#include<iostream>

#include<fstream>

#include<curses.h>

#include<stdio.h>

#include<iomanip>

#include<stdlib.h>

#include<string.h>

using namespace std;

#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();

}

void opener(fstream &sfile,char\* fn,ios\_base::openmode mode)

{

sfile.open(fn,mode);

if(!sfile)

{

cout<<"Unable to open the file\n";

exit(1);

}

}

void index::write()

{

opener(indfile,indexfile,ios::out);

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

indfile<<id[i].iusn<<"|"<<id[i].addr<<"\n";

indfile.close();

}

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;

}

void Student::read()

{

cout<<"Enter the usn no.\n";

scanf("%s",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";

scanf("%s",name);

cout<<"Enter the age\n";

scanf("%s",age);

cout<<"Enter the branch\n";

scanf("%s",branch);

cout<<"Enter the semester\n";

scanf("%s",sem);

pack();

stdfile.seekg(0,ios::end);

int k=stdfile.tellg();

stdfile<<buffer<<endl;

strcpy(id[i].iusn,dusn);

sprintf(id[i].addr,"%d",k);

indsize++;

}

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,"|");

}

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();

}

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;

}

}

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++;

}

}

}

int main()

{

int ch,pos,flag;

char susn[15];

Student S;

in.initial();

curscr;

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();

}

return 0;

}