Assignment 3

Hash table

Name: Ahmed Mahmoud Ismail

ID: 4572

Assignment Hash table

Introduction:

In this assignment you are required to implement a Hashing based retrieval system for digits images. Given an image as an array of integers you will compute a hash code for every image.
You will use the computed hash codes to populate a hash table. For retrieval purposes you will be given an array that represents an image.
Then you will be asked to
1. Know its label if it was previously inserted.
2. Add it to the hash table
3. Remove it from the hash table.

The Algorithm of the project:-

I. In the start of the project it reads the file of 1000 images and store it in the array of size 1000 and every images have 783 pixels and key.

II. It will give you 2 option to run the program as linear probing or separate chaining and put in the hash table by one of the two methods.

III. If you choose separate chaining and the program will ask you to the name of the file you want to search or delete from it and will read its data and store it in array of images then will put in the hash table the 1000 images using separate chain by making every node a pointer to linked list and if the node is null will initialize it and add tail with the id of the image and then you will get the menu of separate chaining that contains search w delete and exit the project.

IV. search will call the get chain which will return the id of the image by calculate the hash code of the image and go to the index in the hash table and search in them and compare each pixel till you find it and return its ID.

V. And delete will call remove chain which make a pointer to a head and calculate the hash code and find the image like search and if he finds it he has 3 option if node you want to remove is the head will make head and tail equal null and free it and if node is tail will make another pointer will loop from head till the node before target and make the tail is the node it and free the node and if between will loop till the node before it and make the next of the node before it the next of the node and free it.

VI. And if you used the program as linear probing will call the propping which put the 1000 images in the hash table using linear probing method its array of integer if its equal null or AVAILABLE will enter the data in it but if it's not empty will increase the hash code of image until it finds an empty place and if its equal to 1000 will return again and check for empty place from zero.

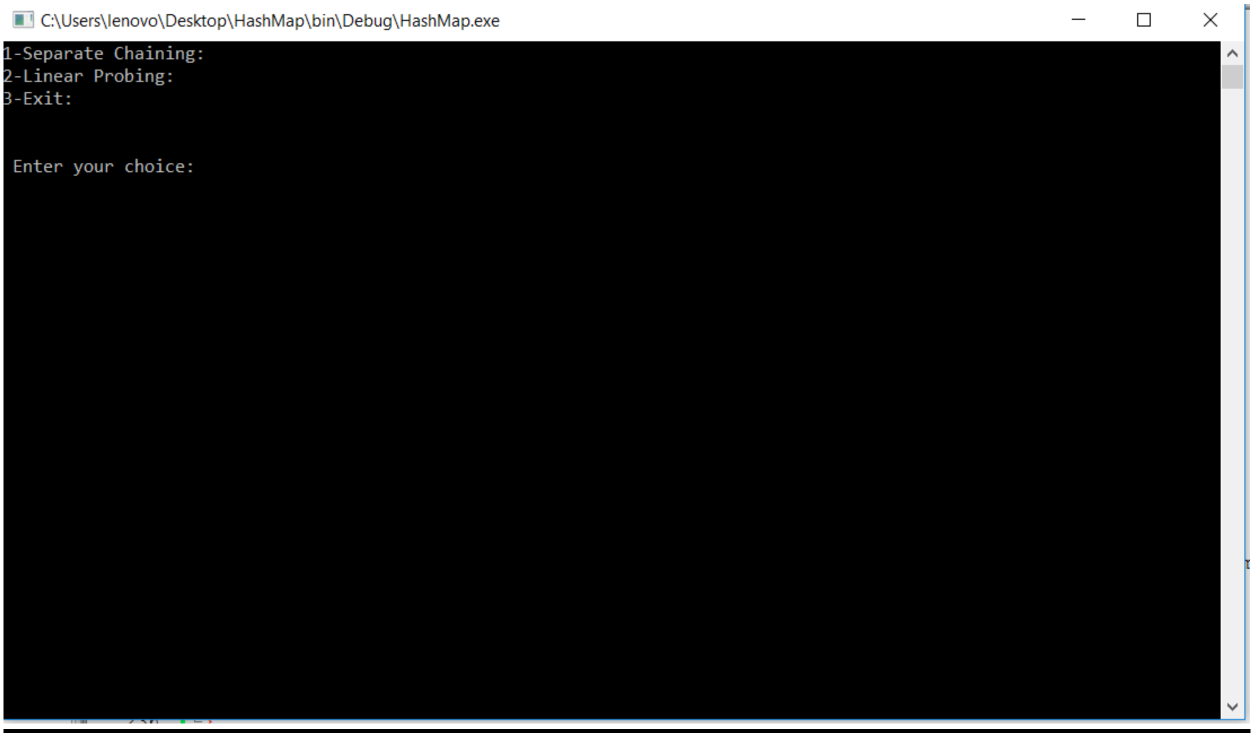
VII. And if you choose search will calculate the hash code of the image and get the ID of the hash table with the same index and if its AVAILABLE means the image is removed and if hash code of id from the hash table equal to the hash code of the image will check the pixels if equal will return the ID if not will increase the hash code start number till he finds it and if it reaches 1000 will begin again from zero to continue checking.

