:

```
kroستي@Christofer-Mint:~/Escritorio/eda$ gcc laboratorio1.c -o laboratorio.exe
```

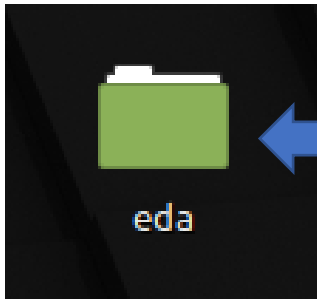
7. Se debe escribir “gcc” “espacio” “nombre del programa con su extensión” “espacio” “-o” “nombre del ejecutable.exe”.

Cabe destacar que el primer elemento destacado corresponde al nombre del archivo, este nombre puede variar, pero siempre debe ser igual al nombre actual del programa; mientras que el segundo elemento destacado es el nombre del ejecutable, este nombre puede ser cambiado a gusto, pero cuando se desee ejecutar, se debe hacer con el nombre dado.

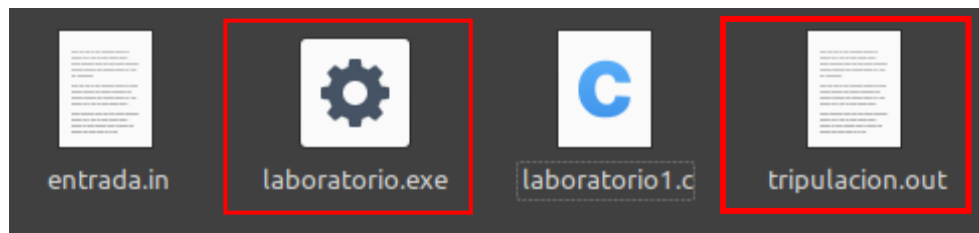
Finalmente, solo resta escribir la última línea de comando para ejecutar el programa:

```
kroستي@Christofer-Mint:~/Escritorio/eda$ gcc laboratorio1.c -o laboratorio.exe
kroستي@Christofer-Mint:~/Escritorio/eda$ ./laboratorio.exe entrada.in
```

8. Aquí se debe escribir “./” inmediatamente seguido el nombre del ejecutable (en este caso es “laboratorio.exe”) “espacio” y “nombre del archivo de entrada” (para este ejemplo, el archivo de entrada tiene el nombre de “entrada.in”).



9. Nuevamente debemos dirigirnos a la carpeta donde tenemos los archivos.



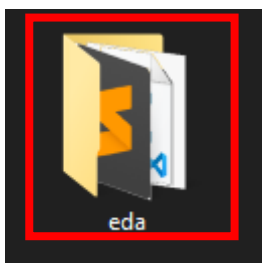
10. Dentro de la carpeta encontraremos un nuevo archivo que corresponde al ejecutable del programa y otro llamado “tripulación.out” que será el archivo con la solución generada al archivo de entrada.

11. Ahora solamente debemos abrir el archivo de salida para poder visualizar el resultado.

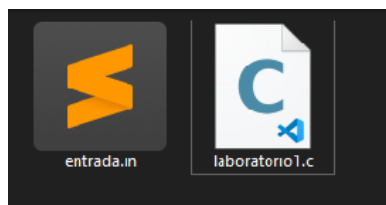
```
tripulacion.out x
postulante1, postulante2, postulante4, postulante5, 530, 285
postulante1, postulante3, postulante5, 770, 258
postulante2, postulante5, postulante3, 740, 243
postulante1, postulante3, postulante5, 790, 238
postulante1, postulante2, postulante4, 440, 235
postulante2, postulante5, postulante3, 760, 223
postulante3, postulante5, postulante5, 660, 218
postulante1, postulante2, postulante5, 460, 215
```

3.2.2 Compilación para Windows

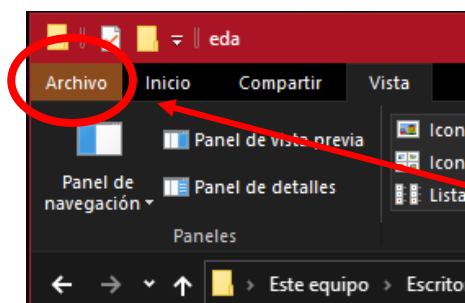
Para poder compilar en Windows, se asume que debe tener instalado MinGW y Windows PowerShell.



