

## PRACTICA 2

### Objetivo General

Desarrollar dos algoritmos sencillos en C que permitan, implementar un Cifrado y De cifrado César y desarrollar un árbol de factor de factorización y nivel dado por el usuario.

### Objetivos Particulares:

- Desarrollar dos programas en C puro para implementar cada uno de los algoritmos del objetivo general de la práctica.
- La parte de hacerlo sobre C puro se refiere al desarrollo de ambos programas con un editor de textos o un IDE y realizar el proceso de compilación del programa con una terminal (sea Linux o Windows).
- Para la primera parte que tiene que ver con la implementación de un cifrado César, tomar de apoyo lo que se revise en la sesión de examen del martes 25 de febrero del 2025. Este programa tiene que abrir un archivo, que contendrá el mensaje original y mostrará en pantalla el cifrado y el descifrado final del mensaje.
- La segunda implementación se debe retomar el TDA de tipo Lista Ligada Simple, lo cual permitirá comprender el concepto de apuntadores y asignación de memoria dinámica que será mucha de ayuda en gran parte del curso.

### Documentos para entregar

- Ambos programas en C, los cuales serán enviados por correo antes de que empiece el examen del próximo martes.
- Informe de lo realizado en la práctica, con su portada, introducción, desarrollo y conclusiones usando cualquier editor de texto.

### Plazo de entrega

La hora y fecha límite para enviar los códigos fuente y el informe antes de que empiece el laboratorio del martes 04 de marzo del 2025, ya que la próxima semana no habrá clases ese día en particular.

**Nota. No se recibirá ninguna práctica fuera de ese horario, sin ninguna excepción.**

## Especificaciones del programa para entregar:

### Cifrado y Decodificado César:

Para esta parte es importante que se presente al examen de la Semana 3 del martes 25 de enero del 2025, sin embargo, se muestra en la Figura 2 un ejemplo de la salida que se desea para el cifrado y descifrado César:

Se deja libre tanto la implementación como la salida del descifrador, siempre que conserve los fines establecidos en el objetivo de la práctica.



```
G:\Documentos_Benja\Trabajo_UAM\Trimestre_17-O\Materia - Programaci...n Concurrente\Pruebas\Practicas_Compiladores\Practica1\Practica1_Cifrado_Cesa...
Fichero: telegrama_Zimmermann.txt -> existe (ABIERTO)

El mensaje cifrado del fichero telegrama_Zimmermann.txt es:

Tuy vxuvutksuy iusktfgx kr vxoskxu jk lkhxkxu rg makxxg yahsgxotg, yot xkyzxoiiout. Tu uhyzgtzk, tuy kyluxfgxks
uy vgxg sgtzktkx rg tkazxgrojgj jk ruy Kyzgjuy Atojuy jk Gskxoig.

Kt igyu jk tu zktkx kdozu, vxuvutksuy g Skdoi atg grogtfg yuhxk rgy yomaoktzky hgyky: ngikx patzuy rg makxxg,
jkingxg patzuy rg vgf; gvuxzgxsuy ghatjgtzk geajg lotgtioxg; e kr ktzktjosoktzu vux takyzxg vgxzk jk wak Skd
oiu ng jk xkiutwaoyzg kr zkxxozouxou vxkjoju kt Takbu Skdoi, Zkdeg e Gxofutg. Ruy jkzgrky jkr giakxju wakjgt
g ya joyixkiout jk But Kiqgxjz.

Wakjg ayzkj ktigxmgju jk otluxsgx gr vxkyojktzk jk Skdoi jk zuju ru gtzkjoinu, jk rg luxsg sgy ykixkzg vuyohrk
, zgt vxutzu iusu kr kyzgrroju jk rg makxxg iut ruy Kyzgjuy Atojuy jk Gskxoig ykg at nkinu ykmaxu. Jkhk gjksy
yamkxoxrk wak zusk rg otoiogzobg jk otbozgx g Pgvut g gjnkxoxyk jk luxsg otskjogzg g kyzk vrgt, ulxkio|@tjuyk g
r soysu zoksvu iusu skjogjux ktzxx Pgvut e tuyuzxuy.

Ngmg tuzgx gr Vxkyojktzk wak kr ayu jkyvogjgju jk takyzxuy yahsgxotuy eg ngik vxkboyohrk wak Otmgzkxxg yk bkg
uhromjg g vkjox rg vgf kt ruy vxudosuy skyky.

El mensaje decifrado del fichero telegrama_Zimmermann.txt es:

Nos proponemos comenzar el primero de febrero la guerra submarina, sin restriccion. No obstante, nos esforzarem
os para mantener la neutralidad de los Estados Unidos de America.

En caso de no tener exito, proponemos a Mexico una alianza sobre las siguientes bases: hacer juntos la guerra,
declarar juntos la paz; aportaremos abundante ayuda financiera; y el entendimiento por nuestra parte de que Mex
ico ha de reconquistar el territorio perdido en Nuevo Mexico, Texas y Arizona. Los detalles del acuerdo quedan
a su discrecion de Von Eckardt.

Queda usted encargado de informar al presidente de Mexico de todo lo antedicho, de la forma mas secreta posible
, tan pronto como el estallido de la guerra con los Estados Unidos de America sea un hecho seguro. Debe ademas
sugerirle que tome la iniciativa de invitar a Japon a adherirse de forma inmediata a este plan, ofreci|@ndose a
l mismo tiempo como mediador entre Japon y nosotros.

Haga notar al Presidente que el uso despiadado de nuestros submarinos ya hace previsible que Inglaterra se vea
obligada a pedir la paz en los proximos meses.

.
Fichero cerrado

-----
Process exited after 0.3032 seconds with return value 0
Presione una tecla para continuar . . .
```

