Program 1

/* Write a program to read series of name, one per line, from standard input and write these names spelled in reverse order to the standard output using I/O redirection and pipes. Repeat the exercise using an input file specified by the user instead of the standard input and using an output file specified by the user instead of the standard output */

```
#include<iostream.h>
#include<stdio.h>
#include<fstream.h>
#include<conio.h>
#include<iomanip.h>
#include<stdlib.h>

// function to reverse the string
void reverse(char *s,char *r)
{
        int j,len=0;
        while(s[len]!='\0')          // to calculate the length of string
        len++;
        for(j=len-1;j>=0;j--)
          r[len-j-1]=s[j];
        r[len]='\0';
}

// main program

void main()
{
        char name[10][20],rev[10][20],input[20],output[20],str[20],rstr[20];
        int i,n,len;
        fstream ifile,ofile;
        clrscr();
        cout<<"enter the number of names to read "<<endl;
        cin>>n;
        cout<<"enter the names"<<endl;
        for(i=0;i<n;i++)
        gets(name[i]);
        for(i=0;i<n;i++)
        reverse(name[i],rev[i]);
        cout<<"the names and its reverese order are"<<endl;
        for(i=0;i<n;i++)
        cout<<name[i]<<setw(25)<<rev[i]<<endl;
        cout<<"enter the filename which contain list of names"<<endl;
```

```
        cin>>input;
        ifile.open(input,ios::in);
        if(!ifile)
        {
                cout<<"file doesnot exist";
                getch();
                exit(1);
        }
        cout<<"enter the filename to store names in reverse order"<<endl;
        cin>>output;
        ofile.open(output,ios::out);
        if(!ofile)
        {
                cout<<"file doesnot exit";
                getch();
                exit(1);
        }
        while(!ifile.eof())
        {
                ifile.getline(str,20,'\n');
                reverse(str,rstr);
                ofile<<rstr<<endl;
        }
        getch();
}
```

Output

Output 1:

```
enter the number of names to read
3
enter the names
michael j folk
bill zoellick
greg riccardi

the names and its reverese order are
michael j folk        klof j leahcim
bill zoellick         kcilleoz llib
greg riccardi         idraccir gerg

enter the filename which contain list of names
abc.dat
enter the filename to store reverese the names
xyz.dat

c:\tc>type abc.dat
```

manoj kumar
praveen kollegal
vikaram narayan
sathish madappa
nemi chand
yadhu nandan

c:\tc>type xyz.dat

ramuk jonam
lagellok neevarp
nayaran marakiv
appadam hsihtas
dnahc imen
nadnan uhday

Output 2:

enter the number of names to read
2
enter the names
nagaraj poojari
shivaraj

the names and its reverese order are
nagaraj poojari irajoop jaragan
shivaraj jaravihs

enter the filename which contain list of names
pqr.txt
file doesnot exist

Output 3: using I/O redirection and pipes (Run the program in Command prompt)

I/O redirection : Redirect the ouput from *stdout* to a file aaa.txt
Syntax : program1 >filename

NOTE: go to command prompt
File- DOS Shell
C:\tc>program name >any.txt file
Ex: c:\tc>prog1>aaa.txt

c:\tc>prog1 > aaa.txt
1
rnsit college
zzz.txt

c:\tc>type aaa.txt

enter the number of names to read
enter the names
the names and its reverese order are
rnsit college egelloc tisnr
enter the filename which contain list of names

file doesnot exist

Pipes : take any *stdout* output from program 1 and use it in place of any *stdin* input to program2.
Syntax : program1 | program 2

c:\tc>type xyz.dat | sort

appadam hsihtas
dnahc imen
lagellok neevarp
nadnan uhday
nayaran marakiv
ramuk jonam

