Assignment-18 (String and Functions in C Language)

1. Write a function to calculate length of the string

#include<stdio.h>

int stringLength(char []);

int stringLength(char str[])

{

int i;

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

return i;

}

#include<stdio.h>

int main()

{

char str[50];

printf("Enter a string to find its length: ");

gets(str);

printf("Length of %s is %d.", str, stringLength(str));

return 0;

}

2. Write a function to reverse a string.

#include<stdio.h>

#include<string.h>

void reverseString(char []);

void reverseString(char str[])

{

int i, str\_length = 0, last\_num;

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

str\_length++;

last\_num = str\_length - 1;

for(i = 0; i < str\_length / 2; i++)

{

str[i] = str[i] + str[last\_num];

str[last\_num] = str[i] - str[last\_num];

str[i] = str[i] - str[last\_num];

last\_num--;

}

}

int main()

{

char str[50];

printf("Enter a string to reverse: ");

fgets(str, 49, stdin);

str[strlen(str) - 1] = '\0';

reverseString(str);

printf("String after reversing is %s", str);

return 0;

}

3. Write a function to compare two strings.

#include<stdio.h>

int stringComparison(char s1[], char s2[])

{

int i;

for(i = 0; s1[i] != '\0' || s2[i] != '\0'; i++)

if(s1[i] != s2[i])

return s1[i] - s2[i];

return 0;

}

int main()

{

printf("%d", stringComparison("Hello", "Hello"));

return 0;

}

4. Write a function to transform string into uppercase

#include<stdio.h>

#include<string.h>

void tranformStringToUpperCase(char str[])

{

int i;

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

if(str[i] >= 97 && str[i] <= 122)

str[i] = str[i] - 32;

}

int main()

{

char str[50];

printf("Enter a string to convert in upper case: ");

fgets(str, 49, stdin);

str[strlen(str) - 1] = '\0';

tranformStringToUpperCase(str);

printf("String in upper case is %s", str);

return 0;

}

5. Write a function to transform a string into lowercase

#include<stdio.h>

#include<string.h>

void transformStringToLowerCase(char []);

void transformStringToLowerCase(char str[])

{

int i;

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

if(str[i] >= 65 && str[i] <= 90)

str[i] = str[i] + 32;

}

int main()

{

char str[50];

printf("Enter a string to convert in lower case: ");

fgets(str, 49, stdin);

str[strlen(str) - 1] = '\0';

transformStringToLowerCase(str);

printf("String in lower case is %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>

int isStringAlphanumeric(char s[])

{

int i, alphabetPresence = 0, digitPresence = 0;

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

if(s[i] < 48)

return 0;

else if(s[i] >= 58 && s[i] < 65)

return 0;

else if(s[i] >= 91 && s[i] < 97)

return 0;

else if(s[i] >= 123)

return 0;

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

{

if(s[i] >= 65 && s[i] <= 90 || s[i] >= 97 && s[i] <= 122)

{

alphabetPresence = 1;

break;

}

}

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

{

if(s[i] >= 48 && s[i] <= 57)

{

digitPresence = 1;

break;

}

}

if(digitPresence && alphabetPresence)

return 1;

else

return 0;

}

int main()

{

printf("%d", isStringAlphanumeric("1e"));

return 0;

}

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

#include<stdio.h>

#include<string.h>

void reverseString(char str[])

{

int i, str\_length = 0, last\_num;

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

str\_length++;

last\_num = str\_length - 1;

for(i = 0; i < str\_length / 2; i++)

{

str[i] = str[i] + str[last\_num];

str[last\_num] = str[i] - str[last\_num];

str[i] = str[i] - str[last\_num];

last\_num--;

}

}

int isStringPalindrome(char s[])

{

char s2[20];

int i;

strcpy(s2, s);

reverseString(s2);

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

if(s[i] != s2[i])

return 0;

return 1;

}

int main()

{

printf("%d", isStringPalindrome("naman"));

return 0;

}

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

#include<stdio.h>

int wordsCountInString(char s[])

{

int i, words\_count = 1;

for(i = 1; s[i]; i++)

if(s[i] == ' ')

words\_count++;

return words\_count;

}

int main()

{

printf("Number of words in string are %d.", wordsCountInString("Hello World"));

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 stringReverseWordWise(char s[])

{

int i, j, ind = strlen(s), k = 0;

char s2[100];

for(i = strlen(s) - 1; i >= 0; i--)

if(s[i] == ' ' || i == 0)

{

if(i)

{

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

{

s2[k] = s[j];

k++;

}

s2[k] = ' ';

k++;

ind = i;

}

else

for(j = i; j < ind; j++)

{

s2[k] = s[j];

k++;

}

}

s = strcpy(s, s2);

}

int main()

{

char str[50];

printf("Enter a string to reverse it word wise: ");

gets(str);

stringReverseWordWise(str);

printf("String after reversing word wise is %s", str);

return 0;

}

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

#include<stdio.h>

#include<string.h>

void printrepeatedCharactersOfString(char str[])

{

int i, charsFrequency[256] = {0}, k;

char uniqueChars[strlen(str)];

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

{

if(charsFrequency[str[i]] == 0)

{

uniqueChars[k] = str[i];

k++;

}

charsFrequency[str[i]] += 1;

}

for(i = 0; i < k; i++)

{

if(charsFrequency[uniqueChars[i]] > 1)

printf("%c is repeated %d times in %s\n", uniqueChars[i], charsFrequency[uniqueChars[i]] - 1, str);

}

}

int main()

{

printrepeatedCharactersOfString("Hello to the World");

return 0;

}