**Figura 2.** Aspecto visual de la segunda parte de la practica

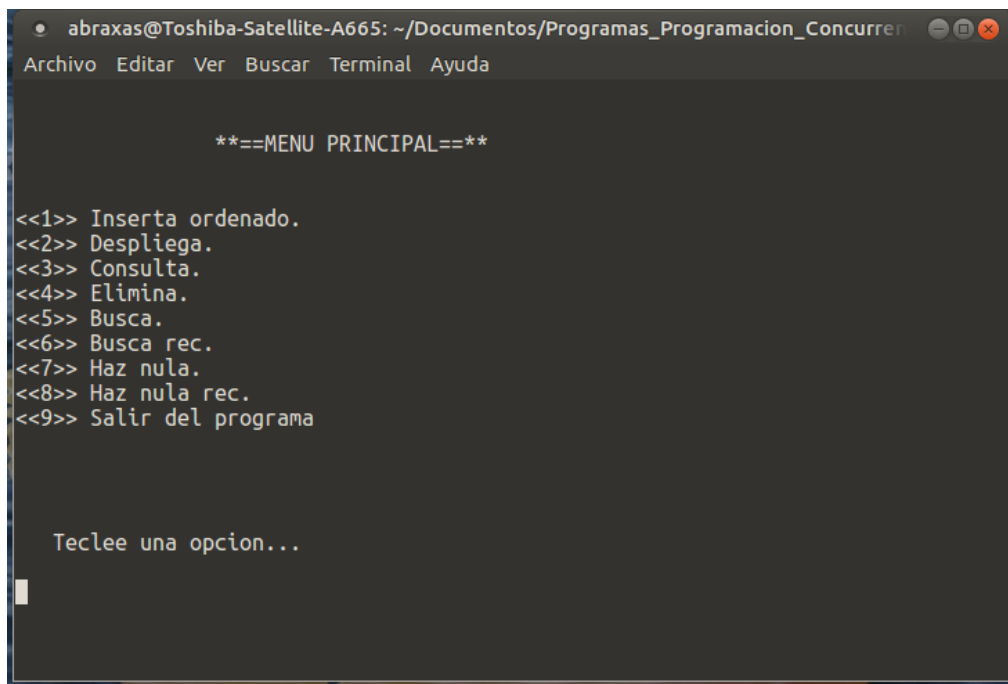
## Listas ligadas en C:

En esta parte de la práctica se pretende desarrollar una lista ligada simple implementada mediante el uso de apuntadores a estructuras (struct) que son muy similares a las clases de tipo nodo que se implementan en Java o Python.