Program 2

/* Write a program to read and write student objects with fixed length records and the fields delimited by "|". Implement pack () and unpack (), modify() and search() methods */

```
#include<iostream.h>
#include<fstream.h>
#include<conio.h>
#include<stdio.h>
#include<iomanip.h>
#include<stdlib.h>
#include<string.h>

#define filename "std2.txt"

fstreamifile;

class student
{
        char usn[15],name[20],age[5],branch[6],sem[5];
       public:
              void opener(fstream&ifile,char *fn,int mode);
              void read();
              void pack();
              void display();
              void unpack();
              int search();
              void modify(int);
};

// function  to  open a file
void student::opener(fstream&sfile,char *fn,int mode)
{
       sfile.open(fn,mode);
       if(!sfile)
       {
              cout<<"unable to open a file"<<endl;
              getch();
              exit(1);
       }
}
```

```
//function to read the student record
void student::read()
{
cout<<"enter the usn number:";          gets(usn);
cout<<"enter the name:";       gets(name);
cout<<"enter the age:";        gets(age);
cout<<"enter the branch:";     gets(branch);
cout<<"enter the sem:";        gets(sem);
pack();

}

// function to pack the student record using delimiter
void student::pack()
{
char buffer[75];
strcpy(buffer,usn);                strcat(buffer,"|");
strcat(buffer,name);               strcat(buffer,"|");
strcat(buffer,age);                strcat(buffer,"|");
strcat(buffer,branch);                     strcat(buffer,"|");
strcat(buffer,sem);                        strcat(buffer,"|");
ifile.fill('*');
ifile<<setiosflags(ios::left)<<setw(sizeof(student))<<buffer<<endl;
}


//function to display student record
void student::display()
 {
char buffer[75];
cout<<setiosflags(ios::left);
cout<<setw(15)<<"USN"<<setw(20)<<"NAME"<<setw(5)<<"AGE";
cout<<setw(10)<<"BRANCH"<<setw(5)<<"SEM"<<endl;
while(1)
  {
        unpack();
        if(ifile.eof())
                break;
        if(usn[0]!='$')
         {
                cout<<setw(15)<<usn<<setw(20)<<name<<setw(5)<<age;
                cout<<setw(10)<<branch<<setw(5)<<sem<<endl;
         }
   }
}
```

```
// function to  unpack
void student::unpack()
{
        char dummy[75];
        ifile.getline(usn,15,'|');
        ifile.getline(name,20,'|');
        ifile.getline(age,5,'|');
        ifile.getline(branch,6,'|');
        ifile.getline(sem,5,'|');
        ifile.getline(dummy,75,'\n');
}

//function to search student record based on USN.
int student::search()
{
        int flag;
        charsusn[15];
        cout<<"enter the usn to be searched:";
        cin>>susn;
        while(!ifile.eof())
         {
                flag=ifile.tellg();
                unpack();
                if(usn[0]!='$' &&strcmp(usn,susn)==0)
                 {

                cout<<"USN:"<<usn<<"\n"<<"NAME:"<<name<<"\n"<<"AGE:"<<age;
                cout<<"\n"<<"BRANCH:"<<branch<<"\n"<<"SEM:"<<sem<<"\n";
                return flag;
                 }
         }
        return -1;
}

//function to modify record.
void student::modify(intrecpos)
{
ifile.seekp(recpos,ios::beg);
ifile.put('$');
ifile.seekp(0,ios::end);
read();

}
```

```
//main program
void main()
{
        intch,flag;
        student s;
        clrscr();
        for(;;)
         {
                cout<<endl<<"1.- read\t2- display\t 3 .-search\t4.- modify\t5.- exit"<<endl;
                cout<<"enter the choice:";
                cin>>ch;
                switch(ch)
                {
                        case 1: s.opener(ifile,filename,ios::app);
                                cout<<"enter the student details\n";
                                s.read();
                                break;
                        case 2: s.opener(ifile,filename,ios::in);
                                cout<<"The student details are:"<<endl;
                                s.display();
                                break;
                        case 3:s.opener(ifile,filename,ios::in);
                                cout<<"Searching based on USN number"<<endl;
                                flag=s.search();
                                if(flag==-1)
                                cout<<"Record not found"<<endl;
                                break;

                        case 4: s.opener(ifile,filename,ios::in | ios::out | ios::nocreate);
                                cout<<"To modify the record based on USN"<<endl;
                                flag=s.search();
                                if(flag==-1)
                                        cout<<"Record not found"<<endl;
                                else
                                        s.modify(flag);
                                break;
                        default:
                                exit(0);
                }
        ifile.close();
          }
}
```

Output :

```
1.- read  2- display      3 .-search      4.- modify      5.- exit
enter the choice:1
enter the student details
enter the usn number:100
enter the name:ajay
enter the age:30
enter the branch:ise
enter the sem:5

1.- read  2- display      3 .-search      4.- modify      5.- exit
enter the choice:1
enter the student details
enter the usn number:200
enter the name:suresh
enter the age:21
enter the branch:cse
enter the sem:6

1.- read  2- display      3 .-search      4.- modify      5.- exit
enter the choice:1
enter the student details
enter the usn number:300
enter the name:shashi
enter the age:20
enter the branch:me
enter the sem:2

1.- read  2- display      3 .-search      4.- modify      5.- exit
enter the choice:2
The student details are:
USN        NAME        AGE    BRANCH      SEM
100        ajay         30    ise          5
200        suresh       21    cse          6
300        shashi       20    me           2

1.- read  2- display      3 .-search      4.- modify      5.- exit
enter the choice:3
Searching based on USN number
enter the usn to be searched:200
USN:200
NAME:suresh
AGE:21
BRANCH:cse
SEM:6

1.- read  2- display      3 .-search      4.- modify      5.- exit
enter the choice:3
Searching based on USN number
enter the usn to be searched:125
Record not found

1.- read  2- display      3 .-search      4.- modify      5.- exit
enter the choice:4
To modify the record based on USN
```

enter the usn to be searched:300
USN:300
NAME:shashi
AGE:20
BRANCH:me
SEM:2
enter the usn number:450
enter the name:yadhu
enter the age:18
enter the branch:ece
enter the sem:2

1.- read 2- display 3 .-search 4.- modify 5.- exit
enter the choice:2
The student details are:

USN	NAME	AGE	BRANCH	SEM
100	ajay	30	ise	5
200	suresh	21	cse	6
450	yadhu	18	ece	2

1.- read 2- display 3 .-search 4.- modify 5.- exit
enter the choice:5

c:\tc\std2.txt

100|ajay|30|ise|5|**********
200|suresh|21|cse|6|**********
450|yadhu|18|ece|2|**********

Program 3

/* Write a program to read and write student objects with variable-Length records using any suitable record structures. Implemet pack (), unpack (), modify () and search () methods. */

```
#include<iostream.h>
#include<fstream.h>
#include<conio.h>
#include<stdio.h>
#include<iomanip.h>
#include<stdlib.h>
#include<string.h>

#define filename "std3.txt"

fstream ifile;

class student
{
       char usn[15],name[20],age[5],branch[6],sem[5];
      public:
            void opener(fstream&ifile,char *fn,int mode);
            void read();
            void pack();
            void display();
            void unpack();
            int search();
            void modify(int);
};

// function to open a file
void student::opener(fstream&sfile,char *fn,int mode)
{
     sfile.open(fn,mode);
     if(!sfile)
     {
            cout<<"unable to open a file"<<endl;
            getch();
            exit(1);
     }
}

//function to read the student record
void student::read()
{
cout<<"enter the usn number:";      gets(usn);
```

```
cout<<"enter the name:";            gets(name);
cout<<"enter the age:";       gets(age);
cout<<"enter the branch:";          gets(branch);
cout<<"enter the sem:";             gets(sem);
pack();

}
```

