**Assignment 11**

1. Write a program to scan string from user then scan a single character and search it in a accepted string.

#include <stdio.h>

#include <string.h>

char \*myStrChr(char[], char);

void main()

{

    char str[50];

    printf("\nEnter a String :");

    scanf("%s", str);

    char chr;

    printf("\nEnter a charater to search in string :");

    fflush(stdin);

    scanf("%c", &chr);

    printf("\n Using Inbuilt Function");

    printf("\n%c found at index : %d", chr, strchr(str, chr) - str);

    printf("\n Using User define Function");

    printf("\n%c found at index : %d", chr, myStrChr(str, chr) - str);

}

char \*myStrChr(char str1[], char chr)

{

    for (int i = 0; str1[i] != '\0'; i++)

    {

        if (str1[i] == chr)

        {

            return &str1[i];

        }

    }

    return NULL;

}

Output:

PS C:\Code> & 'c:\Users\bhagv\.vscode\.... \TDM-GCC-64\bin\gdb.exe' '--interpreter=mi'

Enter a String :Bhagvat

Enter a charater to search in string :a

Using Inbuilt Function

a found at index : 2

Using User define Function

a found at index : 2

PS C:\Code>

1. WAP Replace all Occurrences of ‘a’ with $ in a String

#include <stdio.h>

#include <string.h>

char \*replaceA(char[]);

void main()

{

    char str[50];

    printf("\nEnter a String :");

    scanf("%s", str);

    printf("\nNew STring:%s", replaceA(str));

}

char \*replaceA(char str[])

{

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

    {

        if (str[i] == 'a')

        {

            str[i] = '$';

        }

    }

    return str;

}

Output:

PS C:\Code> & 'c:\Users\bhagv\.vscode…. \TDM-GCC-64\bin\gdb.exe' '--interpreter=mi'

Enter a String :Bhagvat

New STring:Bh$gv$t

PS C:\Code>

1. WAP to Remove the nth Index Character from a Non-Empty String

#include <stdio.h>

char \*removeN(char[], int);

void main()

{

    char str[50];

    int n;

    printf("\nEnter A string :");

    scanf("%s", str);

    printf("\nEnter The of element you want to delete :");

    scanf("%d", &n);

    printf("\nResult :%s", removeN(str, n));

}

char \*removeN(char str[], int n)

{    for (int i = n; str[i] != '\0'; i++)

    {

        str[i] = str[i + 1];

    }

    return str;

}

Output:

PS C:\Code> & 'c:\Users\bhagv\.vscode\...... \TDM-GCC-64\bin\gdb.exe' '--interpreter=mi'

Enter A string :Bhagvat

Enter The of element you want to delete :5

Result :Bhagvt

PS C:\Code>

1. WAP to Form a New String where the First Character and the Last Character have been Exchanged

#include <stdio.h>

char \*swapFL(char[]);

void main()

{

    char str[50];

    int n;

    printf("\nEnter A string :");

    scanf("%s", str);

    printf("\nResult :%s", swapFL(str));

}

char \*swapFL(char str[])

{

    int len = strlen(str) - 1;

    char temp = str[0];

    str[0] = str[len];

    str[len] = temp;

    return str;

}

Output:

PS C:\Code> & 'c:\Users\bhagv\.vscode\ \TDM-GCC-64\bin\gdb.exe' '--interpreter=mi'

Enter A string :Bhagvat

Result :thagvaB

PS C:\Code>

1. WAP to Count the Number of Vowels in a String

#include <stdio.h>

int noOfVowelsInSTR(char[]);

void main()

{

    char str[50];

    printf("\nEnter A string :");

    gets(str);

    printf("\nNo Of Vowels in string are :%d", noOfVowelsInSTR(str));

}

int noOfVowelsInSTR(char str[])

{

    int cnt = 0, i = 0;

    while (str[i] != '\0')

    {

        if (str[i] == 'A' || str[i] == 'E' || str[i] == 'I' || str[i] == 'O' || str[i] == 'U' || str[i] == 'a' || str[i] == 'e' || str[i] == 'i' || str[i] == 'o' || str[i] == 'u')

        {

            cnt++;

        }

        i++;

    }

    return cnt;

}

Output:

PS C:\Code> & 'c:\Users\bhagv\.vscode\..... \TDM-GCC-64\bin\gdb.exe' '--interpreter=mi'

Enter A string :Hello How are you

No Of Vowels in string are :7

PS C:\Code>

1. WAP to Take in a String and Replace Every Blank Space with special symbol.

#include <stdio.h>

#include <string.h>

char \*spaceReplaceHashtag(char[]);

void main()

{

    char str[50];

    printf("\nEnter a String :");

    gets(str);

    printf("\nNew STring:%s", spaceReplaceHashtag(str));

}

char \*spaceReplaceHashtag(char str[])

{

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

    {

        if (str[i] == ' ')

        {

            str[i] = '#';

        }

    }

    return str;

}

Output : PS C:\Code> & 'c:\Users\bhagv\.vscode\.... \TDM-GCC-64\bin\gdb.exe' '--interpreter=mi'