En el video del jueves 27 de febrero del 2025 se dará un breve repaso sobre las estructuras de C y en base a este breve repaso se tendrán que retomar cursos como APALOO, APANLOO y Inteligencia Artificial para poder definir las operaciones que se piden para la elaboración de la lista ligada las cuales se enlistan a continuación;

1. Inserta ordenado.
2. Despliega.
3. Consulta.
4. Elimina.
5. Busca
6. Busca recursivo
7. Haz nula.
8. Haz nula recursivo

Estas operaciones se pueden comprobar en la Figura 5:



```
abraxas@Toshiba-Satellite-A665: ~/Documentos/Programas_Programacion_Concurren
Archivo Editar Ver Buscar Terminal Ayuda

**==MENU PRINCIPAL==**

<<1>> Inserta ordenado.
<<2>> Despliega.
<<3>> Consulta.
<<4>> Elimina.
<<5>> Busca.
<<6>> Busca rec.
<<7>> Haz nula.
<<8>> Haz nula rec.
<<9>> Salir del programa

Teclee una opcion...
```

**Figura 5.** Menú de la segunda parte de la Practica 1.

Se describirá el funcionamiento de cada opción presentada en la Figura 5.

**Inserta ordenado():** En esta opción se tendrá que implementar una forma en la cual el usuario al momento que vaya ingresando enteros (la lista será de tipo entero) estos se vayan ordenando de manera automática. Esto agiliza la parte de la búsqueda que también es parte del menú de opciones. El proceso de inserción de algún elemento se muestra en la Figura 6.

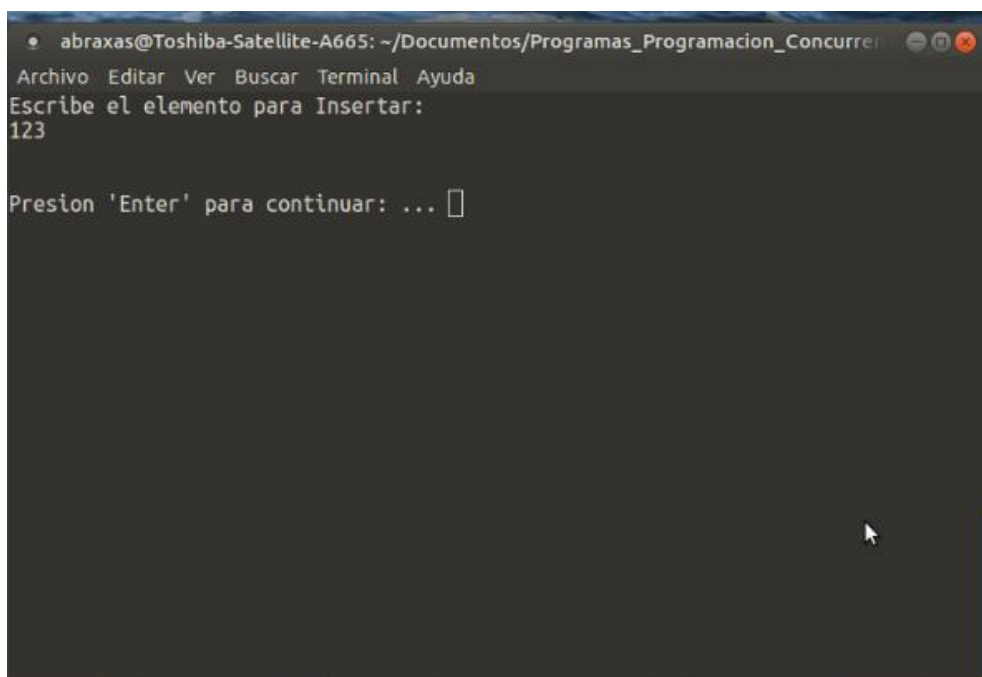


```
abraxas@Toshiba-Satellite-A665: ~/Documentos/Programas_Programacion_Concurren
Archivo Editar Ver Buscar Terminal Ayuda

***==MENU PRINCIPAL==***

<<1>> Inserta ordenado.
<<2>> Despliega.
<<3>> Consulta.
<<4>> Elimina.
<<5>> Busca.
<<6>> Busca rec.
<<7>> Haz nula.
<<8>> Haz nula rec.
<<9>> Salir del programa

Teclee una opcion...
1
```

