|  |
| --- |
|  |
| ONLINE EXAMINATION MANAGEMENT SYSTEM |
|  |

|  |
| --- |
| **Vignesh P R** |

**ONLINE EXAMINATION MANAGEMENT SYSTEM**

**Aim**

The program has its main aim to provide an effective tool for schools and colleges to conduct online examination.

It also helps the student know their status after attempting the question paper by displaying their result without any delay.

**Working**

The program allows the student to enter/click answers and also prints their result when the student has submitted or when time given is over, without any delay.

The project involves mouse which is enabled for the convenience of the student as it serves as an alternate for keyboard input.

**Some important functions:**

1. *callmouse()*

Purpose: Calls mouse interrupt.

1. *mouseposi(int\*,int\*,int\*);*

Purpose: The function takes in address of 3 integers as parameter and assigns the value of x & y co-ordinates of the present location of the mouse to them. The third parameter is given value depending on the click status (0 if no button is clicked and 1 if left button is clicked)

1. *getpassword(char\*)*

Purpose: Accepts password from user (using getch) and sets it in the string passed as its parameter.

1. *noquestion1()*

Purpose: Returns number of single correct questions in the file "Question.dat".

1. *noquestion2()*

Purpose: Returns number of multi-correct questions in the file "multicor.dat".

1. *nostudent()*

Purpose: Returns the number of students who have registered from the file

"Student.dat".

1. *insertspace(int)*

Purpose: Prints a given number of spaces which is passed as parameter.

1. *insertline(int)*

Purpose: Prints a given number of lines which is passed as parameter.

1. *back(int)*

Purpose: Moves the cursor back by a given number passed as its parameter.

1. *backspace(int)*

Purpose: Moves the cursor back and also prints space by a given number passed as its pa Moves the cursor back by a given number passed as its parameter.

**User Types**

1. Admin (super user): The admin can
2. alter/delete/add the questions in it and the answer key before the exam starts
3. view the result of any student.
4. alter the duration of the test before anyone takes up the test.
5. remove any student from the list.
6. Student (user): The student can
7. answer the questions within the allotted time.
8. can modify his/her details any time before the test.
9. can view his/her result after finishing the test answers submitting their answers.

**Classes used**

1. *Question\_s (also named as Questionsc or Question)*

Data members:

1. An integer to store question number.
2. Four 1-D character arrays to store the options for the question.
3. A string to store the question.
4. A character to store the correct answer for the question.

Purpose: This class stores the questions and answers for each single choice (single option correct) question in order to read from/write into the binary file.

1. *Question\_m (also named as Questionmc or Question1)*

Data members:

1. An integer to store question number.
2. A string to store the question.
3. Four 1-D character arrays to store the options for the question.
4. A character variable which stores a unique character corresponding to the option/s which are true for the answer.

Purpose: This class stores questions and answers for each multiple choice (more than one option correct) question in order to read from/write into the binary file.

The character variable used to store the correct answer/s for the question stores a unique character for each different set of answers.

The logic behind it is as follows:

For eg. If the correct answers are options 'a' and 'd’. Then it will store the character corresponding to the binary value 00001001(assuming 8bits).

So, the fifth place is '1' if option 'a' is correct, sixth place is '1' if option 'b' is correct seventh place is '1' if option 'c' is correct eighth place is '1' if option 'd' is correct.

In the above example if 'a' and 'd' are entered the variable stores the character corresponding to the ASCII value '9'(2^3+2^0).

**Files used**

1. Name: Question.dat

Type: Binary

Purpose: Stores questions, options and answers for all single choice (single option correct) questions.

More precisely it stores objects of type 'Question\_s'.

1. Name: multicor.dat

Type: Binary

Purpose: Stores questions, options and answers for all multiple choice (more than one option correct) questions.

More precisely it stores objects of type 'Question\_m'.

1. Name: Student.dat

Type: Binary

Purpose: Stores the details of each student (name, gender, roll no, contact no, marks etc.).

More precisely it stores objects of type 'Student'.

1. Name: instruct.txt

Type: Text

Purpose: Stores general instructions for the students for the examination.

1. Name: Time.dat

Type: binary

Purpose: Stores the duration of exam in seconds.

1. Name: Admin.dat

Type: Binary

Purpose: Stores the password.

**Source code/s:**

The project consists of 6 "cpp" files combined to form a "prj" file. By creating a “.prj"file the functions from from a file can be used in the others by simply writing "extern <function prototype>" in that file.