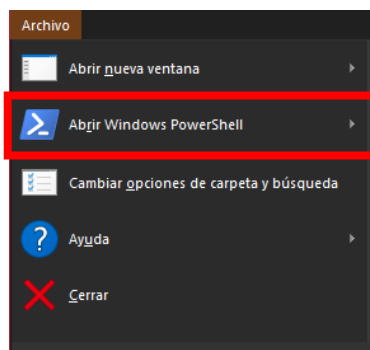
1. Para comenzar, debemos tener los archivos dentro de una carpeta, se recomienda elegir una carpeta de fácil acceso; para el caso de este ejemplo, se ha elegido una carpeta en el escritorio (Desktop).



2. A continuación, se debe abrir la carpeta y confirmar que se encuentran los siguiente 2 archivos.

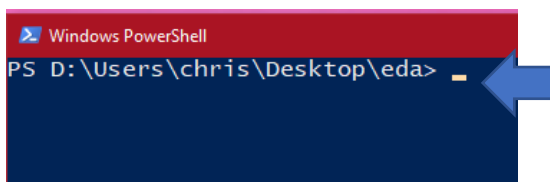


3. Una vez dentro de la carpeta, debemos presionar la pestaña llamada “Archivo”, como se puede apreciar en la imagen, esta se encuentra en la esquina superior izquierda de la ventana.

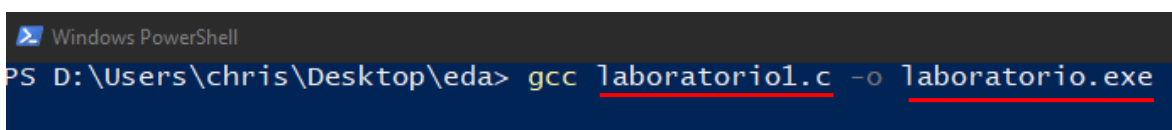


4. Una vez presionada la pestaña, se desplegará una ventana como la que se muestra en la imagen, aquí debemos presionar la opción que se muestra señalada “Abrir Windows PowerShell”.

Al seleccionar esta opción, la siguiente ventana emergente será mostrada:

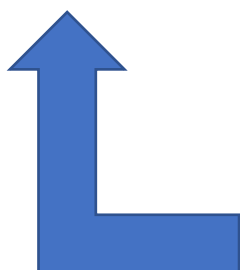
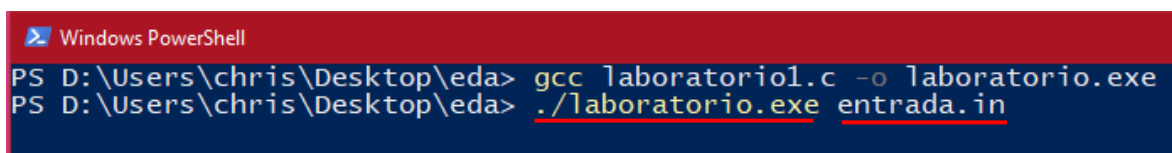


5. En esta consola de comandos, debe ingresar los comandos indicados a continuación para compilar y posteriormente ejecutar el programa.



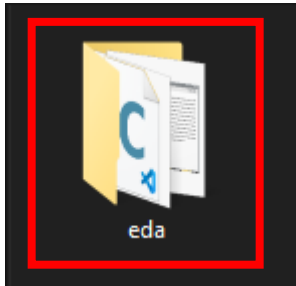
6. Primero se debe escribir “gcc” “espacio” “nombre del programa con su extensión” “espacio” “-o” “espacio” “nombre del ejecutable”, luego de escribir el comando, es necesario presionar la tecla “enter” para su ejecución.
- 7.

Cabe mencionar que si bien, el primer nombre debe siempre corresponder al nombre del archivo (primer elemento destacado en la imagen), el segundo corresponde al nombre del archivo ejecutable (segundo elemento destacado en la imagen) y este puede ser modificado.