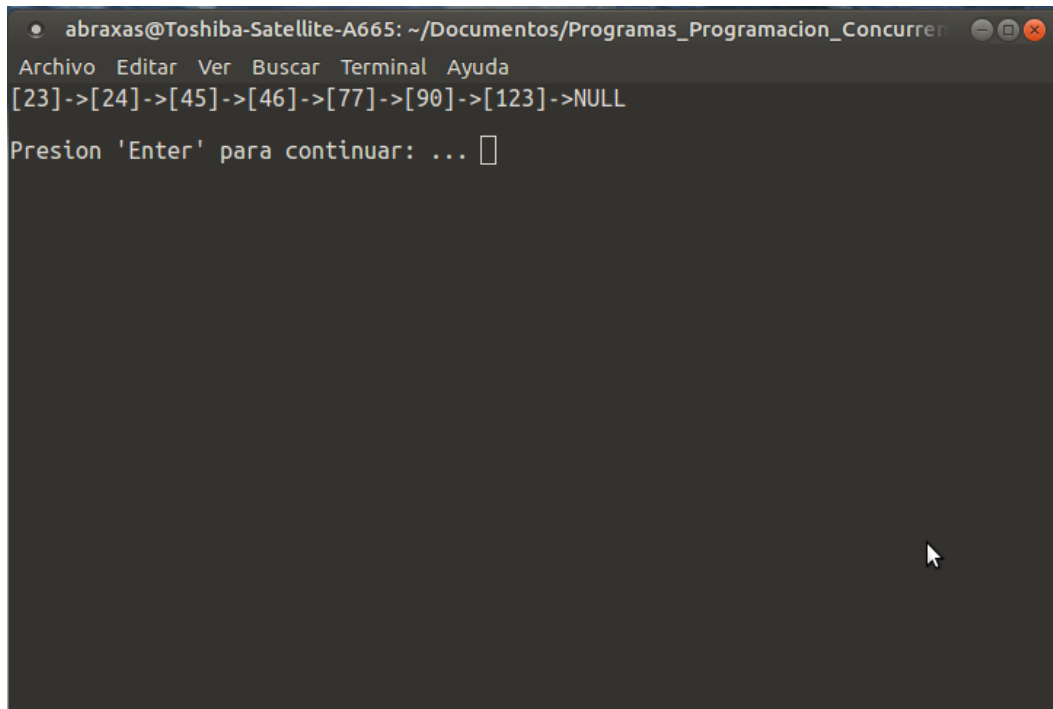


```
abraxas@Toshiba-Satellite-A665: ~/Documentos/Programas_Programacion_Concurren
Archivo Editar Ver Buscar Terminal Ayuda
Escribe el elemento para Insertar:
123

Presion 'Enter' para continuar: ...
```