The 6 "cpp" files are:

1)ADMIN.cpp

2)LIBRARY.cpp

3)MOUSEFUN.cpp

4)STUDENT.cpp

5)WORK1.cpp

6)WORK2.cpp

1)ADMIN.cpp

This file contains the functions required/used by the admin such as altering/deleting/adding questions to the exam and many more.

#include<iostream.h>

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<conio.h>

#include<dos.h>

#include<fstream.h>

extern insertspace(int),noquestion1(),noquestion2(),getpassword(char\*),insertline(int),back(int),backspace(int) ;

extern nostudent();

struct Student1

{

char name[21],username[21],pass[21],contactno[10],gender;

int age,marks[3],std,rno,attendance;

Student1();

void getpass();

void accept();

void display();

void comprusername(char A[]);

void checkpass(char D[]);

};

Student1::Student1()

{

for(int y=0;y<strlen(name);y++)

name[y]='\0';

for(int i=0;i<10;i++)

{

gender='M';

contactno[i]=0;

}

for(int u=0;u<3;u++)

marks[u]=0;

attendance=rno=0;

}

void Student1:: display()

{

cout<<"Name: "<< name <<endl

<<"Gender: "<< gender <<endl

<<"Age: "<< age <<endl

<<"Username: "<< username <<endl;

}

class questionmc

{

char q[1000],a[20],b[20],c[20],d[20],ans;

public:

questionmc();

void accept();

void display();

int ansret();

};

void questionmc::questionmc()

{

for(int y=0;y<1000;y++)

q[y]=32;

for(int j=0;j<20;j++)

a[j]=b[j]=c[j]=d[j]=32;

ans=32;

}

void questionmc::accept()

{

cout<<endl<<"Enter the question: ";

gets(q);

cout<<endl<<"Enter option A: ";

gets(a);

cout<<endl<<"Enter option B: ";

gets(b);

cout<<endl<<"Enter option C: ";

gets(c);

cout<<endl<<"Enter option D: ";

gets(d);

char c='\0';

int x[5];

for(int i=0;i<5;i++)

x[i]=0;

cout<<"Is A a correct answer to this question?(Y/N)"<<endl;

cin>>c;

if(c=='y' ||c=='Y')

x[1]=1;

else;

cout<<"Is B a correct answer to this question?(Y/N)"<<endl;

cin>>c;

if(c=='y' ||c=='Y')

x[2]=1;

else;

cout<<"Is C a correct answer to this question?(Y/N)"<<endl;

cin>>c;

if(c=='y' ||c=='Y')

x[3]=1;

else;

cout<<"Is D a correct answer to this question?(Y/N)"<<endl;

cin>>c;

if(c=='y' ||c=='Y')

x[4]=1;

else;

ans=((x[1]\*8)+(x[2]\*4)+(x[3]\*2)+(x[4]));

}

void questionmc::display()

{

gotoxy(20,6);

puts(q);

insertline(2);

insertspace(13);

cout<<"A)( )"<<a <<" "<<endl;

insertspace(13);

cout<<"B)( )"<<b <<" "<<endl;

insertspace(13);

cout<<"C)( )"<<c<<" "<<endl;

insertspace(13);

cout<<"D)( )"<<d<<" "<<endl;

insertspace(13);

cout<<" ( )Clear response";

}

int questionmc:: ansret()

{

return ans;

}

class questionsc

{

char q[1000],a[20],b[20],c[20],d[20],ans;

public:

questionsc();

void display();

void accept();

char ansret();

};

void questionsc::questionsc()

{

for(int i=0;i<1000;i++)

q[i]=32;

for(int y=0;y<20;y++)

a[y]=b[y]=c[y]=d[y]=32;

ans=32;

}

void questionsc::accept()

{

cout<<endl<<"Enter the question: ";

gets(q);

cout<<endl<<"Enter option A: ";

gets(a);

cout<<endl<<"Enter option B: ";

gets(b);

cout<<endl<<"Enter option C: ";

cin>>c;

cout<<endl<<"Enter option D: ";

gets(d);

cout<<endl<<"Enter correct answer: ";

delay(100);

cin>>ans;

}

void questionsc::display()

{

gotoxy(20,6);

cout<<q;

insertline(3);

insertspace(13);

cout<<"A)( )"<<a <<endl;

insertspace(13);

cout<<"B)( )"<<b <<endl;

insertspace(13);

cout<<"C)( )"<<c<<endl;

insertspace(13);

cout<<"D)( )"<<d<<endl;

insertspace(13);

cout<<" ( )Clear response";

