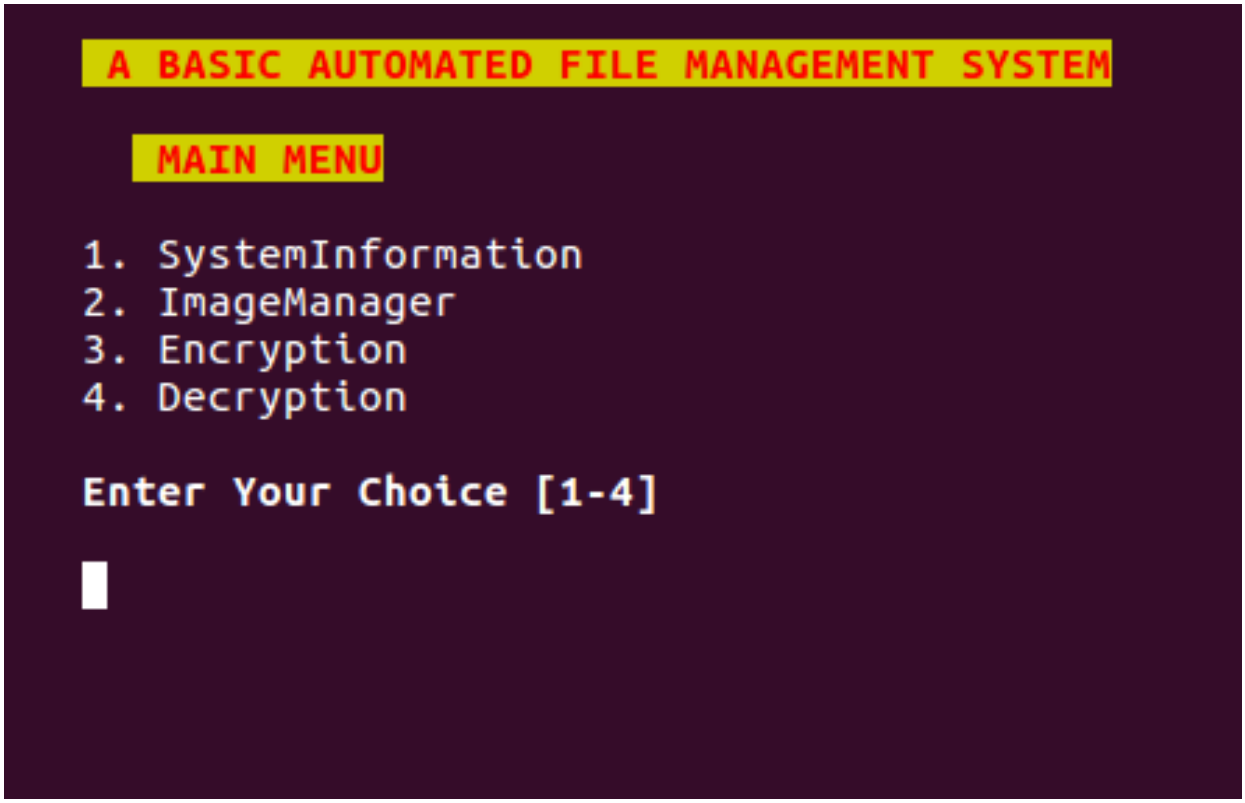
RESULTS AND ANALYSIS

1.) The execution code start within an user-interactive interface asking for input from the user for what to proceed with.



2.) Prompts the user enter keys for specified actions.Suppose if user enters **1** then he enters in a non interactive interface dispalying availaible information from the system.

```
                    HOSTNAME INFORMATION

   Static hostname: airgaz-Inspiron-5555
         Icon name: computer-laptop
           Chassis: laptop
        Machine ID: 4e48fe8c52494c8fa9265a3ae1883d77
           Boot ID: 32d1dfd1d1c742c5853cbb5808729762
  Operating System: Ubuntu 16.04.3 LTS
            Kernel: Linux 4.10.0-40-generic
      Architecture: x86-64

                  FILE SYSTEM DISK SPACE USAGE

Filesystem      Size  Used Avail Use% Mounted on
udev            3.4G     0  3.4G   0% /dev
tmpfs           699M   27M  673M   4% /run
/dev/sda5        15G  8.7G  5.1G  64% /
tmpfs           3.5G   36M  3.4G   2% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
tmpfs           3.5G     0  3.5G   0% /sys/fs/cgroup
tmpfs           699M  100K  699M   1% /run/user/1000

               FREE AND USED MEMORY

              total        used        free      shared  buff/cache   available
Mem:        7150304     2892488     2305204      112656     1952612     3823052
Swap:             0           0           0

               SYSTEM UPTIME AND LOAD

 00:10:16 up 1 day, 11:34,  1 user,  load average: 0.79, 0.64, 0.57

                  CURRENTLY LOGGED-IN USERS

airgaz   tty7         2017-11-23 18:46 (:0)

               TOP 5 MEMORY-CONSUMING processes

%MEM %CPU COMMAND
 5.3  3.8 soffice.bin
 5.2  4.9 firefox
 5.1  2.2 Web Content
 3.6  0.8 Web Content
 3.6  5.4 Web Content

               CHECKING FILE SYSTEM USAGE

The remaining available space in /dev/sda5 is critically low. Used: 64%
The remaining available space in total is critically low. Used: 35%

Done.

Do You Wish To Continue ? (1.(Yes) 2.(No)
```

