Lab 2 Tasks

Q1. This is a continuation of the phonebook project from the previous labs:

A program named 'phonebook.exe' is provided in the ftp which contains all the functionalities of the phonebook we have created so far. Your task is to modify your own codes so that the menu navigation of your program is similar to that of the given program.

You can use *system("cls")* and *getche()* to create the menu.

system("cls") – removes anything that is written in the console. It is under the *stdlib* header file

getche() – it takes a single character as input from the user. The user does not have to press *enter* after typing the character. The input is taken as soon the character is pressed.

[Hint: the ASCII code for UP arrow key is 72, DOWN arrow key is 80 and ENTER is 13]

Q2. One of the functions of the 'string' library is strtok(). Implement your own version of this function called mystrtok().

char *strtok(char *str, const char *delim)
The C library function char *strtok(char *str, const char *delim) breaks string str into a series of tokens using the delimiter delim.

Parameters:

- str -- The contents of this string are modified and broken into smaller strings (tokens).
- delim -- This is the C string containing the delimiters. These may vary from one call to another.

Return Value:
This function returns a pointer to the last token found in the string. A null pointer is returned if there are no tokens left to retrieve.

The following example shows the usage of strtok() function:

```
#include <string.h>
#include <stdio.h>
int main()
{
    char str1[80] = "abcd-efg-hij";
    char str2[80] = "*klm-n*opq-rst";
    char s1[5] = "-";
    char s2[5] = "*-";
    char *token;

    token = strtok(str1, s1);
    puts(token);

    token = strtok(NULL, s1);
    puts(token);

    token = strtok(str2, s2);
    puts(token);

    token = strtok(NULL, s2);
    puts(token);

    token = strtok(NULL, s2);
    puts(token);

    token = strtok(NULL, s2);
    puts(token);
}
```

OUTPUT:

```
abcd
efg
klm
n
opq
rst
```