}

char questionsc:: ansret()

{

return ans;

}

void addquestion()

{

cout<<"Do you wish to add question in single correct or multi-correct(s/m) ";

char v;

cin>>v;

if(v=='s' || v=='S')

{

fstream fout("Question.dat",ios::out| ios::app);

questionsc l;

l.accept();

fout.write((char\*)&l,sizeof(l));

}

else if(v=='m' || v=='M')

{

fstream fout("multicor.dat",ios::out| ios::app);

questionmc l;

l.accept();

fout.write((char\*)&l,sizeof(l));

}

else;

}

void alter()

{

char k;

do

{

clrscr();

cout<<"Do you wish to alter"

<<" single correct or multicorrect question(s/m): ";

char u;

cin>>u;

if(u=='S'||u=='s'||u=='m'||u=='M')

{

int r;

cout<<"Enter the question number you wish to alter: ";

cin>>r;

int e=0;

while(e==0)

{

if(u=='s' || u=='S')

{

if(r<=noquestion1())

e++;

}

else if(u=='m' || u=='M')

{

if(r<=noquestion2())

e++;

}

if(e==0)

{

cout<<"Invalid question number!Please re-enter: ";

cin>>r;

}

}

fstream f;

fstream fout("Temp.dat",ios::binary | ios::out);

if(u=='s' || u=='S')

{

f.open("Question.dat",ios::binary | ios::in);

questionsc s,w;

f.seekg(sizeof(s)\*(r-1));

f.read((char\*)&s,sizeof(s));

f.close();

fstream fin("Question.dat",ios::binary | ios::in);

s.display();

gotoxy(20,20);

cout<<"Is this the question you wish to alter(Y/N): ";

char c;

cin>>c;

if(c=='y' || c=='Y')

{

clrscr();

for(int i=1;i<r;i++)

{

questionsc q;

fin.read((char\*)&q,sizeof(q));

fout.write((char\*)&q,sizeof(q));

}

w.accept();

delay(100);

fout.write((char\*)&w,sizeof(w));

fin.read((char\*)&s,sizeof(s));

for(int y=(r+1);y<=noquestion1();y++)

{

questionsc q;

fin.read((char\*)&q,sizeof(q));

fout.write((char\*)&q,sizeof(q));

}

fin.close();

fout.close();

int p=noquestion1();

fstream f1("Question.dat", ios::binary | ios::out);

fstream f2("Temp.dat",ios::binary | ios::in);

for(int x=1;x<=p;x++)

{

questionsc q;

f2.read((char\*)&q,sizeof(q));

f1.write((char\*)&q,sizeof(q));

}

f1.close();

f2.close();

cout<<"SUCCESSFULLY ALTERED"<<endl;

}

}

else if(u=='m'|| u=='M')

{

f.open("multicor.dat",ios::binary | ios::in);

clrscr();

questionmc s,w;

f.seekg(sizeof(s)\*(r-1));

f.read((char\*)&s,sizeof(s));

f.close();

s.display();

gotoxy(20,20);

cout<<"Is this the question you wish to alter(Y/N): ";

char c;

cin>>c;

if(c=='y' || c=='Y')

{

fstream fin("multicor.dat",ios::binary | ios::in);

fstream fout("Temp.dat",ios::binary | ios::out);

for(int h=1;h<r;h++)

{

questionmc q;

fin.read((char\*)&q,sizeof(q));

fout.write((char\*)&q,sizeof(q));

}

w.accept();

fout.write((char\*)&w,sizeof(w));

fin.read((char\*)&s,sizeof(s));

for(int j=(r+1);j<=noquestion2();j++)

{

questionmc q;

fin.read((char\*)&q,sizeof(q));

fout.write((char\*)&q,sizeof(q));

}

fin.close();

fout.close();

int g=noquestion2();

fstream f1("multicor.dat", ios::binary | ios::out);

fstream f2("Temp.dat",ios::binary | ios::in);

for(int y=1;y<=g;y++)

{

questionmc q;

f2.read((char\*)&q,sizeof(q));

f1.write((char\*)&q,sizeof(q));

}

f1.close();

f2.close();

cout<<"SUCCESSFULLY ALTERED"<<endl;

}

}

delay(100);

cout<<"Do you wish to alter any other question(Y/N)? ";

cin>>k;

}

else

k='n';

}while(k=='Y' || k=='y');

}

void passchange()

