7b.+ and== overloading

```
const int sz=80;
enum boolean{false,true};
class string
{
char str[80];
public:
string(){strcpy(str," ");}
string(char s[ ]) {strcpy(str,s);}
void display() {cout << str;}
void getstr() { gets(str);}
boolean operator==(string ss)
{
return
(strcmp(str,ss.str)==0)?true:false;
}
string operator +(string ss)
{
return strcat(str,ss.str);
}
};
void main() {
string s1,s2,s3;
int ch;
```

```
cout << "enter the first string\n";
s1.getstr();
cout << "enter second string\n";
s2.getstr();
do { cout << "Menu";
cout << "1.compare two strings\n";
cout << "2.concatinate two strings\n";
cout << "enter your choice\n";
cin >> ch;
switch(ch) {
case 1:if(s1==s2)
    cout << "strings are equal\n";
    else
    cout << "strings are not equal\n";
    break;
case 2:s3=s1+s2;
    cout << "concatenated string is\n";
    s3.display();
    break;   }
}while(ch==1);
getch(); }
```

7c.

6.Write a C++ program to create a class called STRING and implement the following operations.

Display the results after every operation by overloading <<.

i) STRING s1 = “ISE”

ii) STRING s2 = “MSRIT”

iii) STRING s3 = s1+s2 (Use copy constructor)

```
#include<iostream>

#include<cstring>

#include<cstdlib>

using namespace std;

class strng

{

char str[20];

public:

strng()

{

str[0]=&#39;\0&#39;;

}

strng(char temp[])

{

strcpy(str,temp);

}


strng(strng &temp)

{

strcpy(str,temp.str);

}

void display();
```

```
friend strng operator+(strng s1,strng s2);

friend ostream & operator<<(ostream&,strng&);

};

void strng::display()

{

cout<<"\nstring is"<<str;

}

strng operator+(strng s1,strng s2)

{

strcat(s1.str,s2.str);

return s1;

}

ostream & operator<<(ostream& os,strng& s)

{

os<<s.str<<endl;

return os;

}

int main()


{

strng s1("ise");

strng s2("MSRIT");

strng s3;

cout<<"\nBEFORE CONCATINATION";

cout<<"\n s1="<<s1;
```

```
cout<<"\n s2="<<s2;

s3=s1+s2;

cout<<"\nAFTER CONCATENATION"<<"\n:";

cout<<"\n s1+s2="<<s3;

return 0;

}
```

7d.

Write a C++ program to create a class called OCTAL, which has the characteristics of an octal

number. Implement the following operations by writing an appropriate constructor and an

overloaded operator +.

i. OCTAL h = x ; where x is an integer

ii. int y = h + k ; where h is an OCTAL object and k is an integer.

Display the OCTAL result by overloading the operator <<. Also display the values of h and y.

```
#include<iostream>

using namespace std;
```

```
class octal

{

int oct,dec,ten;

public:

octal()

{

oct=0;

ten=1;

}

void operator=(int x)

{

int r;

dec=x;
```

```
while(x!=0)

{

r=x%8;

x=x/8;

oct=oct+ten*r;

ten=ten*10;

}

}

int operator+(int k)

{

return(dec+k);

}
```

```
friend ostream & operator<<(ostream&,octal& c);


};
 ostream & operator<<(ostream& sout,octal&c)
 {
 sout<<c.oct;
 }
 int main()
 {
 octal h;
 int n,k;
 cout<<"Enter a integer to change to octal: ";
 cin>>n;
 cout<<endl;
 h=n;
 cout<<"The octal value of "<<n<<" is: "<<h<<endl;
 cout<<"\nEnter integer to be added to previous octal: ";
 cin>>k;
 cout<<endl;
 int y=h+k;
 cout<<"Integer sum of octal and integer is: "<<y<<"\n";
 return 0;
 }
```