Hashing

Programming Assignment 7 – Hashing
Optional – Individual Work

Due Date : Monday December 2nd , 2019
No Later than 12:45 pm.

Write a C++ program that does the following : Hard code Store

1. Create an char array of size 20. Assign * to each location in the array indicating that the array is empty..

capital only

2. Populate first 15 elements of the array with random characters between A - Z. inclusive. Use the following formula in order to hash and store each character in its proper position/location.

   Generated character % 10;

3. Should a collision occurs , use **Linear probing** to find next available position / location. Use the following probing hashing function.

   (Generated character + 1) % 10 ;

4. Display and introduction message followed by the generated array. The generated array should be displayed in 2 lines. Each line contain 10 characters separated by 3 spaces

5. At the end , the program should display the number of linear probing that took place when inserting an element , searching for an element , or deleting an element.

At the beginning , the program displays a menu on the screen allowing the user to do the following :

A) Display the generated array.

B) Search for a Character (between A – Z) in the array.

C) Insert new Character (between A - Z) in the array. according to th hash

D) Delete a character (between A – Z) from the array.

see if it's there

remove the character and replace it with a star

X) End the program

-linear probing
-hashing

NOTES:

- **Must use the program that is provided in the my Hashing lecture notes. You are allowed to modify the program.**

- **Just one .cpp file.**

- **A Message is displayed when inserting a char in the table.**

- **A Message is displayed when searching for a char and char is not found.**

- **A Message is displayed when deleting a char and char is not found.**

- **Modify the search and delete functions such that the functions uses hashing and rehashing functions when searching and deleting characters form the table. You are not allowed to use sequential search for locating and deleting character from the table. Add appropriate code in order to keep a track of linear probing when inserting characters in the hash table.**

use hashing for search and delete

- **Menu selection is case sensitive and validation on menu selection items must be implemented**

Rules :

1. **This program must be done as an individual. Group work will not be accepted.**

2. Your program **must compile** and run. The program will be tested using the latest version of Codeblocks for windows. No grades will be given for a program that compiles but does not run.
3. **You re not allowed vectors , linked lists. Must only use arrays.**
4. Your program must be properly **documented according the style above . See the website for the sample programming style program.**
5. You must name your program as :

    - **3358_1_LastName_FirstName_Oprional_F2019_PG7.cpp**

    Where LastName is your Last Name and FirstName is your First Name. For example , the file name should look something like :

    3358_1_Gholoom_Husain_Optional_F2019_PG7.cpp (not .cbp)

6. You must upload your programs no later than the indicated time above on the due date. This a individual work and No late assignments will be accepted.

    USE TRACS To upload your program .

7. You must **also** turn in hard copy of your source code no later than the due date / time . Should the hard copy consist of more than one page , then , the hard copy must be **stapled**. if you are unable to turn in a printout during class, you can take the program to the computer science department and hand it to the front desk personal (Comal 211) before the deadline. Make sure that the front office stamps the program. Make sure that include the date and time. Finally ,make sure that they place the program in my mailbox.

   DO NOT slide your program under my office door – It will **NOT** be accepted.

   Also , No late submission hard copy of the assignment will be accepted and a grade of zero will be assigned .

The following points will be deducted if :

- Incorrect file format such as uploading .cbp instead of .cpp , missing electronic copy , missing the hardcopy , compilation errors , using vectors , linked lists , not using the program provided to you in the class, not using the indicated hashing and quadratic probing functions when inserting , or searching a char element in the array or deleting a char element from the array. Not using class in implementing this assignment , Done as group work. **(- 10 points)**

- Each logical error : **(- 1.25 points)**

- Other : **(2 points each)** if **any** of the following takes a place :

    - Unable to read the source code due to unclear printing
    - Incorrect program file name.
    - Hard copy is not stapled or printed in landscape orientation.
    - Incorrect Style **such as but not limited to** missing Header , missing footer , missing comments or program documentations , missing roster number , missing section number , missing signature line … etc

Style Guidelines:

At the beginning of your program (and **before** the #include statement), include the following :

Header comments (file documentation block) should be at the top of each file and should contain: Author / s, Due Date, Assignment Number, Course number and section, Instructor, and a brief description of the purpose of the code in the file. For example :

```
//        Serial  Number   :        xxxxxxxxx
//
//        Author  : (Your name  here!!)
//        Due Date  :
//        Programming Assignment Number  7- Optional
//
//        Fall   2019   -   CS 3358   -   Your Section  Number
//
//        Instructor:    Husain Gholoom.
//
//         <Brief description of the  purpose of the program>
```

Variable names :

- Must be meaningful.
- The initial letter should be lowercase, following words should be capitalized, no other caps or punctuation (i.e. `weightInPounds`).
- Each variable must be declared on a separate line with a descriptive comment.

Named constants :

- Use for most numeric literals.
- All capitals with underscores (i.e. `TX_STATE_SALES_TAX`)
- Should occur at top of function, or global (only if necessary)

Line length of source code should be no longer than 80 characters (no wrapping of lines).