VIII. And delete will find it in the same way but if he finds it will make the data in it equal AVAILABLE means it's removed and can put in it after that.

IX. And computing hash code will loop from the image to 783 and will sum the number in it and if its modulus 28 means it ends the row so we will take the sum and multiply it in row number which it start from 1 and then increase the hash code which is the sum of all columns multiply its rows and then reset these sum and increase the number of rows.

Screenshots of the output screen:

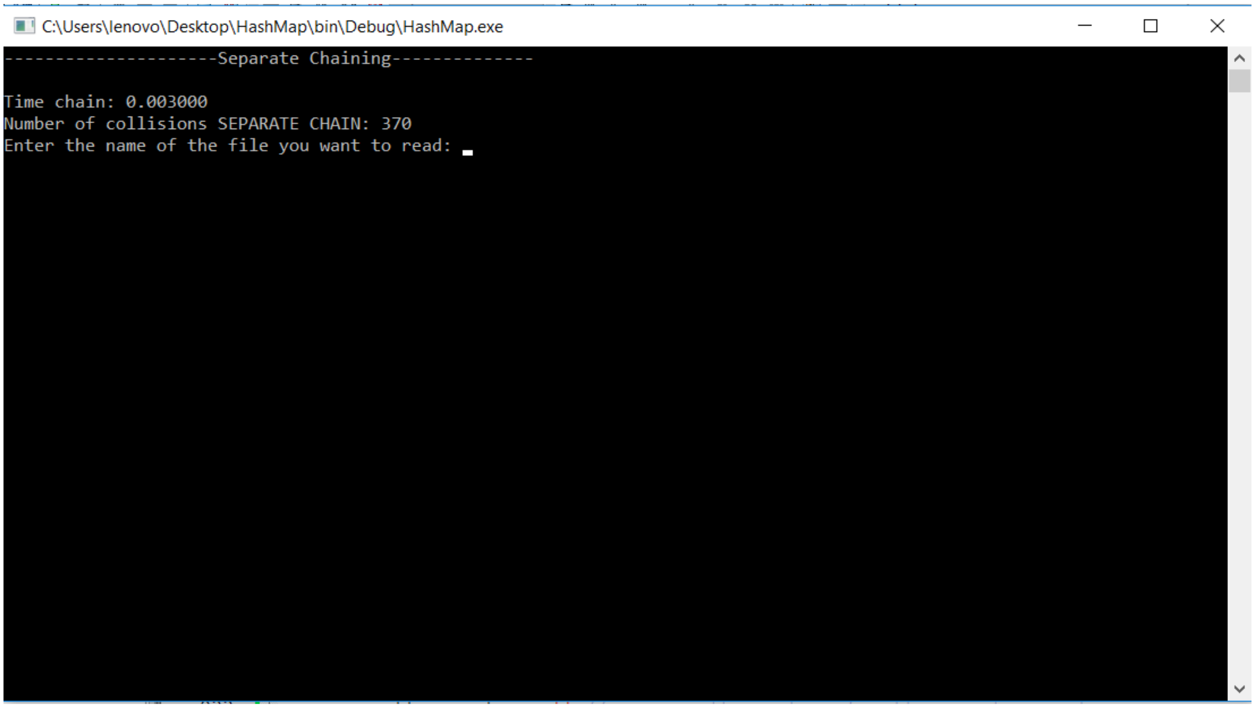
1-menu



```
1-Separate Chaining:
2-Linear Probing:
3-Exit:


 Enter your choice:
```