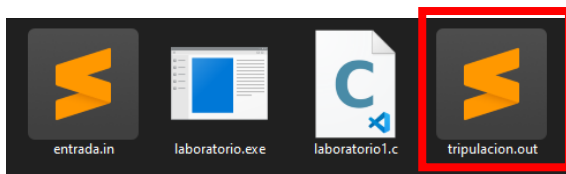


8. Una vez pasados unos segundos, volverá a aparecer la ruta de la carpeta (texto de la izquierda) y ahora debemos escribir “./” seguido inmediatamente del “nombre del ejecutable” (en el caso de este ejemplo es “laboratorio.exe”) “espacio” y el nombre del archivo de entrada que puede tener cualquier nombre (en el caso de este ejemplo es “entrada.in”) y presionar la tecla “enter”.

Dependiendo de su computador y el archivo de entrada, esto puede tardar unos segundos.



9. Una vez terminado de ejecutarse el programa (volverá a aparecer el texto de la izquierda en la consola), debe dirigirse a la carpeta donde se encontraban sus archivos.



10. Una vez dentro de la carpeta, se encontrará con dos archivos nuevos, uno es el archivo ejecutable del programa (“laboratorio.exe”) y el otro es el archivo de salida del programa, este archivo tendrá el nombre de “tripulación.out” (archivo resaltado en la imagen).

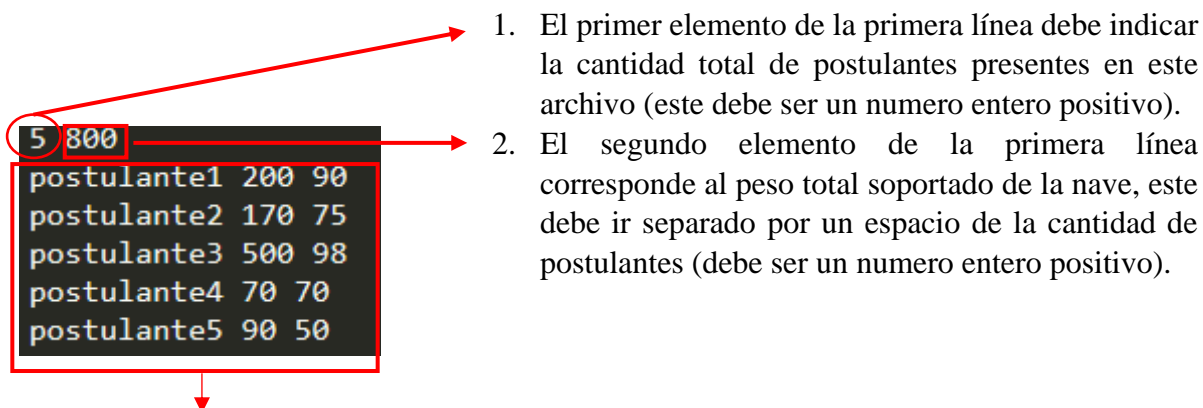
11. Ahora solo resta hacer doble click en el archivo “tripulación.out” para poder ver las soluciones generadas en base al archivo de entrada ingresado.

```
tripulacion.out: Bloc de notas
```

Archivo	Edición	Formato	Ver	Ayuda
postulante1,	postulante2,	postulante4,	postulante5,	530, 285
postulante1,	postulante3,	postulante5,	770,	258
postulante2,	postulante5,	postulante3,	740,	243
postulante1,	postulante3,	postulante5,	790,	238
postulante1,	postulante2,	postulante4,	440,	235
postulante2,	postulante5,	postulante3,	760,	223
postulante3,	postulante5,	postulante5,	660,	218
postulante1,	postulante2,	postulante5,	460,	215
postulante1,	postulante4,	postulante5,	360,	210
postulante2,	postulante5,	postulante4,	330,	195

3.3 Funcionalidades del programa

El presente programa es capaz de generar todas las combinaciones de tripulantes para la nave, siempre y cuando se respete la siguiente estructura en los archivos de entrada:



The diagram shows a text file with the following content:

```
5 800
postulante1 200 90
postulante2 170 75
postulante3 500 98
postulante4 70 70
postulante5 90 50
```

Red arrows indicate the structure:

- 1. El primer elemento de la primera línea debe indicar la cantidad total de postulantes presentes en este archivo (este debe ser un numero entero positivo).
- 2. El segundo elemento de la primera línea corresponde al peso total soportado de la nave, este debe ir separado por un espacio de la cantidad de postulantes (debe ser un numero entero positivo).

