**6. Strings**

**a. Write a C program to convert a Roman numeral ranging from I to L to its decimal equivalent.**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

void main()

{

char rom[30];

int a[30], len, i, k;

printf("Enter the roman number\n");

scanf("%s", &rom);

len =strlen(rom);

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

{

switch (rom[i])

{

case 'I': a[i] = 1;

break;

case 'V': a[i] = 5;

break;

case 'X': a[i] = 10;

break;

case 'L': a[i] = 50;

break;

case 'C': a[i] = 100;

break;

case 'D': a[i] = 500;

break;

case 'M': a[i] = 1000;

break;

default : printf("Invalid choice");

break;

}

}

k = a[len - 1];

for(i = len - 1; i > 0; i--)

{

if(a[i] > a[i – 1])

{

k = k - a[i – 1];

}

if(a[i] <= a[i – 1])

{

k = k + a[i – 1];

}

}

printf("decimal equivalent is %d", k);

}

**Output**

**b. Write a C program that converts a number ranging from 1 to 50 to Roman equivalent**

#include <stdio.h>

int main () {

int num;

printf("Enter your input:");

scanf("%d", &num);

printf("Roman Number: ");

while (num > 0) {

if (num >= 1000) {

/\* M - 1000 \*/

printf("M");

num = num - 1000;

} else if (num >= 500) {

/\*

\* D is 500. CM is 900

\* CM = M - C = 1000 - 100 => 900

\*/

if (num >= 900) {

printf("CM");

num = num - 900;

} else {

printf("D");

num = num - 500;

}

} else if (num >= 100) {

/\* C is 100. CD is 400

\* CD = D - C = 500 - 100 => 400

\*/

if (num >= 400) {

printf("CD");

num = num - 400;

} else {

printf("C");

num = num - 100;

}

} else if (num >= 50) {

/\* L is 50. XC is 90

\* XC = C - X = 100 - 10 => 90

\*/

if (num >= 90) {

printf("XC");

num = num - 90;

} else {

printf("L");

num = num - 50;

}

} else if (num >= 9) {

/\* XL is 40. IX is 9. X is 10

\* XL = L - X = 50 - 10 = 40

\* IX = X - I = 10 - 1 = 9

\*/

if (num >= 40) {

printf("XL");

num = num - 40;

} else if (num == 9) {

printf("IX");

num = num - 9;

} else {

printf("X");

num = num - 10;

}

} else if (num >= 4) {

/\* V is 5 and IV is 4

\* IV = V - I = 5 - 1 => 4

\*/

if (num >= 5) {

printf("V");

num = num - 5;

} else {

printf("IV");

num = num - 4;

}

} else {

printf("I");

num = num - 1;

}

}

printf("\n");

}

**Output**

**c. Write a C program that uses functions to perform the following operations:**

**i. To insert a sub-string in to a given main string from a given position.**

#include<stdio.h>

#include<string.h>

void main()

{

char str1[20], str2[20];

int l1, l2, n, i;

puts("Enter the string 1\n");

gets(str1);

l1 = strlen(str1);

puts("Enter the string 2\n");

gets(str2);

l2 = strlen(str2);

printf("Enter the position where the string is to be inserted\n");

scanf("%d", &n);

for(i = n; i < l1; i++)

{

str1[i + l2] = str1[i];

}

for(i = 0; i < l2; i++)

{

str1[n + i] = str2[i];

}

str2[l2 + 1] = '\0';

printf("After inserting the string is %s", str1);

}

**Output:**

**ii. To delete n Characters from a given position in a given string.**

#include<stdio.h>

#include<string.h>

void main()

{

char str[20];

int i, n, l, pos;

puts("Enter the string\n");

gets(str);

printf("Enter the position where the characters are to be deleted\n");

scanf("%d", &pos);

printf("Enter the number of characters to be deleted\n");

scanf("%d", &n);

l = strlen(str);

for(i = pos + n; i < l; i++)

{

str[i - n] = str[i];

}

str[i - n] = '\0';

printf("The string is %s", str);

}

**Output:**

**f. Write a C program to determine if the given string is a palindrome or not (Spelled same in both directions with or without a meaning like madam, civic, noon, abcba, etc.)**

#include <stdio.h>

#include <string.h>

void main()

{

char str[20];

int i, l, f = 0;

printf("Enter any string\n");

gets(str);

l = strlen(str);

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

{

if(str[i] == str[l - 1 - i])

f = f + 1;

}

if(f == l)

{

printf("The string is palindrome");

}

else

{

printf("The string is not a palindrome");

}

}

**Output:**

**g. Write a C program that displays the position of a character ch in the string S or – 1 if S doesn‘t contain ch.**

#include<stdio.h>

#include<string.h>

void main()

{

char s[30], t[20];

char \*found;

puts("Enter the first string: ");

gets(s);

puts("Enter the string to be searched: ");

gets(t);

found = strstr(s, t);

if(found)

{

printf("Second String is found in the First String at %d position.\n", found - s);

}

else

{

printf("-1");

}

}

**Output:**

**h. Write a C program to count the lines, words and characters in a given text.**

#include <stdio.h>

#include <string.h>

void main()

{

char str[100];

int i = 0, l = 0, f = 1;

puts("Enter any string\n");

gets(str);

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

{

l = l + 1;

}

printf("The number of characters in the string are %d\n", l);

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

{

if(str[i] == ' ')

{

f = f + 1;

}

}

printf("The number of words in the string are %d", f);

}

**Output:**