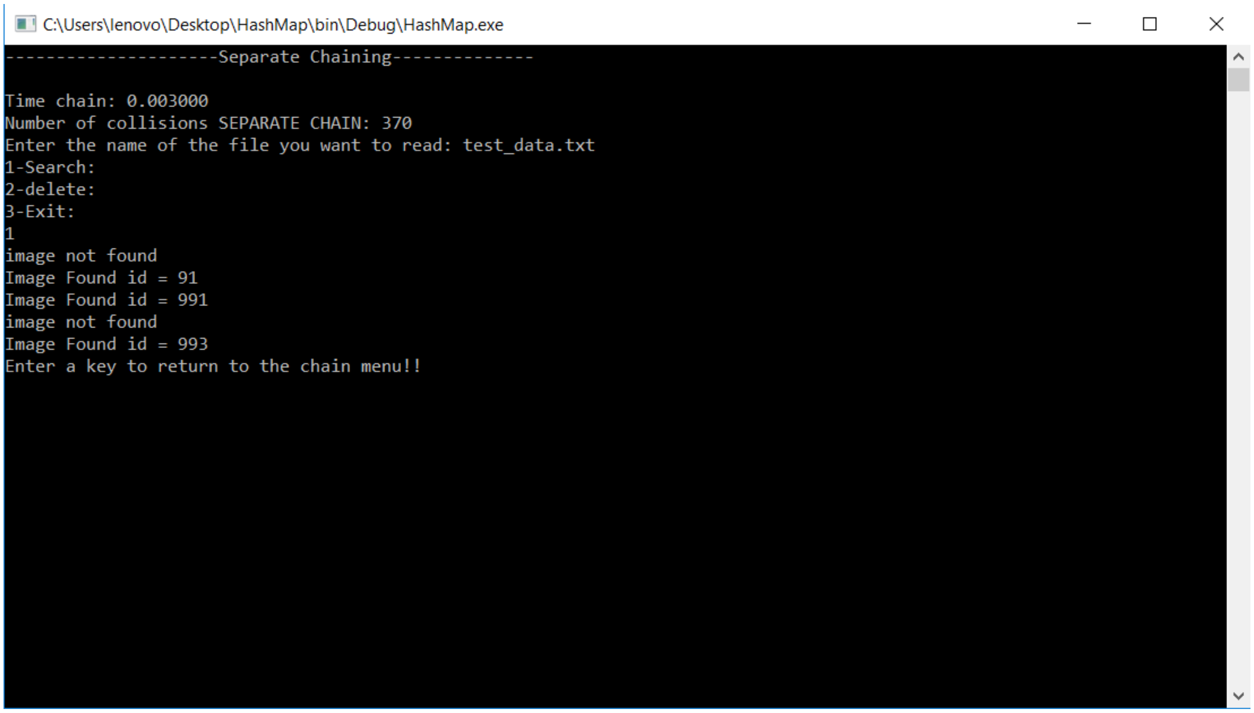
2-Separate chain number of collision and time



```
--------------------Separate Chaining-------------

Time chain: 0.003000
Number of collisions SEPARATE CHAIN: 370
Enter the name of the file you want to read:
```