This interface provides an complete information about system.

Well when one of the inputs is entered for “yes” or “no” then the execution returns to the main menu ,Providing the user for further continuation.

3.) when user enters the other option specified by there key. Option 2 specifes for image manager,this takes the user to an interactive interface.

```
Welcome To ImageManager

Enter The Name Of The Source Folder
imagesource

Enter The Name Of The Destination Folder
picture

Initiating Process
```

It asks the user to enter the source from where images are to be taken and sorted on the basis of their data such as date .Then it asks for the destination folder where the data are to be stored.

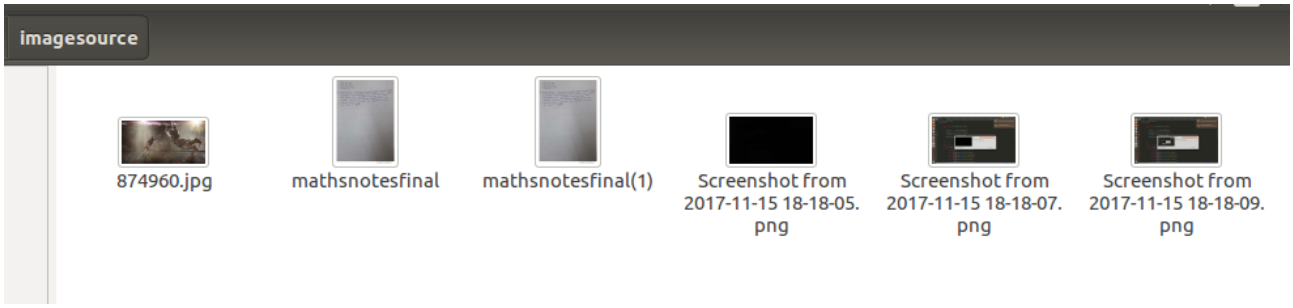
It initiates the process of image managing and takes the user to non-interactive mode where it updates the user with the processes happening in the background.

```
Welcome To ImageManager

Enter The Name Of The Source Folder
imagesource

Enter The Name Of The Destination Folder
Pictures

Initiating Process

Welcome To ImageManager
Image[/home/airgaz/imagesource/Screenshot from 2017-11-15 18-18-09.png] processed
Image[/home/airgaz/imagesource/Screenshot from 2017-11-15 18-18-07.png] processed
Image[/home/airgaz/imagesource/Screenshot from 2017-11-15 18-18-05.png] processed
Image[/home/airgaz/imagesource/874960.jpg] processed
Mission Succesfully finished :-)

Do You Want To Remove The Left Empty Folder ? [ 1.(Yes) or 2.(No) ]
```

Asks the user wheather to remove the left empty folder or not.

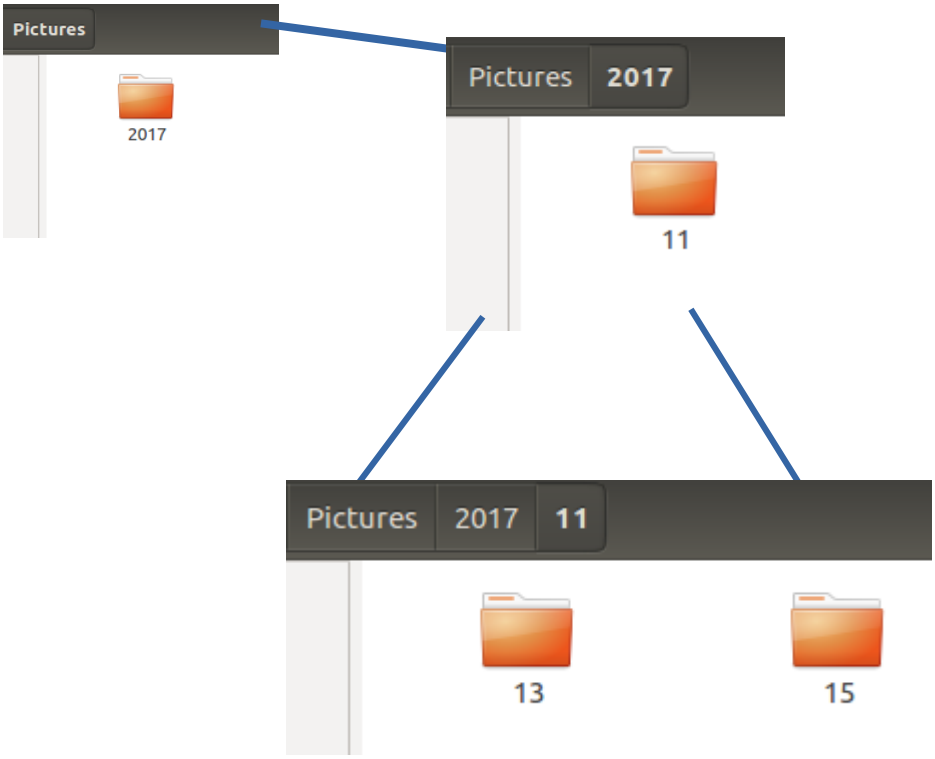
Then after completion of the process we will have a glance at the **source folder** and the **destination folder** .