// function to pack the student record using delimiter

```
Voidstudent::pack()
{
char buffer[75];
strcpy(buffer,usn);                 strcat(buffer,"|");
strcat(buffer,name);                strcat(buffer,"|");
strcat(buffer,age);                 strcat(buffer,"|");
strcat(buffer,branch);        strcat(buffer,"|");
strcat(buffer,sem);           strcat(buffer,"|");
ifile<<buffer<<"#";
}
```

//function to display student record

```
void student::display()
 {
char buffer[75];
cout<<setiosflags(ios::left);
cout<<setw(15)<<"USN"<<setw(20)<<"NAME"<<setw(5)<<"AGE";
cout<<setw(10)<<"BRANCH"<<setw(5)<<"SEM"<<endl;
while(1)
  {
      unpack();
      if(ifile.eof())
      break;
      if(usn[0]!='$')
      {
           cout<<setw(15)<<usn<<setw(20)<<name<<setw(5)<<age;
           cout<<setw(10)<<branch<<setw(5)<<sem<<endl;
       }
   }
}
```

// function to unpack

```
void student::unpack()
{
      char dummy[75];
```

```
        ifile.getline(usn,15,'|');
        ifile.getline(name,20,'|');
        ifile.getline(age,5,'|');
        ifile.getline(branch,6,'|');
        ifile.getline(sem,5,'|');
        ifile.getline(dummy,10,'#');
}

//function to search student record based on USN.
int student::search()
{
        int flag;
        char susn[15];
        cout<<"enter the usn to be searched:";
        cin>>susn;
        while(!ifile.eof())
         {
                flag=ifile.tellg();
                unpack();
                if(usn[0]!='$' &&strcmp(usn,susn)==0)
                 {
        cout<<"USN:"<<usn<<"\n"<<"NAME:"<<name<<"\n"<<"AGE:"<<age;

        cout<<"\n"<<"BRANCH:"<<branch<<"\n"<<"SEM:"<<sem<<"\n";
                return flag;
                 }
         }
        return -1;
}
//function to modify record.
void student::modify(int recpos)
{
ifile.seekp(recpos,ios::beg);
ifile.put('$');
ifile.seekp(0,ios::end);
read();
}

//main program
void main()
{
        int ch,flag;
student s;
clrscr();
for(;;)
 {
```

```
cout<<endl<<"1.- read\t2- display\t 3 .-search\t4.- modify\t5.- exit"<<endl;
cout<<"enter the choice:";
cin>>ch;
switch(ch)
{
                    case 1: s.opener(ifile,filename,ios::app);
                            cout<<"enter the student details\n";
                            s.read();
                            break;
                    case 2: s.opener(ifile,filename,ios::in);
                            cout<<"The student details are:"<<endl;
                            s.display();
                            break;
                    case 3:s.opener(ifile,filename,ios::in);
                            cout<<"Searching based on USN number"<<endl;
                            flag=s.search();
                            if(flag==-1)
                                    cout<<"Record not found"<<endl;
                            break;

                    case        4:s.opener(ifile,filename,ios::in|ios::out        |
ios::nocreate);
                            cout<<"To modify the record based on USN"<<endl;
                            flag=s.search();
                            if(flag==-1)
                                    cout<<"Record not found"<<endl;
                            else
                                    s.modify(flag);
                            break;
        default:
                            exit(0);
                }
        ifile.close();
   }
}
```

Output

```
1.- read  2- display          3 .-search          4.- modify          5.- exit
enter the choice:1
enter the student details
enter the usn number:100
enter the name: amar
enter the age:20
enter the branch: ise
enter the sem:6

1.- read  2- display          3 .-search          4.- modify          5.- exit
```

```
enter the choice:1
enter the student details
enter the usn number:200
enter the name:chethan
enter the age:21
enter the branch:cse
enter the sem:7

1.- read  2- display        3 .-search        4.- modify        5.- exit
enter the choice:1
enter the student details
enter the usn number:300
enter the name:guru
enter the age:22
enter the branch:8
enter the sem:ece

1.- read  2- display        3 .-search        4.- modify        5.- exit
enter the choice:8

1.- read  2- display        3 .-search        4.- modify        5.- exit
enter the choice:1
enter the student details
enter the usn number:400
enter the name:krishna
enter the age:23
enter the branch:eee
enter the sem:6

1.- read  2- display        3 .-search        4.- modify        5.- exit
enter the choice:5

1.- read  2- display        3 .-search        4.- modify        5.- exit
enter the choice:2
The student details are:
USN         NAME        AGE        BRANCH        SEM
100         amar        20         ise           6
200         chethan     21         cse           7
300         guru        22         ece           8
400         krishna     23         eee           6

1.- read  2- display        3 .-search        4.- modify        5.- exit
enter the choice:3
Searching based on USN number
enter the usn to be searched:250
Record not found

1.- read  2- display        3 .-search        4.- modify        5.- exit
enter the choice:3
Searching based on USN number
enter the usn to be searched:200
USN:200
NAME:chethan
AGE:21
BRANCH:cse
SEM:7
```

```
1.- read  2- display        3 .-search        4.- modify        5.- exit
enter the choice:4
To modify the record based on USN
enter the usn to be searched:300
USN:300
NAME:guru
AGE:22
BRANCH:8
SEM:ece
enter the usnnumber:guruprasad
enter the name:20
enter the age:ece
enter the branch:7
enter the sem:2

1.- read  2- display        3 .-search        4.- modify        5.- exit
enter the choice:2
The student details are:
USN        NAME       AGE       BRANCH      SEM
100         amar        20         ise          6
200        chethan      21         cse          7
 300       guruprasad   20         ece          7
400         krishna     23         eee          6

1.- read  2- display        3 .-search        4.- modify        5.- exit
enter the choice:5

c:\tc\std3.txt
100|amar|20|ise|6|#200|chethan|21|cse|7|#guruprasad|20|ece|7|2|#400|krishna|23|eee|6|#
```