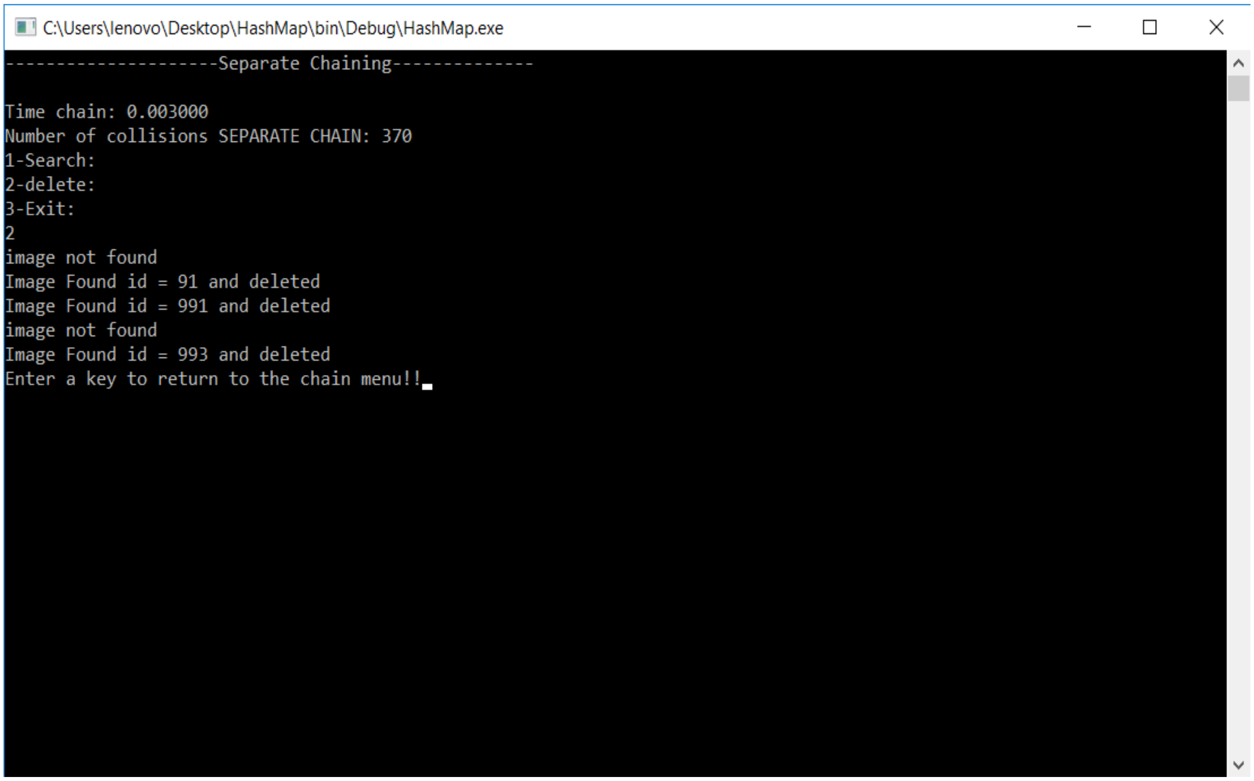
3- Separate chain search



```
--------------------Separate Chaining-------------

Time chain: 0.003000
Number of collisions SEPARATE CHAIN: 370
Enter the name of the file you want to read: test_data.txt
1-Search:
2-delete:
3-Exit:
1
image not found
Image Found id = 91
Image Found id = 991
image not found
Image Found id = 993
Enter a key to return to the chain menu!!
```

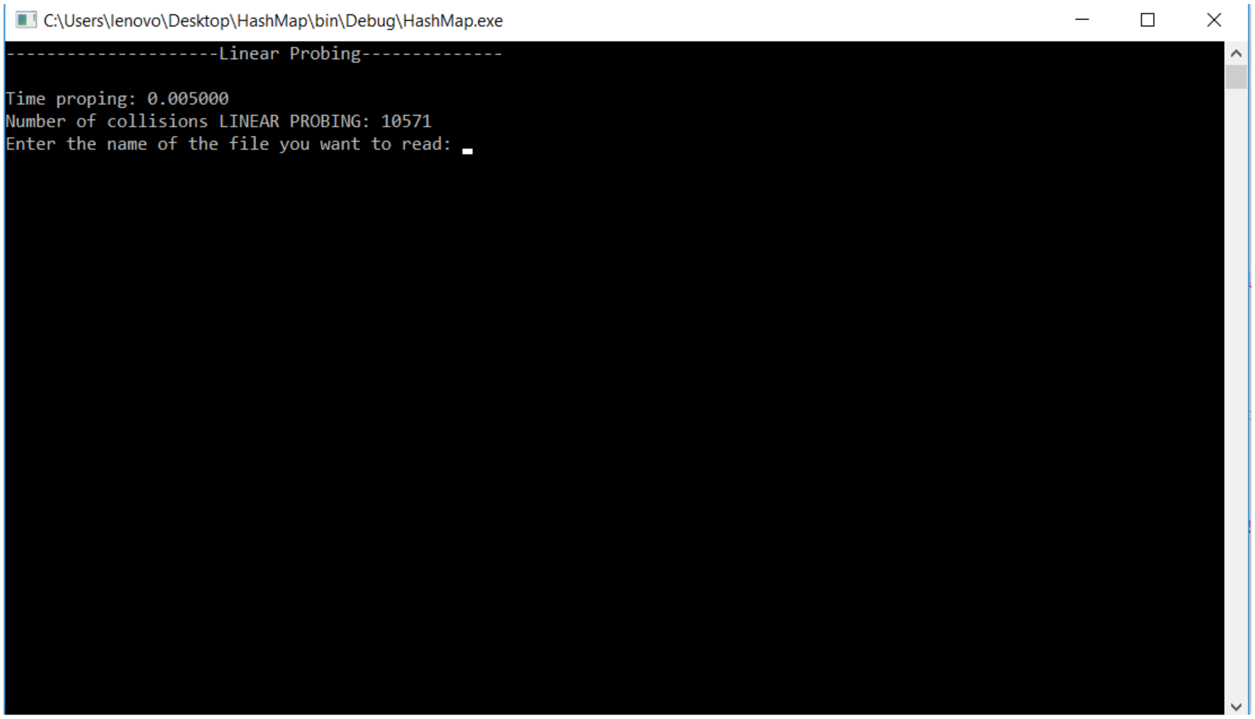
4-separate chain delete



```
--------------------Separate Chaining-------------

Time chain: 0.003000
Number of collisions SEPARATE CHAIN: 370
1-Search:
2-delete:
3-Exit:
2
image not found
Image Found id = 91 and deleted
Image Found id = 991 and deleted
image not found
Image Found id = 993 and deleted
Enter a key to return to the chain menu!!
```