Enter a String :Helo bro how are you

New STring:Helo#bro#how#are#you#

PS C:\Code>

1. WAP to Remove the Characters of Odd Index Values in a String

#include <stdio.h>

char \*removeOddIndexValues(char[]);

void main()

{

    char str[50];

    int n;

    printf("\nEnter A string :");

    gets(str);

    printf("\nResult :%s", removeOddIndexValues(str));

}

char \*removeOddIndexValues(char str[])

{

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

    {

        if (i % 2 != 0)

        {

            str[i] = ' ';

        }

    }

    return str;

}

Output :

PS C:\Code> & 'c:\Users\bhagv\.vscode\..... \TDM-GCC-64\bin\gdb.exe' '--interpreter=mi'

Enter A string :Hello I AM Bhagvat

Result :H l o I A h g a

PS C:\Code>

1. WAP to Calculate the Number of Words Present in a String

#include <stdio.h>

int noOfWords(char[]);

void main()

{

    char str[50];

    int n;

    printf("\nEnter A string :");

    gets(str);

    printf("\nNo of words in string are: %d", noOfWords(str));

}

int noOfWords(char str[])

{

    int cnt = 1;

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

    {

        if (str[i] == ' ')

        {

            cnt++;

            // printf("\n%d Count ", cnt);

        }

    }

    // printf("\n%d Count ", cnt);

    return cnt;

}

Output :

PS C:\Code> & 'c:\Users\bhagv\.vscode\ \TDM-GCC-64\bin\gdb.exe' '--interpreter=mi'

Enter A string :Hey Bro how are you ? Im here to help you!

No of words in string are: 11

PS C:\Code>

1. WAP to Take in Two Strings and Display the Larger String without Using Built-in Functions

#include <stdio.h>

int myStrCmp(char[], char[]);

int main()

{

    char str1[100], str2[100];

    printf("Enter the first string: ");

    gets(str1);

    printf("Enter the second string: ");

    gets(str2);

    int re = myStrCmp(str1, str2);

    if (re == 1)

    {

        printf("The larger string is: %s", str1);

    }

    else if (re == -1)

    {

        printf("The larger string is: %s", str2);

    }

    else

    {

        printf("Both strings are of equal length.");

    }

}

int myStrLen(char str[])

{

    int i = 0;

    for (; str[i] != '\0'; i++)

        ;

    return i;

}

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

{

    int len1 = myStrLen(str1);

    int len2 = myStrLen(str2);

    if (len1 > len2)

    {

        return 1;

    }

    else if (len2 > len1)

    {

        return -1;

    }

    else

    {

        return 0;

    }

}

Output:

PS C:\Code> & 'c:\Users\bhagv\.vscode\..\TDM-GCC-64\bin\gdb.exe' '--interpreter=mi'

Enter the first string: Bhagvat

Enter the second string: Bhagvat

Both strings are of equal length.

PS C:\Code> & 'c:\Users\bhagv\.vscode\extensions\.... \TDM-GCC-64\bin\gdb.exe' '--interpreter=mi'

Enter the first string: hello

Enter the second string: hi

The larger string is: hello

PS C:\Code>

1. Write a program to check the string is palindrome or not.

#include <stdio.h>

#include <stdlib.h>

char \*strDup(char[]);

int myStrLen(char[]);

char \*myStrRev(char[]);

int myStrCmp(char[], char[]);

int main()

{

    char str1[100], str2[100];

    printf("Enter the string: ");

    gets(str1);

    if (myStrCmp(strDup(str1), myStrRev(str1)))

    {

        printf("\nString is not Palindrome.!!");

    }

    else

    {

        printf("\nString is Palindrome.!!");

    }

}

char \*strDup(char str[])

{

    int len = myStrLen(str);

    int i;

    char \*dup = (char \*)malloc(sizeof(char) \* len);

    for (i = 0; i < len; i++)

    {

        dup[i] = str[i];

    }

    dup[i] = '\0';

    return dup;

}

int myStrLen(char str[])

{

    int i = 0;

    for (; str[i] != '\0'; i++)

        ;

    return i;

}

char \*myStrRev(char str[])

{

    int len = myStrLen(str);

    char \*temp = (char \*)malloc(sizeof(char) \* len);

    int j = 0;

    for (int i = len - 1; i >= 0; i--, j++)

    {

        temp[j] = str[i];

    }

    temp[j] = '\0';

    return temp;

}

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

{

    int i = 0;

    while (str1[i] != '\0' && str2[i] != '\0')

    {

        if (str1[i] != str2[i])

        {

            return str1[i] - str2[i];

        }

        i++;

    }

    if (str1[i] == '\0' && str2[i] == '\0')

    {

        return 0;

    }

    else

    {

        return 1;

    }

}

Output :

PS C:\Code> & 'c:\Users\bhagv\.vscode\.... \TDM-GCC-64\bin\gdb.exe' '--interpreter=mi'

Enter the string: Bhagvat

String is not Palindrome.!!

PS C:\Code> & 'c:\Users\bhagv\.vscode\... \TDM-GCC-64\bin\gdb.exe' '--interpreter=mi'

Enter the string: naman

String is Palindrome.!!

PS C:\Code>