Indentation :

- Use 2-4 spaces (but be consistent throughout your program).
- Indent blocks, within blocks, etc.
- Use blank lines to separate sections.

Comments for variables :

All variable definitions should be commented as follows:

```
int  gender;    // integer value for the gender,
                // 1
```

Sample Run

```
Welcome to my Hashing Program

------------------------------

A.  Creates an char array of size 30. Assigning - to each
    location in the array indicating that the array is empty.
B.  Populates 10 elements of the array with random characters
    between A and Z inclusive.
C.  If a collision occurs, linear probing will find the next
    available position / location.
D.  The generated array will be displayed in 2 lines.
    Each line contains 15 characters separated by 2 spaces.

 The Generated Array.
.          .          .          .
.          .          .          .
```

Header

```
Choose from the following menu options

A.  Display the generated array.
B.  Search for a char (between A - Z) in the array.
C.  Insert a new char (between A – Z ) in the array.
D.  Delete a char (between A – Z ) from the array.
X.  End the program.
```

menu

```
Choose your option: B

Enter a char you want to search (between A – Z ): X

The char  X was found in location   0
```

```
Choose from the following menu options

A.  Display the generated array.
B.  Search for a char (between A - Z) in the array.
C.  Insert a new char (between A – Z ) in the array.
D.  Delete a char (between A – Z ) from the array.
X.  End the program.
```

while (Userchoice =X)

```
Choose your option: B

Enter a char you want to search (between A – Z ): B

The char  B  was not found in the array
```

search is a bool message

```
Choose from the following menu options

A.  Display the generated array.
B.  Search for a char (between A - Z) in the array.
C.  Insert a new char (between A – Z ) in the array.
D.  Delete a char (between A – Z ) from the array.
X.  End the program.

Choose your option: C

Enter a char you want to insert (between A – Z ):  P

The char P was inserted in location number 5.

Choose from the following menu options

A.  Display the generated array.
B.  Search for a char (between A - Z) in the array.
C.  Insert a new char (between A – Z ) in the array.
D.  Delete a char (between A – Z ) from the array.
X.  End the program.

Choose your option: C
```

```
Enter a char you want to insert (between A – Z ):  P

The char P was inserted in location number 8.


Choose from the following menu options

A.  Display the generated array.
B.  Search for a char (between A - Z) in the array.
C.  Insert a new char (between A – Z ) in the array.
D.  Delete a char (between A – Z ) from the array.
X.  End the program.

Choose your option: A

Displaying the generated array.
.     .     .     .
.     .     .     .




Choose from the following menu options

A.  Display the generated array.
B.  Search for a char (between A - Z) in the array.
C.  Insert a new char (between A – Z ) in the array.
D.  Delete a char (between A – Z ) from the array.
X.  End the program.

Choose your option: D


Enter a char you want to delete (between A - Z): L

L  is deleted  from the array
```

```
Choose from the following menu options

A.  Display the generated array.
B.  Search for a char (between A - Z) in the array.
C.  Insert a new char (between A – Z ) in the array.
D.  Delete a char (between A – Z ) from the array.
X.  End the program.

Choose your option: D


Enter a char you want to delete (between A - Z): L

L   not found in the array


Choose from the following menu options

A.  Display the generated array.
B.  Search for a char (between A - Z) in the array.
C.  Insert a new char (between A – Z ) in the array.
D.  Delete a char (between A – Z ) from the array.
X.  End the program.


Choose your option: P

*** Invalid Option ***
```

```
Choose from the following menu options

A.  Display the generated array.
B.  Search for a char (between A - Z) in the array.
C.  Insert a new char (between A – Z ) in the array.
D.  Delete a char (between A – Z ) from the array.
X.  End the program.


Choose your option: x


*** Invalid Option ***


Choose from the following menu options

A.  Display the generated array.
B.  Search for a char (between A - Z) in the array.
C.  Insert a new char (between A – Z ) in the array.
D.  Delete a char (between A – Z ) from the array.
X.  End the program.


Choose your option: 1

*** Invalid Option ***
```

```
Choose from the following menu options

A.  Display the generated array.
B.  Search for a char (between A - Z) in the array.
C.  Insert a new char (between A – Z ) in the array.
D.  Delete a char (between A – Z ) from the array.
X.  End the program.



Choose your option: C

Enter a char you want to insert (between A – Z ):  P

Array is Full.



Choose from the following menu options

A.  Display the generated array.
B.  Search for a char (between A - Z) in the array.
C.  Insert a new char (between A – Z ) in the array.
D.  Delete a char (between A – Z ) from the array.
X.  End the program.

Choose your option: X


The number of linear probes when each number is hashed and collision occurred  when adding a
character in the array is 5

The number of linear probes when each number is hashed and collision occurred  when searching for a
character in the array is 2

The number of linear probes when each number is hashed and collision occurred  when deleting a
character from the array is 35


This hashing program was implemented  by
Husain Gholoom – 12 – 2 - 2019
```