Program 4

/* Write a program to write student objects with Variable – Length records using any suitable record structure and to read from this file a student record using RRN. */

```
#include<iostream.h>
#include<fstream.h>
#include<conio.h>
#include<stdio.h>
#include<iomanip.h>
#include<stdlib.h>
#include<string.h>

#define filename "std4.txt"

fstream ifile;

class student
{
  char usn[15],name[20],age[5],branch[6],sem[5];
  public:
                void opener(fstream &sfile, char *fn , int mode);
                void read();
                void pack();
                void display();
                void unpack();
                int search();
};

void student:: opener(fstream &sfile, char *fn , int mode)
{
        sfile.open(fn,mode);
        if(!sfile)
        {
                cout<<"unable to open a file"<<endl;
                getch();
                exit(1);
        }
}
```

//function to read the student record

```
void student::read()
{
        cout<<"enter the usn number:";          gets(usn);
        cout<<"enter the name:";                gets(name);
        cout<<"enter the age:";                 gets(age);
```

```
        cout<<"enter the branch:";          gets(branch);
        cout<<"enter the sem:";             gets(sem);
        pack();
}


// function to pack the student record using delimiter
void student::pack()
{
        char buffer[75];
        strcpy(buffer,usn);         strcat(buffer,"|");
        strcat(buffer,name);        strcat(buffer,"|");
        strcat(buffer,age);         strcat(buffer,"|");
        strcat(buffer,branch);      strcat(buffer,"|");
        strcat(buffer,sem);         strcat(buffer,"|");
        ifile<<buffer<<"#";
}

//function to display student record
 void student::display()
 {
         int count=0;
        cout<<setiosflags(ios::left);
        cout<<setw(5)<<"RRN"<<setw(15)<<"USN"<<setw(20)<<"NAME"<<setw(5);
        cout<<"AGE"<<setw(10)<<"BRANCH"<<setw(5)<<"SEM"<<endl;
        while(1)
        {
                ifile.getline(usn,15,'|');
                 if(ifile.eof())
                        break;
                unpack();
                count++;
                cout<<setw(5)<<count<<setw(15)<<usn<<setw(20)<<name<<setw(5)<<age;
                cout<<setw(10)<<branch<<setw(5)<<sem<<endl;
        }
 }

// function to  unpack
void student::unpack()
{
        char dummy[75];
        ifile.getline(name,20,'|');
        ifile.getline(age,5,'|');
        ifile.getline(branch,6,'|');
        ifile.getline(sem,5,'|');
        ifile.getline(dummy,75,'#');}
```

```
//function to search student record based on rrn.
int student::search()
{
        int rrn,count=0;
        char dummy[75];
        cout<<"enter the rrn to be searched:";
        cin>>rrn;
        cout<<"RRN:"<<rrn;


       while(1)
       {
              ifile.getline(usn,15,'|');
              if(sfile.eof())
                     break;
              count++;
              if(rrn==count)
              {
                cout<<"\nRecord found\n";
                unpack();
                cout<<"USN:"<<usn<<"\n"<<"NAME:"<<name<<"\n"<<"AGE:"<<age;
                cout<<"\n"<<"BRANCH:"<<branch<<"\n"<<"SEM:"<<sem<<"\n";
                return 1;
               }
              else
                     ifile.getline(dummy,100,'#');
       }
       return -1;
}
// MAIN PROGRAM
void main()
{
       int ch,pos;
       student s;
       clrscr();
       for(;;)
       {
              cout<<endl<<"1.for read\t2.for display\t3.for search\t4.for exit\n";
              cout<<"Enter the choice:";
              cin>>ch;
              switch(ch)
              {
                 case 1: s.opener(ifile,filename,ios::app);
                           cout<<"enter the student details\n";
                           s.read();
                          break;
```

```
                case 2: s.opener(ifile,filename,ios::in);
                        cout<<"The student details are:"<<endl;
                        s.display();
                        break;
                case 3:s.opener(ifile,filename,ios::in);
                        cout<<"To search record based on Relative record number(RRN) \n";
                        pos=s.search();
                        if(pos==-1)
                            cout<<"\nRRN number is out of range-Record not found\n";
                       break;
              default:exit(0);
         }
        ifile.close();
  }
}
```

Output:

```
1.for read      2.for display    3.for search     4.for exit
Enter the choice:1
enter the student details
enter the usn number:100
enter the name:adi
enter the age:21
enter the branch:ise
enter the sem:5

1.for read      2.for display    3.for search     4.for exit
Enter the choice:1
enter the student details
enter the usn number:200
enter the name:arya
enter the age:20
enter the branch:cse
enter the sem:6

1.for read      2.for display    3.for search     4.for exit
Enter the choice:1
enter the student details
enter the usn number:300
enter the name:harsha
enter the age:19
enter the branch:me
enter the sem:6

1.for read      2.for display    3.for search     4.for exit
Enter the choice:2
The student details are:
```

RRN	USN	NAME	AGE	BRANCH	SEM
1	100	adi	21	ise	5
2	200	arya	20	cse	6
3	300	harsha	19	me	6

1.for read 2.for display 3.for search 4.for exit
Enter the choice:3
To search record based on Relative record number(RRN)
enter the rrn to be searched:RRN:2

Record found
USN:200
NAME:arya
AGE:20
BRANCH:cse
SEM:6

1.for read 2.for display 3.for search 4.for exit
Enter the choice:3
To search record based on Relative record number(RRN)
enter the rrn to be searched:RRN:50