SOURCE FOLDER : $HOME/airgaz/imagesource is the path of the source folder,before manipulation it had the input images to be given,and looked like somewhat this.



after the completion of the execution the source folder files are moved to the destination folder.

DESTINATION FOLDER : $HOME/airgaz/Pictures is the path to the destination folder, hence we get all the files sorted in different folders according to there data in the image in the suitable folder.



Once the execution gets completed the code returns to the main menu and asks the user for further inputs.

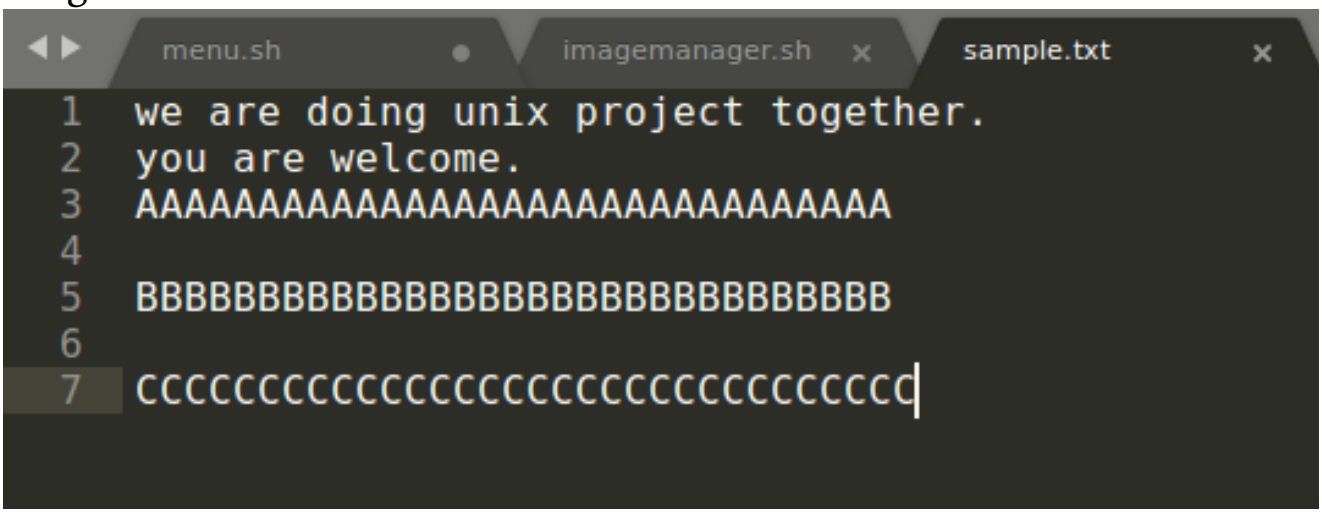
4.) when the user enters the **3 rd** option the code returns a menu asking for **directories** where the file that is to be **encrypted** is present.Here if the user enetrs the wrong **Directory code doesnt terminates** and runs until the user enetrs the corect directory ,hence this way an exception is handled in the code.Here when the users enters the correct filename along with extensions its asks for passpharase from the user for securing filename **sample.txt** after that,its asks for the final destination where the encrypted data is to be stored and removes the original file.

```
Enter a Valid Directory Name. . . . .

******************** Welcome To The File Encrypter ********************

Enter The Directory Where The File Is Present :
unixproject/decrypted

Enter the Exact File Name with extension
sample.txt

encrypting file. . . .

Enter passphrase:
```