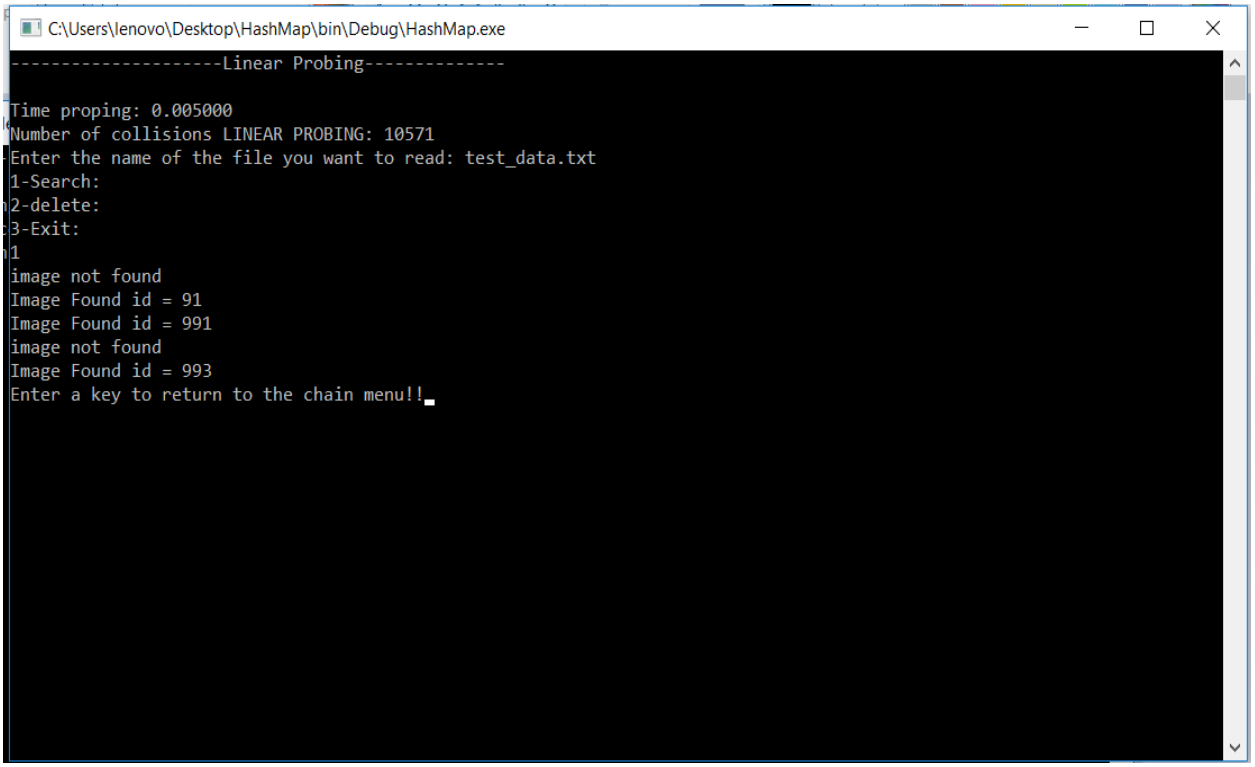
5-linear menu and number of collision and time



```
--------------------Linear Probing-------------

Time proping: 0.005000
Number of collisions LINEAR PROBING: 10571
Enter the name of the file you want to read:
```