RRN number is out of range-Record not found

1.for read 2.for display 3.for search 4.for exit
Enter the choice:4

C:\tc\std4.txt
100|adi|21|ise|5|#200|arya|20|cse|6|#300|harsha|19|me|6|#

Program-5

/* Write a program to implement simple index on primary key for a file of student objects. Implement add ( ), search ( ), delete ( ) using the index.*/

```
#include<iostream.h>
#include<fstream.h>
#include<conio.h>
#include<stdio.h>
#include<iomanip.h>
#include<stdlib.h>
#include<string.h>
#define max 10
#define datafile "student5.txt"
#define indexfile "index5.txt"
fstream stdfile, indfile;
int i,indsize;
char buffer[80];
class Student
{
        char dusn[15],name[20],age[5],branch[5],sem[5];
        public:
                void read();
                void pack();
                friend int search(char*);
                void recDisp(int);
                void remove(int);
                void dataDisp();
                void unpack();
};
class index
{
        public:
                char iusn[15],addr[5];
                void initial();
                void write();
}in,id[max];
void index::initial()
{
        indfile.open(indexfile,ios::in);
        if(!indfile)
        {
                indsize=0;
                return;
        }
        for(indsize=0;;indsize++)
        {
```

```
                indfile.getline(id[indsize].iusn,15,'|');
                indfile.getline(id[indsize].addr,5,'\n');
                if(indfile.eof())
                        break;
        }
        indfile.close();
}

// function to  open file

void opener(fstream &sfile,char* fn,int mode)
{
        sfile.open(fn,mode);
        if(!sfile)
        {
                cout<<"Unable to open the file\n";
                exit(1);
        }
}

// function to  write

void index::write()
{
        opener(indfile,indexfile,ios::out);
        for(i=0;i<indsize;i++)
                indfile<<id[i].iusn<<"|"<<id[i].addr<<"\n";
        indfile.close();
}

// function to  read

void Student::read()
{
        int k;
        cout<<"Enter the usn no.\n";
        gets(dusn);
        if(search(dusn)>=0)
        {
                cout<<"usn is already present,we can't add to index file\n";
                return;
        }
        for(i=indsize;i>0;i--)
        {
                if(strcmp(dusn,id[i-1].iusn)<0)
                        id[i]=id[i-1];
```

```
                else
                        break;
        }
        opener(stdfile,datafile,ios::app);
        cout<<"Enter the Name\n";
        gets(name);
        cout<<"Enter the age\n";
        gets(age);
        cout<<"Enter the branch\n";
        gets(branch);
        cout<<"Enter the semester\n";
        gets(sem);
        pack();
        stdfile.seekg(0,ios::end);
        k=stdfile.tellg();
        stdfile<<buffer<<endl;
        strcpy(id[i].iusn,dusn);
        itoa(k,id[i].addr,10);
        indsize++;
}
```

// function to pack

```
void Student::pack()
{
        strcpy(buffer,dusn); strcat(buffer,"|");
        strcat(buffer,name); strcat(buffer,"|");
        strcat(buffer,age); strcat(buffer,"|");
        strcat(buffer,branch); strcat(buffer,"|");
        strcat(buffer,sem); strcat(buffer,"|");
}

int search(char* fusn)
{
        int low=0,high=indsize-1;
        int mid;
        while(low<=high)
        {
                mid=(low+high)/2;
                if(strcmp(fusn,id[mid].iusn)==0)
                        return mid;
                else if(strcmp(fusn,id[mid].iusn)>0)
                        low=mid+1;
                else
                        high=mid-1;
        }
```

```
        return -1;
}

// function to  record display

void Student::recDisp(int pos)
{
        opener(stdfile,datafile,ios::in);
        stdfile.seekg(atoi(id[pos].addr),ios::beg);
        cout<<"The searched record details are:\n";
        cout<<setw(16)<<"USN"<<setw(16)<<"Name"<<setw(16)<<"Age"<<setw(16)
       <<"Branch"<<setw(16)<<"Sem"<<endl;
       unpack();
}

// function to  Remove

void Student::remove(int pos)
{
        opener(stdfile,datafile,ios::in|ios::out);
        stdfile.seekg(atoi(id[pos].addr),ios::beg);
        stdfile.put('$');
        for(i=pos;i<indsize;i++)
        id[i]=id[i+1];
        indsize--;
}

// function to   data display

void Student::dataDisp()
{
        cout<<setiosflags(ios::left);
        cout<<setw(16)<<"USN"<<setw(16)<<"Name"<<setw(16)<<"Age" \
        <<setw(16)<<"Branch"<<setw(16)<<"Sem"<<endl;
        while(1)
        {
                unpack();
                if(stdfile.eof())
                        break;
        }
}

// function to  unpack

void Student::unpack()
{
```

```
        stdfile.getline(buffer,100,'\n');
        i=0;
        if(buffer[i]!='$')
        {
                cout<<"\n";
                while(buffer[i]!='\0')
                {
                        if(buffer[i]=='|')
                                cout<<"\t\t";
                        else
                                cout<<buffer[i];
                                i++;
                }
        }
}

void main()
{
         int ch,pos,flag;
         char susn[15];
         Student S;
         in.initial();
         clrscr();
         for(;;)
         {
                cout<<endl<<"1.Read\n2.Display\n3.Search\n4.Delete\n5.exit\n";
                cin>>ch;
                switch(ch)
                {
                        case 1: cout<<"Enter student details\n";
                                S.read();
                                in.write();
                                break;
                        case 2: opener(stdfile,datafile,ios::in);
                                cout<<endl<<"Student Details\n";
                                S.dataDisp();
                                cout<<endl<<"Index file details are:\n";
                                cout<<setw(10)<<"USN"<<setw(10)<<"Address";
                                for(i=0;i<indsize;i++)
                                {

        cout<<endl<<setw(10)<<id[i].iusn<<setw(10)<<id[i].addr<<endl;
                                }
                                break;
                        case 3: cout<<"Enter the USN to be searched\n";
                                cin>>susn;
```

```
                        flag=search(susn);
                        if(flag==-1)
                                cout<<"Record Not found\n";
                        else
                                S.recDisp(flag);
                        break;
                case 4: cout<<"Enter the usn no to delete from the record\n";
                        cin>>susn;
                        pos=search(susn);
                        if(pos==-1)
                                cout<<"Usn No. not found\n";
                        else
                        {
                                S.remove(pos);
                                in.write();
                        }
                        break;
                default: exit(0);
        }
        stdfile.close();
    }
}
```

