Assignment - 18 A Job Ready Bootcamp in C++, DSA and IOT MySirG

String and Functions in C Language

1. Write a function to calculate length of the string

#include <stdio.h>

int strLength(char str[])

{

int i;

for (i = 0; str[i]; i++);

return i;

}

int main()

{

char str[100];

printf("Enter string: ");

fgets(str, 100, stdin);

printf("Length of given string is: %d", strLength(str));

return 0;

}

2. Write a function to reverse a string.

#include <stdio.h>

#include <string.h>

void strRev(char str[])

{

int strLength = strlen(str);

for (int i = 0; i < strLength / 2; i++)

{

int temp = str[i];

str[i] = str[strLength - 2 - i];

str[strLength - 2 - i] = temp;

}

}

int main()

{

char str[100];

printf("Enter string: ");

fgets(str, 100, stdin);

strRev(str);

printf("Reverse string: %s",str);

return 0;

}

3. Write a function to compare two strings.

#include <stdio.h>

#include <string.h>

int strCompare(char str1[], char str2[])

{

for (int i = 0; str1[i] || str2[i]; i++)

{

if (str1[i] != str2[i])

return 0;

}

if (strlen(str1) == strlen(str2))

return 1;

return 0;

}

int main()

{

char str1[100], str2[100];

printf("Enter first string: ");

fgets(str1, 100, stdin);

printf("Enter second string: ");

fgets(str2, 100, stdin);

if (strCompare(str1, str2))

printf("\nString is equal");

else

printf("\nString is not equal");

return 0;

}

4. Write a function to transform string into uppercase

#include <stdio.h>

#include <string.h>

void strUppercase(char str[])

{

for (int i = 0; str[i]; i++)

if (str[i] >= 'a' && str[i] <= 'z')

str[i] -= 32;

}

int main()

{

char str[100];

printf("Enter first string: ");

fgets(str, 100, stdin);

strUppercase(str);

printf("\nUppercase string: %s", str);

return 0;

}

5. Write a function to transform a string into lowercase

#include <stdio.h>

#include <string.h>

void strLowercase(char str[])

{

for (int i = 0; str[i]; i++)

if (str[i] >= 'A' && str[i] <= 'Z')

str[i] += 32;

}

int main()

{

char str[100];

printf("Enter first string: ");

fgets(str, 100, stdin);

strLowercase(str);

printf("\nLowercase string: %s", str);

return 0;

}

6. Write a function to check whether a given string is an alphanumeric string or not.

(Alphanumeric string must contain at least one alphabet and one digit)

#include <stdio.h>

#include <string.h>

int strAlphaNumeric(char str[])

{

int digit = 0, alphabet = 0;

for (int i = 0; str[i]; i++)

if (str[i] >= '0' && str[i] <= '9')

digit = 1;

else if ((str[i] >= 'A' && str[i] <= 'Z') || (str[i] >= 'a' && str[i] <= 'z'))

alphabet = 1;

if (digit && alphabet)

return 1;

else

return 0;

}

int main()

{

char str[100];

printf("Enter first string: ");

fgets(str, 100, stdin);

if (strAlphaNumeric(str))

printf("\nString is Alphanumeric!");

else

printf("\nString is Not Alphanumeric!");

return 0;

}

7. Write a function to check whether a given string is palindrome or not.

#include <stdio.h>

#include <string.h>

int strPalindrome(char str[])

{

for (int i = 0; i < strlen(str) / 2; i++)

if (str[i] != str[strlen(str) - 2 - i])

return 0;

return 1;

}

int main()

{

char str[100];

printf("Enter first string: ");

fgets(str, 100, stdin);

if (strPalindrome(str))

printf("\nString is Palindrome!");

else

printf("\nString is Not Palindrome!");

return 0;

}

8. Write a function to count words in a given string

#include <stdio.h>

#include <string.h>

int strCountWord(char str[])

{

int countSpace = 1;

for (int i = 0; str[i]; i++)

if (str[i] == ' ')

countSpace++;

return countSpace;

}

int main()

{

char str[100];

printf("Enter first string: ");

fgets(str, 100, stdin);

printf("\nTotal words in given string: %d", strCountWord(str));

return 0;

}

9. Write a function to reverse a string word wise. (For example if the given string is

“Mysirg Education Services” then the resulting string should be “Services Education

Mysirg” )

#include <stdio.h>

#include <string.h>

void strRevByWords(char str[])

{

int strLength = strlen(str); // str string length - 17

int arr[strLength]; // create 17 length arr array for store reverse word string

int arrInd = 0; // index of arr array

int i = strLength-1; // length - 15 for accessing last index but before null character

int j,k; // extra variable for looping pupose

str[strLength-1] = ' '; // for maintain space (change null to space)

while(i>=0)

{

for(j=i-1; str[j]!=' ' && j>=0; j--); // check it is contain space or less than 0

for(k=j+1; k<=i; k++)

arr[arrInd++] = str[k]; // copy words in arr string

i = j; // update i

}

arr[strLength-1] = '\0'; // maintain null character in last index for exit loop

for(int i=0; arr[i]; i++) // just copy array

str[i] = arr[i];

}

int main()

{

char str[100];

printf("Enter first string: ");

fgets(str, 100, stdin);

strRevByWords(str);

printf("\nReverse string by words: %s", str);

return 0;

}

10. Write a function to find the repeated character in a given string.

#include <stdio.h>

#include <string.h>

void strSort(char str[])

{

for (int i = 0; str[i]; i++)

{

for (int j = 0; str[j]; j++)

{

if (str[i] < str[j])

{

int temp = str[i];

str[i] = str[j];

str[j] = temp;

}

}

}

}

void reapetedChar(char str[])

{

strSort(str);

int i = 0, j, flag;

while (i<strlen(str))

{

flag = 1;

for (j = i + 1; str[j]; j++)

{

if (str[i] == str[j])

flag = 0;

else

break;

}

if (flag == 0)

printf("%c ", str[i]);

i = j;

}

}

int main()

{

char str[100];

printf("Enter first string: ");

fgets(str, 100, stdin);

printf("\nAvailable reapted character in given string- ");

reapetedChar(str);

return 0;

}