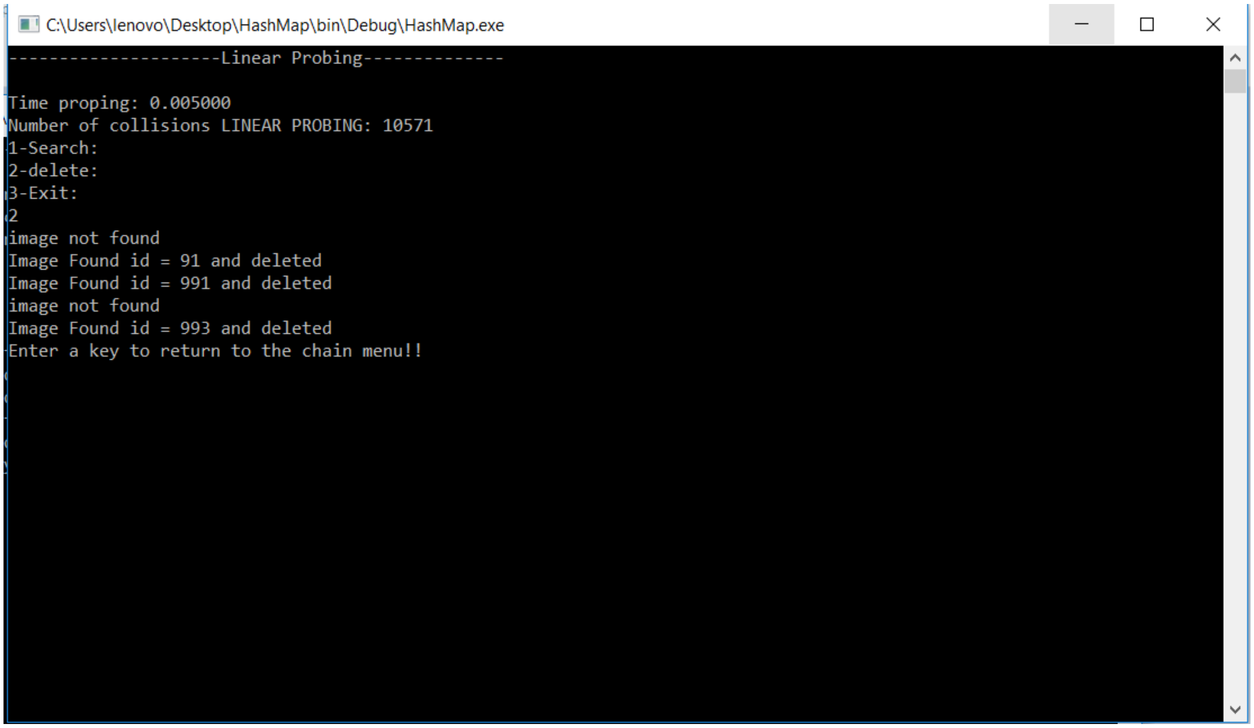
5-linear search



```
--------------------Linear Probing-------------

Time proping: 0.005000
Number of collisions LINEAR PROBING: 10571
Enter the name of the file you want to read: test_data.txt
1-Search:
2-delete:
3-Exit:
1
image not found
Image Found id = 91
Image Found id = 991
image not found
Image Found id = 993
Enter a key to return to the chain menu!!
```

5-linear search



```
---------------------Linear Probing-------------

Time proping: 0.005000
Number of collisions LINEAR PROBING: 10571
1-Search:
2-delete:
3-Exit:
2
image not found
Image Found id = 91 and deleted
Image Found id = 991 and deleted
image not found
Image Found id = 993 and deleted
Enter a key to return to the chain menu!!
```