



Objective:

- Introduce yourself to Character Array Manipulation.

Given to PF – SE-F17

Task – 1: String Functions

Its better to make a header and put all the function in it. You should name files as MyOOPString.h and MyOOPString.cpp

Not allowed to use any library function of string library like strcmp, strlen etc

1. `int getStringLength(char *);`
2. `void concatenate(char * destination , char * source);`
3. `void copyString(char * destination , char * source);`
4. `void insertString(char * destination , char * source, int index);`
5. `int convertToInteger(char * source);`

The function converts the C-string to an integer and returns that value.

Example Usage: `int num = convertToInteger("4569");`

6. `float convertToFloat(char * source);`

The function converts the C-string to a float and returns that value.

Example Usage: `float num = convertToFloat("3.14159");`

7. `int findSubString(char * destination, char * key, int start=0);`
key

A substring to search for.

destination

destination is the array in which searching has to be performed

start

The index of the character in the string to begin the search with, or 0 to start from the beginning

return Value

The first character index of the string that matches the sub string, -1 otherwise

8. `int findSubString(char * destination, char key, int start=0);`
source

destination is the array in which searching has to be performed

key

A character to search for

start

The index of the character in the string to begin the search with, or 0 to start from the beginning

return Value

The character index of the string that matches the ch, -1 otherwise

9. `int insertString(char * destination, int index, char ch);`
index

The index of the character before which the insertion will take place (start to insert at index and move older elements one index ahead)

ch



A character to be inserted
return Value
The length of the changed string

10. int remove(char * destination, int count=1);
count
The number of characters to be removed from left side. If the count is negative then the number of characters has to be removed from right side.
return Value
The length of the changed string

11. int removeSubString(char * destination, char * source, int index);
search source string in destination and remove it.
return Value
The length of the changed string

12. int remove (char * destination, char ch);
ch
The character to be removed from the destination string
return Value
The count of characters removed from the string. Zero if the string is not changed

13. void trim(char * destination);
Removes all leading and trailing occurrences of white spaces from destination string.

14. void replace(char * destination , char * old, char * new);
find old substring in destination and replace it with new substring.

old
A string containing the character to be replaced by *New*
new
A string containing the character replacing *Old*

15. void makeUpper(char * destination);
Converts all characters of destination array to upper case

16. void makeLower(char * destination);
Converts all characters of destination array to lower case

17. void reverse(char * destination);
reverse the character in destination array.

18. int compare(char * a , char * b, bool caseSensitive=true);
returns 0 if a==b, returns 1 if a<b, returns -1 if a>b. By default the comparison will be case sensitive otherwise it will not be case sensitive as represented by 3rd parameter.

Task - 2: Tic-Tac-Toe

Write a program that allows two players to play a game of tic-tac-toe. Use a two-dimensional char array with three rows and three columns as the game board. Each element of the array should be initialized with a unique number. The program should run a loop that

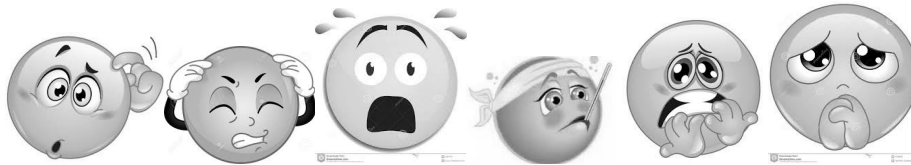
- Displays the contents of the board array
- Allows player 1 to select a location on the board for an X. The program should ask the user to enter the cell number.

- Allows player 2 to select a location on the board for an O. The program should ask the user to enter cell number.
- Determines whether a player has won, or a tie has occurred. If a player has won, the program should declare that player the winner and end the game. If a tie has occurred, the program should say so and end. Player 1 wins when there are three Xs in a row on the game board. The Xs can appear in a row, in a column, or diagonally across the board. A tie occurs when all of the locations on the board are full, but there is no winner.

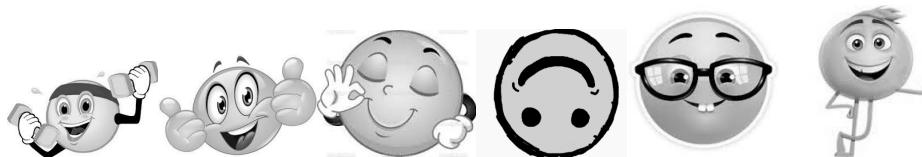
THE PROGRAM
I CODED HAS
LOTS OF BUGS
HOW DO I REMOVE
THEM?



WHY DON'T
YOU PUT ENTIRE
CODE IN
COMMENTS
//



They Can't! Because they think they can't do it. (Always Cry! ☹)



They Can! Because they think they can. (Face it and do it! ☺)



I hope, these smileys
motivate you or you may
also use this ultimate
machine of motivation. ☺