Output :

```
1 for read, 2 for display, 3 for search, 4 for delete, 5 for exit
1
enter student details :
enter the usn number=is101
enter the name=manoj kumar
enter the age=25
enter the branch=ise
enter the semeter=6

1 for read, 2 for display, 3 for search, 4 for delete, 5 for exit
2

the student details are
usn     name            age  branch sem
is101    manoj kumar      25   ise    6

the index file details are
usn     address
is101    0

1 for read, 2 for display, 3 for search, 4 for delete, 5 for exit
1
enter student details :
enter the usn number=cs201
enter the name=vikram narayan
```

```
enter the age=35
enter the branch=cse
enter the semeter=8

1 for read, 2 for display, 3 for search, 4 for delete, 5 for exit
2

the student details are
usn      name              age  branch sem
is101     manoj kumar       25   ise      6
cs201     vikram narayan     35   cse      8


the index file details are
usn      address
cs201     29
is101     0

1 for read, 2 for display, 3 for search, 4 for delete, 5 for exit
1
enter student details :
enter the usn number=me301
enter the name=pradeep
enter the age=24
enter the branch=mec
enter the semeter=7

1 for read, 2 for display, 3 for search, 4 for delete, 5 for exit
1
enter student details :
enter the usn number=ee401
enter the name=shruthi
enter the age=20
enter the branch=eee
enter the semeter=6

1 for read, 2 for display, 3 for search, 4 for delete, 5 for exit
1
enter student details :
enter the usn number=it501
enter the name=sunitha
enter the age=it
enter the branch=it
enter the semeter=6

1 for read, 2 for display, 3 for search, 4 for delete, 5 for exit
2

the student details are
usn            name          age  branch  sem
is101     manoj kumar       25      ise     6
cs201     vikram narayan    35      cse     8
me301      pradeep          24      mec     7
ee401      shruthi          20      eee     6
it501      sunitha          21      it      6
```

```
the index file details are
usn      address
cs201     29
ee401     86
is101     0
it501     111
me301     61


1 for read, 2 for display, 3 for search, 4 for delete, 5 for exit
3
enter usn number to search : me301
usn = me301
name = pradeep
age = 24
branch = mec
sem = 7

1 for read, 2 for display, 3 for search, 4 for delete, 5 for exit
3
enter usn number to search : ec250
usn number record not found for search

1 for read, 2 for display, 3 for search, 4 for delete, 5 for exit
4
enter usn number to delete the record : me301
1 for read, 2 for display, 3 for search, 4 for delete, 5 for exit
2

the student details are
usn      name              age  branch sem
is101     manoj kumar       25   ise       6
cs201     vikram narayan    35   cse       8
ee401     shruthi           20   eee       6
it501     sunitha           21   it        6

the index file details are
usn      address
cs201     29
ee401     86
is101     0
it501     111

1 for read, 2 for display, 3 for search, 4 for delete, 5 for exit
4
enter usn number to delete the record : cv105
usn number not found to delete

1 for read, 2 for display, 3 for search, 4 for delete, 5 for exit
5

C:\tc\typestd5.txt

is101|manoj kumar|25|ise|6|
cs201|vikram narayan|35|cse|8|
$e301|pradeep|24|mec|7|
```

```
ee401|shruthi|20|eee|6|
it501|sunitha|21|it|6|

c:\tc\tyoe index5.txt

cs201|29
ee401|86
is101|0
it501|111
```