**Figura 6.** Proceso de inserción ordenada de elementos

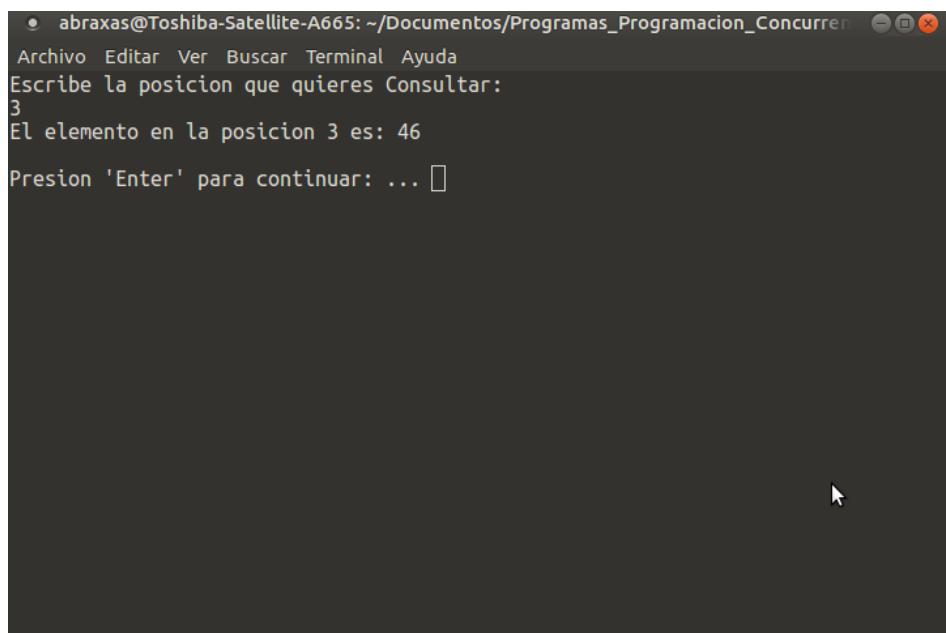
**Despliega():** En esta opción se tiene que mostrar cada uno de los elementos de la lista ligada, tomar de apoyo el despliegue que se muestra en la Figura 7.



```
abraxas@Toshiba-Satellite-A665: ~/Documentos/Programas_Programacion_Concurren
Archivo Editar Ver Buscar Terminal Ayuda
[23]->[24]->[45]->[46]->[77]->[90]->[123]->NULL
Presion 'Enter' para continuar: ...
```

**Figura 7.** Despliegue de los elementos de la lista

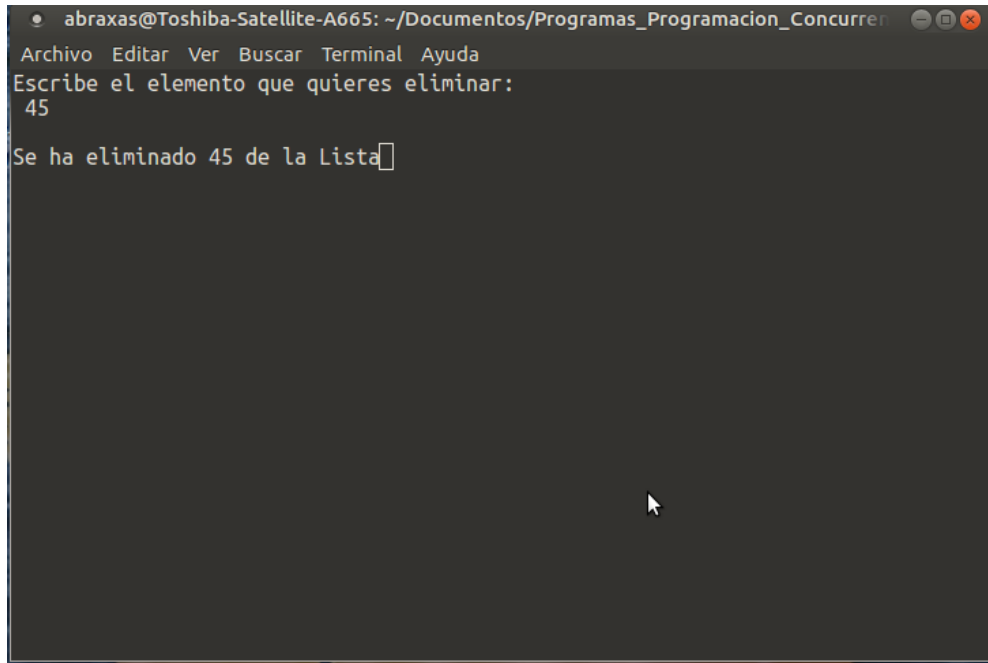
**Consulta():** En esta opción se le solicita la posición de la lista que quiere consultar, en el caso de la implementación que se muestra en este documento las posiciones iniciales es la 0 y la final es las n-1, esto se puede corroborar en la Figura 8.



```
abraxas@Toshiba-Satellite-A665: ~/Documentos/Programas_Programacion_Concurren
Archivo Editar Ver Buscar Terminal Ayuda
Escribe la posicion que quieres Consultar:
3
El elemento en la posicion 3 es: 46
Presion 'Enter' para continuar: ...
```

