**MODULE: SE – Fundamentals of Programming**

**Topics Covered**

**String**

**Que.1 Write a program in C to find the length of a string without using library functions.**

#include <stdio.h>

int main()

{

char str[100];

int i,length=0;

printf("Enter a string: \n");

scanf("%s",str);

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

{

length++;

}

printf("\nLength of input string: %d",length);

return 0;

}

**Que.2 Write a program in C to separate individual characters from a string.**

#include <stdio.h>

#include <string.h>

int main() {

char str[90]=" ";

int l=0;

printf("Enter the string");

scanf("%s",str);

while (str[l]!='\0'){

printf("%c \n",str[l]);

l++;

}

return 0;

}

**Que.3 Write a program in C to print individual characters of a string in reverse order**

#include <stdio.h>

#include <string.h>

int main() {

char str[50]=" ";

int l,i ;

printf("Enter the string");

scanf("%s",str);

l=strlen(str);

for (i=l; i>=0; i--){

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

}

return 0;

}

**Que.4 Write a program in C to count the total number of words in a string.**

#include <stdio.h>

#include <string.h>

int main(){

char name[90];

int i=0;

int len=0, word=0;

printf("Enter the sentence:\n ");

fgets(name,sizeof(name),stdin);

len=strlen(name);

for (i=0;i<len;i++){

if (name[i]==+ ' '){

word++;

}

}

printf("the number of words :%d",word+1);

}

**Que.5 Write a program in C to compare two strings without using string library functions.**

#include<stdio.h>

#include<string.h>

int main()

{

char str[20];

printf("Enter a string: ");

scanf("%s", str);

char newstr[20];

printf("Enter another string: ");

scanf("%s", newstr);

int flag = 0;

int i = 0;

while(i < strlen(str))

{

if(str[i] != newstr[i])

{

flag = 1;

break;

}

i++;

}

if(flag == 0)

printf("Strings are equal");

else

printf("Strings are not equal");

return 0;

}

**Que.6. Write a program in C to count the total number of alphabets, digits and special characters in a string.**

#include <stdio.h>

void main()

{

char str[50];

int chars = 0, digits = 0, spec\_chars = 0, i;

printf("Enter string: ");

gets(str);

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

{

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

chars++;

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

digits++;

else

spec\_chars++;

}

printf("\nNumber of alphabets: %d\n", chars);

printf("Number of digits: %d\n", digits);

printf("Number of special characters: %d\n", spec\_chars);

}

**Que.7 Write a program in C to copy one string to another string.**

#include <stdio.h>

int main() {

char s1[100];

char s2[100];

printf("Enter any string: ");

gets(s1);

int i;

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

s2[i]=s1[i];

}

s2[i]='\0';

printf("original string s1='%s'\n",s1);

printf("copied string s2='%s'",s2);

return 0;

}

**Que.8 Write a program in C to count the total number of vowels or consonants in a string.**

#include <stdio.h>

int main() {

char str[100];

int i = 0, vol = 0, cont = 0;

printf("Enter a string: ");

fgets(str, sizeof(str), stdin);

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') {

vol++;

}

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

cont++;

}

i++;

}

printf("Total number of vowels: %d\n", vol);

printf("Total number of consonants: %d\n", cont);

return 0;

}

**Que.9 Write a program in C to find the maximum number of characters in a string.**

#include <stdio.h>

#define CHARS 255

int main(){

int SIZE=100;

char string[SIZE];

int frequency[CHARS];

int i = 0, maximum;

int value;

printf("Enter the string: ");

gets(string);

for(i=0; i<CHARS; i++){

frequency[i] = 0;

}

i=0;

while(string[i] != '\0'){

value = (int)string[i];

frequency[value] += 1;

i++;

}

maximum = 0;

for(i=0; i<CHARS; i++){

if(frequency[i] > frequency[maximum])

maximum = i;

}

printf("Maximum occurrence character is '%c' = %d times.", maximum,

frequency[maximum]);

return 0;

}

**Que.10 Write a program in C to extract a substring from a given string**

#include <stdio.h>

int main() {

char str[100], substr[100];

int start, length, i;

printf("Enter a string: ");

fgets(str, sizeof(str), stdin);

printf("Enter the starting position: ");

scanf("%d", &start);

printf("Enter the length of the substring: ");

scanf("%d", &length);

for (i = 0; i < length && str[start + i] != '\0'; i++) {

substr[i] = str[start + i];