3. Desde la segunda línea en adelante se debe encontrar el nombre del postulante, su peso separado por un espacio del nombre y su calificación, separada por un espacio de su peso (tanto el peso como la calificación deben ser números enteros positivos).

Es importante que el nombre de los postulantes no contenga tildes o la letra “ñ”, ya que esto puede causar un error inesperado en el proceso de ejecución.

3.4 Posibles errores

Existen algunos errores que pueden ocurrir al momento de ingresar los datos en el archivo de entrada, estos se pueden evitar realizando las siguientes acciones:

En caso de que se ingrese un numero invalido dentro de la sección de cantidad de postulantes, el peso máximo de la nave, el peso de un postulante o su calificación, el programa mostrará dentro de la consola de comando un mensaje indicando que existe un dentro del archivo de entrada y terminará su ejecución, tal como se muestra en la siguiente ilustración:

```
0 -800
postulante1 200 90
postulante2 -170 75
postulante3 500 98
postulante4 70 70
```

```
PS D:\Users\chris\Documents\Eda> ./lab9 entrada.in
Error en el formato del archivo de entrada
PS D:\Users\chris\Documents\Eda>
```

Si esto ocurre, revise el archivo de entrada y asegúrese de que ingreso los datos respetando la estructura explicada anteriormente en este documento.

Otro error común puede existir al momento de ejecutar el programa, en caso de que el nombre del archivo de entrada sea incorrecto o no exista, se mostrara un mensaje en la consola indicando que el archivo no existe y el programa terminará su ejecución.

```
PS D:\Users\chris\Documents\Eda> ./lab9 entada.in
El archivo de entrada no existe
PS D:\Users\chris\Documents\Eda>
```

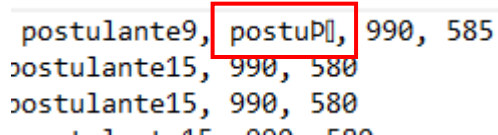
En caso de que esto ocurra, revise el nombre que tiene su archivo de entrada y vuelva a ejecutar el programa con su nombre escrito correctamente.

3.5 Limitaciones del programa

La manera en la que se implementó el programa tiene una pequeña limitación, esta puede afectar tanto a Windows como a Linux, pero la solución es la misma para ambos ambientes.

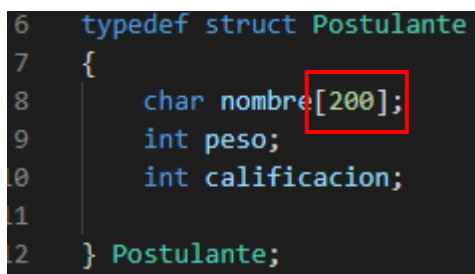
Este problema es provocado por la manera en la que se implementó la solución al problema, debido a que el lugar donde se almacena la combinación de los nombres de los tripulantes que componen una combinación, solo es capaz de almacenar 200 caracteres.

Por lo que, si el nombre de los postulantes es muy largo o el límite de peso de la nave es muy alto, la combinación de nombres será muy larga como para almacenarla, esto provocará que dentro del archivo de salida el nombre de los tripulantes de sea vea cortado de la siguiente manera:



```
postulante9, postuP[], 990, 585  
postulante15, 990, 580  
postulante15, 990, 580  
: : : 15 888 588
```

Para solucionar este problema se debe abrir el archivo del programa (por defecto “laboratorio1.c”), dirigirse a la línea número 6 del documento y cambiar el valor resaltado en la siguiente imagen:



```
6  typedef struct Postulante  
7  {  
8      char nombre[200];  
9      int peso;  
10     int calificacion;  
11  
12 } Postulante;
```

Solamente basta con cambiar el numero dentro de los paréntesis cuadrados por un número más alto, guardar los cambios y volver a compilar como se enseñó en este documento para poder ejecutarlo correctamente.

Cabe destacar que no se debe cambiar este valor por uno muy elevado, ya que esto utilizara más recursos de su computador y con un número elevado de postulantes, puede ralentizar el proceso o generar un error inesperado.