**Figura 8.** Ejecución de la opción consultar.

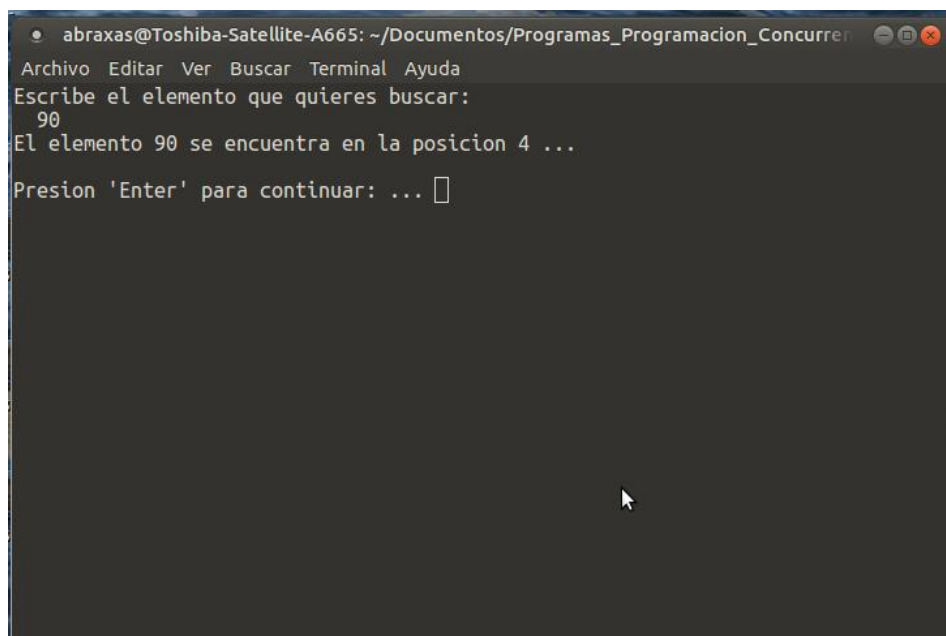
**Elimina():** Esta opción va a permitir eliminar un elemento de la lista, por lo que se debe pedir al usuario que proporcione un valor que este en la lista, caso contrario se enviara un mensaje de error reportando que el elemento no existe, esto se puede ver la Figura 9.



```
abraxas@Toshiba-Satellite-A665: ~/Documentos/Programas_Programacion_Concurren
Archivo Editar Ver Buscar Terminal Ayuda
Escribe el elemento que quieres eliminar:
45
Se ha eliminado 45 de la Lista
```

**Figura 9.** Ejecución de la opción eliminar


**Busca():** Esta opción permite buscar un elemento dentro de la lista y regresa la posición en donde se encuentra el elemento en cuestión, la ejecución de esta opción se muestra en la Figura 10.



```
abraxas@Toshiba-Satellite-A665: ~/Documentos/Programas_Programacion_Concurren
Archivo Editar Ver Buscar Terminal Ayuda
Escribe el elemento que quieres buscar:
90
El elemento 90 se encuentra en la posicion 4 ...
Presion 'Enter' para continuar: ...
```

