Computer Engineering Department

TED University

CMPE 252 – C Programming, Spring 2023

Lab 02

In this lab, you are asked to complete **similars.c** program file which has been already given in LMS. In this program, there are four functions, namely, `main`, `hash_text`, strong_similar, and `weak_similar`. `main` function is already provided, and it is supposed to remain as it is (you should not change it). You are required to implement `hash_text`, strong_similar, and `weak_similar` functions.

Here are the operations performed in `main` function:

- An array of strings with name `list with the size 10` is created to hold name and surname data in "name_surname" format and the elements are read into it.
- An array of integers with name `hashed_vals` is created to hold hashed values of name and and surname in unsigned integer format.
- A two-dimensional array of integers to keep the mapping of similarities between each name surname pair.
- hashed_vals array is initialize by calling hash_text function to calculate the hash values of full name and surname pair.
- strong_similar function is called to find the same names and fill the two-dimensional array called same to keep the mapping of similarities.
- The name surname pairs, the calculated hash values and similarity mapping is printed on the standard output.
- weak_similar function is called to calculate the hash values of each name and surname separately, then build the similarity map using hash values.
- The name surname pairs, and weakly similarity mapping is printed on the standard output.
- Total_characters function calculates the total number of characters in each string and prints

Task 1: Implement `hash_text` function.

```
unsigned int hash_text (char * list) ;
```

A character pointer holding names and surnames is sent as an input, and the hash value of the input text is returned. During the calculation a basic formula is used as follows when the name is ali. The ascii code for a is 97, for l is 108 and for i is 105. The function calculates the hash value by the following formula;

hash = 36^2*97+36*108+105

The algorithm should be applied to all symbols including whitespaces between name and surname. At the end mod 1000 of hash value should be returned.

hash = hash mod 1000

Task 2: Implement `strong_similar` function.

```
void strong_similar (int hval[MAX_ELEM], int map[MAX_ELEM][MAX_ELEM]) ;
```

Calculated hash values of all names-surnames pair are sent to the function. The function modifies and returns the map of exact matches. Exact matching of the entries is calculated only by using hashed values.

Task 3: Implement `weak_similar` function.

```
void weak_similar (char list[MAX_ELEM][STR_LEN],int map[MAX_ELEM][MAX_ELEM]) ;
```

Array of names and surnames list is sent as input and previously calculated similarity map is sent as input-output parameter.

During the execution of the function strtok function is used to split names and surnames by using the space between them. To store the splitted names and surnames a three-dimensional array is created. First dimension is same as the list in main function, second dimension is two either to point name string or surname string. Similarly, two-dimensional array is created to keep hash values of names and surnames for each pair.

The function then calculates hash values of names and surnames for each pair by calling hash_text function. Once the hash values of all names and surnames are calculated the function updates the map array to mark the weak similarities (either names or surnames are same). For example;

ali kemal and mustafa kemal are weakly similar. They both are weakly similar to kemal ahmet.

Task 4: Implement **print_string_lengths** function

print_string_lengths, calculates the length of the each string while excluding spaces. Then print out with this format: “ali kemal – 8”

Sample Run:

See the next page for sample run.


```
Enter 10 elements:
ahmet yuksel
mehmet arslan
mustafa kemal
ali kemal
mustafa kemal
mustafa kemal
mehmet arslan
kemal ahmet
ali kaan
kemal kaan

Elements with hash values and similarities:

ahmet yuksel            328
mehmet arslan           250   6
mustafa kemal           104   4   5
ali kemal               544
mustafa kemal           104   2   5
mustafa kemal           104   2   4
mehmet arslan           250   1
kemal ahmet             376
ali kaan                298
kemal kaan              850
```


```
Elements with weak similarities:
ahmet yuksel            328   7
mehmet arslan           250
mustafa kemal           104   3   7   9
ali kemal               544   2   4   5   7   8   9
mustafa kemal           104   3   7   9
mustafa kemal           104   3   7   9
mehmet arslan           250
kemal ahmet             376   0   2   3   4   5   9
ali kaan                298   3   9
kemal kaan              850   2   3   4   5   7   8

String Lengths Without Spaces):
ahmet yuksel - 11
mehmet arslan - 12
mustafa kemal - 12
ali kemal - 8
mustafa kemal - 12
mustafa kemal - 12
mehmet arslan - 12
kemal ahmet - 10
ali kaan - 7
kemal kaan - 9
```