Program 6

/* Write a program to implement index on secondary key, the name, for a file of student objects. Implement add(), search(), delete () using the secondary index. */

```
#include<fstream.h>
#include<conio.h>
#include<stdio.h>
#include<iomanip.h>
#include<stdlib.h>
#include<string.h>
#define datafile "stud6.txt"
#define indexfile "pri6.txt"
#define sindexfile "sec6.txt"

fstream dfile,ifile,sifile;
int i,indsize,sindsize;
char buffer[100],skey[20];

//function to open
void opener(fstream &file, char *fn,int mode)
{
        file.open(fn,mode);
        if(!file)
        {
                cout<<"unable to open a file";
                getch();
                exit(1);
        }
}
class student
{
        char dusn[15],name[20],age[5],branch[6],sem[5];
        public:
                void read();
                void pack();
                friend int search(char *);
                void remove();
                void datadisp();
                void unpack();
};s;
class index
{
        public:
                char iusn[15],addr[5];
                void initial();
                void write();
}id[50],in;
```

```
class sindex
{
        public :
                char sname[20],susn[15];
                void sinitial();
                void swrite();
}sid[50],sin;


// function to copy index file to array structure
void index::initial()
{
        ifile.open(indexfile,ios::in);
        if(!ifile)
        {
                indsize=0;
                return;
        }
  for(indsize=0;;indsize++)
  {
                ifile.getline(id[indsize].iusn,15,'|');
                ifile.getline(id[indsize].addr,5,'\n');
                if(ifile.eof())
                        break;
  }
  ifile.close();
}

//function to copy sindex file to array structure
void sindex::sinitial()
{
        sifile.open(sindexfile,ios::in);
        if(!sifile)
        {
                sindsize=0;
                return;
        }
  for(sindsize=0;;sindsize++)
  {
                sifile.getline(sid[sindsize].sname,20,'|');
                sifile.getline(sid[sindsize].susn,15,'\n');
                if(sifile.eof())
                        break;
  }
  sifile.close();
```

```
}

// function to update the index file
void index::write()
{
        opener(ifile,indexfile,ios::out);
        for(i=0;i<indsize;i++)
                ifile<<id[i].iusn<<"|"<<id[i].addr<<"\n";
}

//function to upadate the secondary file
void sindex::swrite()
{
        opener(sifile,sindexfile,ios::out);
        for(i=0;i<sindsize;i++)
                sifile<<sid[i].sname<<"|"<<sid[i].susn<<"\n";
}


// function to read the student record
void student::read()
{
        int k;
        cout<<"enter the usn number=";  gets(dusn);
        if(search(dusn)>=0)
        {
                cout<<"usn is already present we can't add to index file\n";
                return;
        }
        for(i=indsize;i>0;i--)
        {
          if(strcmp(dusn,id[i-1].iusn)<0)
                        id[i]=id[i-1];
          else
                        break;
        }
        opener(dfile,datafile,ios::app);
        cout<<"enter the name=";      gets(name);
        cout<<"enter the age=";       gets(age);
        cout<<"enter the branch=";    gets(branch);
        cout<<"enter the semester=";          gets(sem);
        pack();
        dfile.seekg(0,ios::end);
        k=dfile.tellg();
        dfile<<buffer<<"\n";
        strcpy(id[i].iusn,dusn);
```

```
        itoa(k,id[i].addr,10);
        indsize++;
        for(i=sindsize;i>0;i--)
        {
                if(strcmp(name,sid[i-1].sname)<0)
                        sid[i]=sid[i-1];
                else if((strcmp(name,sid[i-1].sname)==0) && (strcmp(dusn,sid[i-1].susn)<0))
                                sid[i]=sid[i-1];
                else
                                break;
        }
        strcpy(sid[i].sname,name);
        strcpy(sid[i].susn,dusn);
        sindsize++;
}

//function to pack
void student::pack()
{
                strcpy(buffer,dusn);    strcat(buffer,"|");
                strcat(buffer,name);    strcat(buffer,"|");
                strcat(buffer,age);     strcat(buffer,"|");
                strcat(buffer,branch);  strcat(buffer,"|");
                strcat(buffer,sem);     strcat(buffer,"|");
}



//function to search based on usn number
int search(char * fusn)
{
        int low=0,high=indsize-1,mid;
        while(low <=high)
        {
                mid = (low+high)/2;
                if(strcmp(fusn,id[mid].iusn)==0)
                        return mid;
                if(strcmp(fusn,id[mid].iusn)>0)
                        low=mid+1;
                else
                        high=mid-1;
        }
        return -1;
}

//function to search based on secondary key
```

```
int sec_search()
{
        int pos,j,flag=-1;
        cout<<"\nenter the name to search(sec key):";
        gets(skey);
        cout<<"the searched record details are :"<<endl;
        cout<<setiosflags(ios::left);
        cout<<"usn"<<"\t\tname"<<endl;
        opener(dfile,datafile,ios::in|ios::out);
        for(j=0;j<sindsize;j++)
                if(strcmp(skey,sid[j].sname)==0)
                {
                        cout<<sid[j].susn<<"\t\t"<<sid[j].sname<<endl;
                        flag=j;
                }
        return flag;
}

// function to remove the record
void student::remove()
{
        char rusn[10];
        int pos,spos;
        cout<<"enter the usn number above listed to delete:";
        cin>>rusn;
        for(i=0;i<sindsize;i++)
        {
                if(strcmp(sid[i].susn,rusn)==0)
                {
                        spos=i;
                        break;
                }
        }

        if(strcmp(sid[spos].sname,skey)==0)
        {
                pos=search(rusn);
                dfile.seekp(atoi(id[pos].addr),ios::beg);
                dfile.put('$');
                for(i=pos;i<indsize;i++)
                        id[i]=id[i+1];
                indsize--;
                for(i=spos;i<sindsize;i++)
                        sid[i]=sid[i+1];
                sindsize--;
        }
```

```
        else
                cout<<"usn number and name doesnot match";
}
```

//function to display the datafile

```
void student::datadisp()
{
        cout<<setiosflags(ios::left);
        cout<<setw(16)<<"usn"<<setw(16)<<"name"<<setw(16)<<"age"<<setw(16);
        cout<<"branch"<<setw(16)<<"sem";
        while(1)
        {
                unpack();
                if(dfile.eof())
                break;
        }
        cout<<endl<<"the index file details are "<<endl;
        cout<<setw(10)<<"usn"<<setw(10)<<"address";
        for(i=0;i<indsize;i++)
                 cout<<endl<<setw(10)<<id[i].iusn<<setw(10)<<id[i].addr;
        cout<<endl<<"\n the secondary file details are " <<endl;
        cout<<setw(20)<<"name"<<setw(15)<<"primary reference";
        for(i=0;i<sindsize;i++)
                 cout<<endl<<setw(20)<<sid[i].sname<<setw(15)<<sid[i].susn;
}
```

//function to unpack the data file

```
void student::unpack()
{
         dfile.getline(buffer,100,'\n');
         i=0;
         if(buffer[i]!='$')
                 while(buffer[i]!='\0')
                 {
                         if(buffer[i]=='|')
                                 cout<<"\t\t";
                         else
                                 cout<<buffer[i];
                         i++;
                 }
}
```

//main program

```
void main()
{
        int ch,flag;
        in.initial();
        sin.sinitial();
        clrscr();
        for(;;)
        {
        cout<<endl<<"1-read,2-display,3-search,4-delete,5-exit\n";
        cin>>ch;
        switch(ch)
        {
                case 1: cout<<endl<<"enter student details : " <<endl;
                        s.read();
                        in.write();
                        sin.swrite();
                        break;
                case 2: opener(dfile,datafile,ios::in);
                          cout<<"\nthe datafile,indexfile and secondary file" <<endl;
                          s.datadisp();
                          break;
                case 3:cout<<"To search based on  sec key ";
                        flag=sec_search();
                        if(flag==-1)
                                cout<<"no data record ";
                        break;
                case 4: flag=sec_search();
                        if(flag==-1)
                                cout<<"no data record found";
                        else
                        {
                                s.remove();
                                in.write();
                                sin.swrite();
                         }
                        break;
                default :  exit(0);
          }
          dfile.close();
          ifile.close();
          sifile.close();
        }
}
```