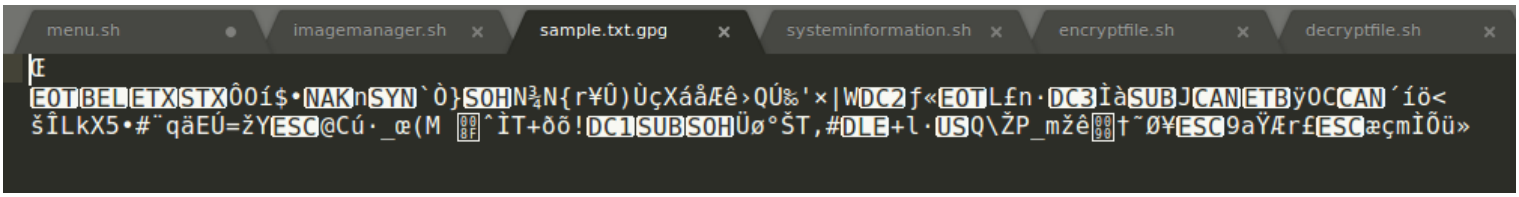
```
Enter a Valid Directory Name. . . . .

******************** Welcome To The File Encrypter ********************

Enter The Directory Where The File Is Present :
unixproject/decrypted

Enter the Exact File Name with extension
sample.txt

encrypting file. . . . . .


File Encrypted Succesfully

Now removing the original file. . . . .

Enter the Location where File Is To be Stored
decrypted

Moving The Encrypted File To decrypted Folder . . . .

***************** Task Completed *****************

Do You Wish To Continue ? (1.(Yes) 2.(No)
```

Once after the complete execution of the program we will check the result wheather the file has been encrypted or not.

Sample.txt : this was the sample file which we encrypted,hence the original data file is.....



we have the encrypted file named as
Sample.txt.gpg : this is the encrypted file which cant be accesed without the passphrase.



hence, the encrypted file is stored in the folder specified by the user, hence the essence of encryption is achieved.

After the execution of this program segment , the program returns to the main menu,hence the user selects wheather to go further r to stop there.

Once it goes to the main menu ,user enters the choice from the main menu.

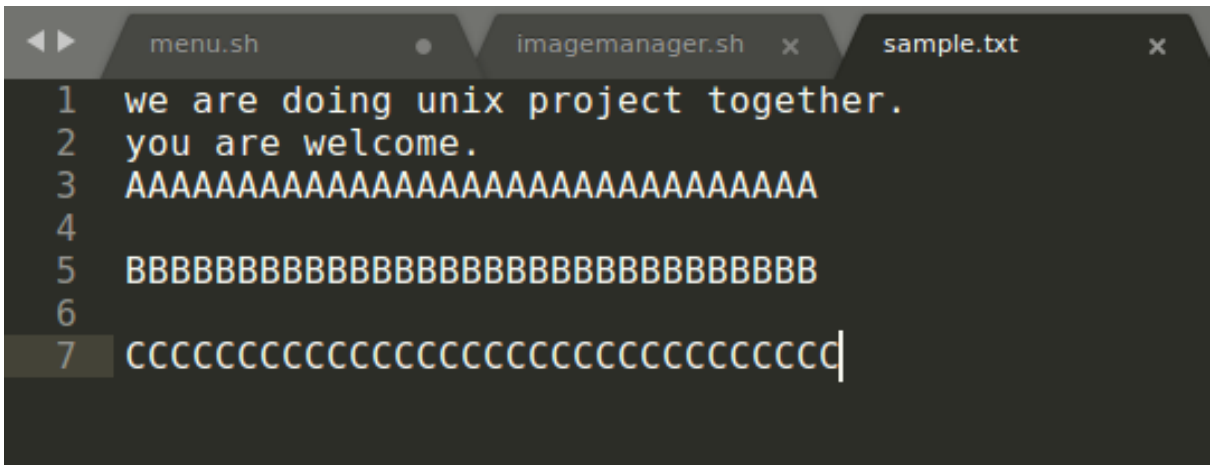
5.) This choice takes the user to the code segment where the encrypted file can be decrypted back once the user has acces to the password .here an interface appears in fron of the user and asks for the source file path and if a valid path is entered it moves further asking for the file to be decrypted.....

```
******************** Welcome To The File Decrypter ********************

Enter The Directory Where The File Is Present :
encrypted

Enter the Exact File Name with extension( example; filename.extension.gpg )
sample.txt.gpg

Decrypting file. . . . . .

Enter the Name For The New File
```

it asks the user to enter the name of the new file to be created for the decryption data to be stored.

Once execution finalizes it asks for the **passphrase** which is used when encrypting the file.

Hence the decrypted file looks like the normal file accesible by the user.



Once the execution finishes the code segment again returns to the main menu and asks for further inputs.