length of the string

```
#include<stdio.h>
#include<stdio.h>
        void main()
        {
        char s[20],*p;
        int c=0;
        printf("enter the string\n");
        scanf("%s",s);
        //gets(s);
        p=s;
        while (*p++!='\0')
       c++;
        printf("length of the string is %d",c);
        }
```

string concatenation

```
#include<stdio.h>
        void main()
        {
        char s1[20],s2[10],*p;int i;
        printf("enter string1\n");
        scanf("%s",s1);
        printf("enter string2\n");
        scanf("%s",s2);
        p=s1;
        while(*p++!='\0');
        p--;
        while(s2[i]!='\0')
        {*p++=s2[i];
        i++;}
        *(p+1)='\0';

        printf("Concatenated string is %s\n",s1);}
```

string comparison

```
#include<stdio.h>
        void main()
        {
        char s1[20],s2[10];
        int i,j;
        printf("enter string1\n");
        scanf("%s",s1);
        printf("enter string2\n");
        scanf("%s",s2);
        i=0;
        while(s1[i]==s2[i] &&(s1[i]!='\0' ||s2[i]!='\0'))
       i++;

        if(s1[i]==s2[i])
        printf("Strings are same\n",s1);
        else
        printf("Strings are different\n");
        }
```

//to insert a sub string

```
#include<stdio.h>
void main()
{
char s[20],subs[10];
 int i,pos,l1,l2,j;
printf("enter the string\n");
scanf("%s",s);
printf("enter the substring\n");
scanf("%s",subs);
printf(" enter the position of the string to be inserted\n");
scanf("%d",&pos);
i=0;
while(s[i++]!='\0');
l1=i-1;
printf("length of string1--%d\n",l1);
i=0;
```

```
while(subs[i++]!='\0');
l2=i-1;
printf("length of string2--%d\n",l2);
for(i=l1;i>=pos;i--)
s[i+l2]=s[l1--];
s[i+1]='\0';
printf("string after moving--%s\n",s);
for(i=pos,j=0;i<pos+l2;i++)
    s[i]=subs[j++];
     printf("string after insertion---%s\n",s);

}
```

to delete a substring to delete a substring

```
#include<stdio.h>

void main()
{
char s[20],subs[10];
 int i,l1,l2,j,pos;
printf("enter the string\n");
gets(s);
printf("enter the substring to be deleted\n");
scanf("%s",subs);
i=0;
while(s[i++]!='\0');
l1=i-1;
i=0;
while(subs[i++]!='\0');
l2=i-1;

for(i=0;i<=l1-l2;i++)
{
j=0;
while(j<l2&&subs[j]==s[i+j])
j++;
if(j==l2)
{pos=i;
break;}}
printf("substring found at %d\n",i);
for(i=pos;i<=l1;i++)
{s[i]=s[i+l2]; printf("%c\n",s[i+2]);}
```

```
s[l1-l2]='\0';
printf("String after deleting substring %s is %s\n",subs,s);
}
```

2. Write a C program to define a student structure with the data members to store name, roll no and grade of the student. Also write the required functions to read, display, and sort student information according to the roll number of the student. All the member functions will have array of objects as arguments.

```
#include<stdio.h>

struct student{

char name[10],grade;

int rollno;

};

int size=10;

void read_student(struct student s[],int n)

{int i;

    printf(" enter student data\n");

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

    {

        printf("enter   name\n");

                scanf(" %s",s[i].name);

                printf("enter roll no\n");

        scanf("%d",&s[i].rollno);

        printf("enter grade\n");

        scanf(" %c",&s[i].grade);


    }
```

```
}
void display_student(struct student s[],int n)
{int i;
     printf(" Student Data\n");
     printf("\n%-13s %-13s %-10s\n", "Name", "Roll No", "Grade");
    for(i = 0;i<n;i++)
  printf("%-13s %-13d %-5c\n",s[i].name, s[i].rollno, s[i].grade);
}
void sort_student(struct student s[],int n)
{int i,j,ps; struct student t;
for(i=0;i<n-1;i++)
    {ps=i;
        for(j=i+1;j<n;j++)
        if(s[ps].rollno>s[j].rollno)
    {ps=j;
    }
    t=s[ps];
        s[ps]=s[i];
        s[i]=t;            }       }

void main()
{int n; struct student p[size];
    printf("enter the number of students\n");
    scanf("%d",&n);
    read_student(p,n);
    display_student(p,n);
    sort_student(p,n);
```

```
display_student(p,n);
}
```

3.

Define a structure Student with the following members:
char name [50] – to store student name as a string
int roll_no – to store roll number
float marks – to store marks
Write a C program that:

i Reads the details of ‘n’ students using a function that uses pointer to structure as an argument.
ii Displays the details of all students using a separate function.
iii Finds and displays the student with the highest marks using pointer-based access.

```
#include<stdio.h>
struct student{
char name[10],grade;
int rollno;
};
int size=10;
void read_student(struct student s[],int n)
{int i;
    printf(" enter student data\n");
    for(i=0;i<n;i++)
    {
        printf("enter   name\n");
                scanf(" %s",s[i].name);
                printf("enter roll no\n");
        scanf("%d",&s[i].rollno);
        printf("enter grade\n");
```

```
        scanf(" %c",&s[i].grade);


    }


}
void display_student(struct student s[],int n)
{int i;
     printf(" Student Data\n");
     printf("\n%-13s %-13s %-10s\n", "Name", "Roll No", "Grade");
    for(i = 0;i<n;i++)
  printf("%-13s %-13d %-5c\n",s[i].name, s[i].rollno, s[i].grade);


}
void sort_student(struct student s[],int n)
{int i,j,ps; struct student t;
for(i=0;i<n-1;i++)
    {ps=i;
        for(j=i+1;j<n;j++)
        if(s[ps].rollno>s[j].rollno)
    {ps=j;
    }
    t=s[ps];
        s[ps]=s[i];
        s[i]=t;


    }
```

```
}



void main()
{int n; struct student p[size];
    printf("enter the number of students\n");
    scanf("%d",&n);
    read_student(p,n);
    display_student(p,n);
    sort_student(p,n);
    display_student(p,n);
    }
```