Output

```
1. read, 2. display, 3. search, 4 .delete, 5 .exit
1
enter student details :
enter the usn number=is101
enter the name=manoj kumar
enter the age=25
enter the branch=ise
enter the semeter=6

 1. read, 2. display, 3. search, 4 .delete, 5 .exit
 2

the student details are
usn      name              age  branch sem
is101     manoj kumar       25   ise    6

the index file details are
usn      address
is101     0

 1. read, 2. display, 3. search, 4 .delete, 5 .exit
 1
enter student details :
enter the usn number=cs201
enter the name=vikram narayan
enter the age=35
enter the branch=cse
enter the semeter=8

1. read, 2. display, 3. search, 4 .delete, 5 .exit
2
the datafile,indexfile and secondary file
the student details are
usn      name              age  branch sem
is101     manoj kumar       25   ise     6
cs201     vikram narayan     35   cse     8


the index file details are
usn      address
cs201     29
is101     0

the secondary file details are

name        usn
is101       manoj kumar
cs101       vikram narayan


1.read, 2. display, 3. search, 4 .delete, 5 .exit
3
To search based on sec key
Enter the name to search (sec key):vikram narayan

usn        name              age    branch   sem
```

```
cs201    vikram narayan     35   cse       8

 1. read, 2. display, 3. search, 4 .delete, 5 .exit
   4

Enter the name to search (sec key):vikram narayan

usn       name              age   branch   sem
cs201    vikram narayan     35   cse       8

enter the usn number above listed to delete
cs201
```


Program 7

/* Write a program to read two lists of names and then match the names in the two lists using consequential Match based on a single loop. Output the names common to both the files */

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<fstream.h>
#include<string.h>

//function to open a file in different mode
void opener(fstream &file,char *fn,int mode)
{
	file.open(fn,mode);
	if(!file)
	{
		cout<<"unable to open the file\n";
		getch();
		exit(1);
	}
}

// function to match the common names from two files
void match(fstream &file1,fstream &file2, fstream &ofile)
{
	char s1[25],s2[25];
	file1.getline(s1,25,'\n');
	file2.getline(s2,25,'\n');
	while(!file1.eof() && !file2.eof())
	{
		if(strcmp(s1,s2)==0)
		{
			ofile<<s1<<"\n";
			cout<<s1<<endl;
			file1.getline(s1,25,'\n');
			file2.getline(s2,25,'\n');
		}
	  else if(strcmp(s1,s2)<0)
			file1.getline(s1,25,'\n');
	  else
			file2.getline(s2,25,'\n');
	}
}

// main program
```

```
void main()
{
	fstream list1,list2, outlist;
	clrscr();
	opener(list1,"name1.txt",ios::in);
	opener(list2,"name2.txt",ios::in);
	opener(outlist, "names.txt", ios::out);
	match(list1,list2,outlist);
	cout<<"name1.txt & name2.txt matching names in names.txt \n";
	list1.close();
	list2.close();
	outlist.close();
	getch();
}
```

NOTE: In file name1.txt and name2.txt
Names should be in ascending order

OUTPUT:

names1.txt

navnish
pavan
sharath
sagar
vallish

names2.txt

navnish
niranjan
pavan
puneeth
sharath
sagar

output.txt

name1.txt & name2.txt matching names in names.txt
navnish
pavan
sharath
sagar

Program 8

/*Write a program to read k Lists of names and merge them using k-way merge algorithm with k = 8. */

```
#include<stdio.h>
#include<conio.h>
#include<iostream.h>
#include<fstream.h>
#include<stdlib.h>
#include<string.h>
#define k 8

// function to open a file in different mode
void opener(fstream &file,char *fn,int mode)
{
        file.open(fn,mode);
        if(!file)
        {
                cout<<"unable to open the file\n";
                getch();
                exit(1);
        }
}

//main program
void main()
{
   fstream list[8],outfile;
   char name[8][20]={"name0.txt", "name1.txt","name2.txt","name3.txt",
                     "name4.txt", "name5.txt","name6.txt","name7.txt"};
   char item[8][20] , min[20]="";
   int i,count=0;
   for(i=0;i<k;i++)
        opener(list[i],name[i],ios::in);
        opener(outfile,"merge8.txt",ios::out);
   for(i=0;i<k;i++)
   {
        list[i].getline(item[i],20,'\n');
        if(list[i].eof())

        count++;
   }
   cout<<"the names after merging using k-way merge algorithm\n";

while(count < k)
```

```
    {
        strcpy(min,"");
        for(i=0;i<k;i++)
            if(!list[i].eof())
            {
                strcpy(min,item[i]);
                break;
            }
        count=0;
        for(i=0;i<k;i++)
        {
            if(list[i].eof())
            count++;
            else if(strcmp(item[i],min)<0)
            strcpy(min,item[i]);
        }
        if(count==8)    break;
        outfile<<min<<"\n";
        cout<<min<<"\n";
        for(i=0;i<k;i++)
            if(strcmp(item[i],min)==0)
            list[i].getline(item[i],20,'\n');
    }
    for(i=0;i<8;i++)
        list[i].close();
    getch();
}
```

OUTPUT

Name0.txt
Akarsh
*

Name2.txt
Navnish
Pavan
*

Name3.txt
Navnish
Sharath
*

Name4.txt
Sharath
Srinidhi
*

Name5.txt

Srinidhi
Vallish
*

Name6.txt
Vallish
*

Name7.txt
*

Merge8.txt
Akarsh
Anurag
Navnish
Pavan
Sharath
Srinidhi
Vallish