}

substr[i] = '\0';

printf("The extracted substring is: %s\n", substr);

return 0;

}

**Que.11 Write a program in C to read a sentence and replace lowercase characters with uppercase and vice versa.**

#include <stdio.h>

int main() {

char str[100];

int i = 0;

printf("Enter a sentence: ");

fgets(str, sizeof(str), stdin);

while (str[i] != '\0') {

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

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

}

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

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

}

i++;

}

printf("The modified sentence is: %s", str);

return 0;

}

**Que.12 Write a program in C to find the number of times a given word 'is' appears in the given string.**

#include <stdio.h>

#include <string.h>

int countWord(char \* str, char \* toSearch);

int main()

{

int MAX\_SIZE= 100;

char str[MAX\_SIZE];

char toSearch[MAX\_SIZE];

int count;

printf("Enter any string: ");

gets(str);

printf("Enter word to find number of times ");

gets(toSearch);

count = countWord(str, toSearch);

printf("Total occurrences of '%s': %d", toSearch, count);

return 0;

}

int countWord(char \* str, char \* toSearch)

{

int i, j, found, count;

int stringLen, searchLen;

stringLen = strlen(str);

searchLen = strlen(toSearch);

count = 0;

for(i=0; i <= stringLen-searchLen; i++)

{

found = 1;

for(j=0; j<searchLen; j++) {

if(str[i + j] != toSearch[j])

{

found = 0;

break; }

}

if(found == 1)

{

count++;

}

}

return count;

}

**Que.13 Write a program in C to remove characters from a string except alphabets.**

#include <stdio.h>

int main() {

char line[150];

printf("Enter a string: ");

fgets(line, sizeof(line), stdin);

for (int i = 0, j; line[i] != '\0'; ++i) {

while (!(line[i] >= 'a' && line[i] <= 'z') && !(line[i] >= 'A' && line[i] <= 'Z') && !(line[i] == '\0')) {

for (j = i; line[j] != '\0'; ++j) {

line[j] = line[j + 1];

}

line[j] = '\0';

}

}

printf("Output String: ");

puts(line);

return 0;

}

**Que.14 Write a program in C to combine two strings manually**

#include <stdio.h>

#include <string.h>

int main() {

char str1[100], str2[100], i, j, l, m, k; /

printf("\n\nConcatenate Two Strings Manually :\n");

printf("-------------------------------------\n");

printf("Input the first string : ");

fgets(str1, sizeof str1, stdin);

printf("Input the second string : ");

fgets(str2, sizeof str2, stdin);

l = strlen(str1);

m = strlen(str2);

for (i = 0; i < l - 1; ++i);

str1[i] = ' ';

i++;

for (j = 0; j < m - 1; ++j, ++i) {

str1[i] = str2[j];

}

k = strlen(str1);

printf("After concatenation the string is : \n ");

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

printf("%c", str1[i]);

}

printf("\n\n");

return 0;

}

**Que.15 Write a program in C to find the largest and smallest words in a string.**

#include <stdio.h>

#include <string.h>

#include <ctype.h>

int main() {

char str[100], word[20], mx[20], mn[20], c;

int i = 0, j = 0, flg = 0;

printf("\n\nFind the largest and smallest word in a string :\n");

printf("-----------------------------------------------------\n");

printf("Input the string : ");

i = 0;

while ((c = getchar()) != '\n' && i < sizeof(str) - 1) { until newline or end of array

if (isalnum(c) || isspace(c)) {

str[i++] = c;

}

}

str[i] = '\0';

for (i = 0; i < strlen(str); i++) {

while (i < strlen(str) && !isspace(str[i]) && isalnum(str[i])) {

word[j++] = str[i++];

}

if (j != 0) {

word[j] = '\0';

if (!flg) {

flg = !flg;

strcpy(mx, word);

strcpy(mn, word);

}

if (strlen(word) > strlen(mx)) {

strcpy(mx, word);

}

if (strlen(word) < strlen(mn)) {

strcpy(mn, word);

}

j = 0;

}

}

printf("The largest word is '%s' \nand the smallest word is '%s' \nin the string: '%s'.\n", mx, mn, str);

return 0;

}