**Figura 10.** Funcionamiento de la opción Consulta()

**Busca rec():** Esta opción va tener un funcionamiento similar a Consulta(), la diferencia radica que en este caso se trata de función recursiva, tanto esta como la función Consulta() se puede tomar de apoyo el materia teórico que estará en la página referente a los métodos de ordenamiento y búsqueda, la ejecución de este operación se muestra en la Figura 11.



```
abraxas@Toshiba-Satellite-A665: ~/Documentos/Programas_Programacion_Concurren
Archivo Editar Ver Buscar Terminal Ayuda
Escribe el elemento que quieres buscar:
77
El elemento 77 se encuentra en la posicion 3 ...
Presion 'Enter' para continuar: ...
```

**Figura 11.** Ejecución de la operación Busca rec().

**Haz nula()** y **Haz nula rec():** Tienen el mismo funcionamiento que en este caso es vaciar o liberar el espacio de memoria asignado en la formación de una lista, la diferencia es que una función es recursiva y la otra iterativa.

En este caso es importante saber que en C a diferencia de Java o Python no existe un recolector de basura, por lo que solo asignar la lista en cuestión a null, no significa que los demás elementos se borren.

En este caso se necesita liberar la memoria de cada elemento de la lista utilizando la instrucción free(nodo); de lo contrario no se considerará un método valido al momento de revisar la práctica. Las ejecuciones de estas operaciones se muestran en la Figura 12.

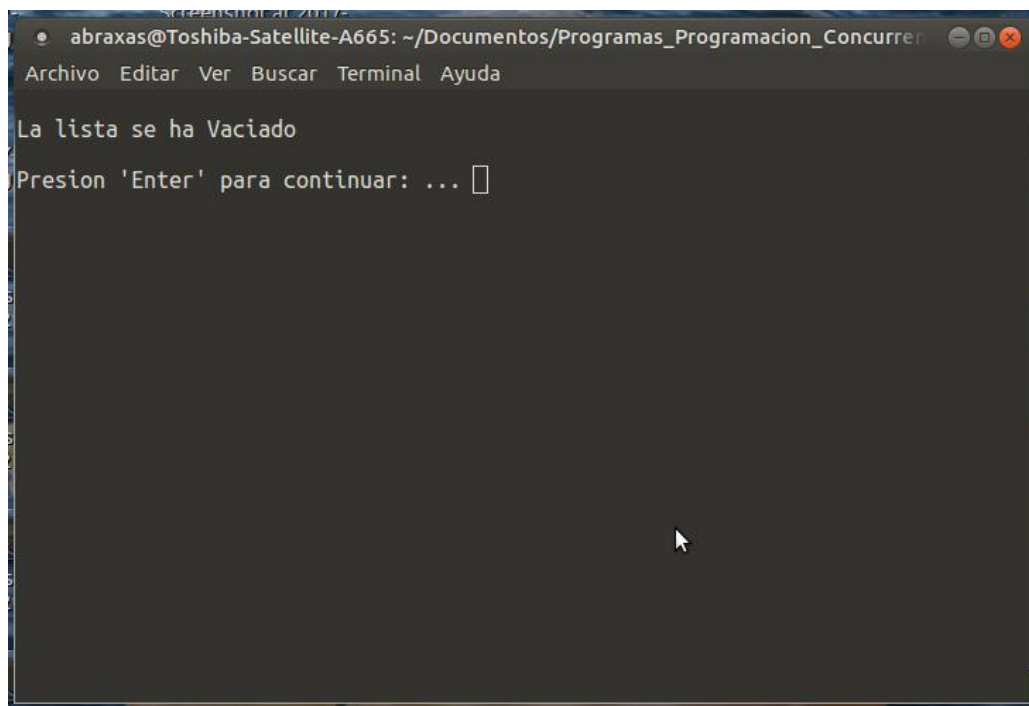


Figura 12. Ejecución de las operaciones Haz nula() y Haz nula rec()

Salir del programa(): La opción salir del programa en efecto permite salir y finalizar el programa.

Es muy importante recomendar que se haga el esfuerzo por comprender el funcionamiento de este par de ejemplos que no representan algo nuevo o que nunca se haya revisado en cursos pasados. De esta manera se podrá obtener una mayor retroalimentación de esta y todas las practicas que se soliciten en lo sucesivo del curso.