# Programming Assignment 7 – Hashing

## Optional - It will Replace One Your Lowest Programming Assignment Grades

## Due Date : Monday November 30th , 2020 No Later than 11:15 am.

Write a C++ program that does the following :

1. Create an char array of size 30. Assign - to each location in the array indicating that the array is empty..

2. Populate first 27 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 rehashing function.

   Current hashing position + 1 ;

4. Display and introduction message followed by the generated array. The generated array should be displayed in 2 lines. Each line contain 15 characters separated by 2 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 in the Array.

C) Insert a New Character in the Array.

D) Delete a Character from the Array.

X) End The Program

## 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.**

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

## Rules :

1. Your program **must compile** and run. We will test your program Code::Blocks version 17.12 for windows. No grades will be given for a program that compiles but does not run.
2. **You re not allowed vectors , linked lists. Must only use static arrays.**
3. Your program must be properly **documented according the style above . See the website for the sample programming style program.**
4. You must name your program as :

   - **3358_7_LastName_FirstI_F20_PG7.cpp**

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

   **3358_7_Gholoom_H_F20_PG7.cpp ( not .cbp )**

5. You must upload your programs no later than 11:15 am on the due date. This a individual work and **No late assignments will be accepted and no extensions will be given.**

   **USE Canvas To upload your program .**

## The following points will be deducted if :

- Incorrect file format such as uploading .cbp instead of .cpp , missing electronic copy , compilation errors , using vectors , linked lists , not using the program provided to you in the class notes, not using the indicated hashing and linear 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 . **( - 10 points )**

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

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

    - Incorrect program file name. Incorrect Style **such as but not limited to** missing Header , Missing Comments or Program Documentations , 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 :

```
//      Author  : (Your name  here!!)
//
//      Due Date  :
//
//      Programming Assignment Number  7- Optional
//
//      Fall   2020   -   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 27 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.
.     .     .     .
.     .     .     .



Choose from the following menu options

A.  Display the generated array.
B.  Search for a char  in the array.
C.  Insert a new char  in the array.
D.  Delete a char from the array.
X.  End the program.

Choose Your Selection  : B

Enter a char you want to search  :     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  in the array.
C.  Insert a new char  in the array.
D.  Delete a char from the array.
X.  End the program.

Choose Your Selection  : B

Enter a char you want to search  :   B

The char  B  was not found in the array


Choose from the following menu options

A.  Display the generated array.
B.  Search for a char  in the array.
C.  Insert a new char  in the array.
D.  Delete a char from the array.
X.  End the program.

Choose Your Selection  : C

Enter a char you want to insert  :  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  in the array.
C.  Insert a new char  in the array.
D.  Delete a char from the array.
X.  End the program.

Choose Your Selection  :  C
```

```
Enter a char you want to insert :  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  in the array.
C.  Insert a new char  in the array.
D.  Delete a char from the array.
X.  End the program.

Choose Your Selection    :   C

Enter a char you want to insert :  X

Array is Full.



Choose from the following menu options

A.  Display the generated array.
B.  Search for a char  in the array.
C.  Insert a new char  in the array.
D.  Delete a char from the array.
X.  End the program.

Choose Your Selection  :  A

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

```
Choose from the following menu options

A.  Display the generated array.
B.  Search for a char  in the array.
C.  Insert a new char  in the array.
D.  Delete a char from the array.
X.  End the program.

Choose Your Selection  :  D


Enter a char you want to delete   : L

L  is deleted  from the array


Choose from the following menu options

A.  Display the generated array.
B.  Search for a char  in the array.
C.  Insert a new char  in the array.
D.  Delete a char from the array.
X.  End the program.

Choose Your Selection  :  D


Enter a char you want to delete  : L

L   not found in the array
```

```
Choose from the following menu options

A.  Display the generated array.
B.  Search for a char  in the array.
C.  Insert a new char  in the array.
D.  Delete a char from the array.
X.  End the program.

Choose Your Selection  :  P

*** Invalid Option ***


Choose from the following menu options

A.  Display the generated array.
B.  Search for a char  in the array.
C.  Insert a new char  in the array.
D.  Delete a char from the array.
X.  End the program.

Choose Your Selection   :  x


*** Invalid Option ***
```

```
Choose from the following menu options

A.  Display the generated array.
B.  Search for a char  in the array.
C.  Insert a new char  in the array.
D.  Delete a char from the array.
X.  End the program.

Choose Your Selection  :  1

*** Invalid Option ***


Choose from the following menu options

A.  Display the generated array.
B.  Search for a char  in the array.
C.  Insert a new char  in the array.
D.  Delete a char from the array.
X.  End the program.

Choose Your Selection  :  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 – 11 – 30 - 2020
```