{

clrscr();

cout<<"Enter your new password: ";

char g[21];

getpassword(g);

cout<<endl<<"Please re-enter your password ";

char e[21];

getpassword(e);

while(strcmp(g,e)!=0)

{

insertspace(10);

cout<<"Password does not match";

back(34);

backspace(strlen(e));

getpassword(e);

}

fstream fout("Admin.dat",ios::binary | ios::out);

fout.write((char\*)&e,sizeof(e));

}

void viewmarks()

{

clrscr();

int a=0;

int l=0;

int m1,m2;

Student1 s;

while(a==0)

{

fstream fin("Student.dat",ios::binary |ios::in);

gotoxy(1,1);

char user[21];

cout<<"Enter username: ";

gotoxy(l+16,1);

backspace(l);

cin>>user;

user[strlen(user)]='\0';

int k=0;

int h=0;

while(h<nostudent() && k==0)

{

fin.read((char\*)&s,sizeof(s));

if(strcmp(user,s.username)==0)

k=1;

h++;

}

if(k==0)

{

clrscr();

cout<<endl<<"Invalid username!!";

l=strlen(user);

a=0;

}

else

a=1;

}

cout<<"Total marks: "<< s.marks[2]<<endl

<<"Marks in single correct: "<< s.marks[0] <<endl

<<"Marks in multicorrect: "<< s.marks[1] <<endl;

getch();

}

void deletequestion()

{

char k;

do

{

clrscr();

cout<<"Do you wish to delete"

<<" single correct or multicorrect question(s/m): ";

char u;

cin>>u;

if(u=='S'||u=='s'||u=='m'||u=='M')

{

int r;

cout<<"Enter the question number you wish to delete: ";

cin>>r;

int e=0;

while(e==0)

{

if(u=='s' || u=='S')

{

if(r<=noquestion1() && r>0)

e++;

}

else if(u=='m' || u=='M')

{

if(r<=noquestion2() && r>0)

e++;

}

if(e==0)

{

cout<<"Invalid question number!Please re-enter: ";

cin>>r;

}

}

fstream f;

fstream fout("Temp.dat",ios::binary | ios::out);

if(u=='s' || u=='S')

{

f.open("Question.dat",ios::binary | ios::in);

questionsc s,w;

f.seekg(sizeof(s)\*(r-1));

f.read((char\*)&s,sizeof(s));

f.close();

s.display();

gotoxy(20,20);

cout<<"Is this the question you wish to delete(Y/N): ";

char c;

cin>>c;

if(c=='y' || c=='Y')

{

fstream fin("Question.dat",ios::binary | ios::in);

clrscr();

for(int i=1;i<r;i++)

{

questionsc q;

fin.read((char\*)&q,sizeof(q));

fout.write((char\*)&q,sizeof(q));

}

questionsc q1;

delay(100);

fin.read((char\*)&q1,sizeof(q1));

for(int y=(r+1);y<=(noquestion1()-1);y++)

{

questionsc q;

fin.read((char\*)&q,sizeof(q));

fout.write((char\*)&q,sizeof(q));

}

fin.close();

fout.close();

int p=noquestion1();

fstream f1("Question.dat", ios::binary | ios::out);

fstream f2("Temp.dat",ios::binary | ios::in);

for(int x=1;x<=(p-1);x++)

{

questionsc q;

f2.read((char\*)&q,sizeof(q));

f1.write((char\*)&q,sizeof(q));

}

f1.close();

f2.close();

cout<<"SUCCESSFULLY DELETED"<<endl;

}

}

else if(u=='m'|| u=='M')

{

f.open("multicor.dat",ios::binary | ios::in);

clrscr();

questionmc s,w;

f.seekg(sizeof(s)\*(r-1));

f.read((char\*)&s,sizeof(s));

f.close();

s.display();

gotoxy(20,20);

cout<<"Is this the question you wish to alter(Y/N): ";

char c;

cin>>c;

if(c=='y' || c=='Y')

{

fstream fin("multicor.dat",ios::binary | ios::in);

fstream fout("Temp.dat",ios::binary | ios::out);

for(int h=1;h<r;h++)

{

questionmc q;

fin.read((char\*)&q,sizeof(q));

fout.write((char\*)&q,sizeof(q));

}

questionmc q1;

fin.read((char\*)&q1,sizeof(q1));

for(int j=(r+1);j<=(noquestion2()-1);j++)

{

questionmc q;

fin.read((char\*)&q,sizeof(q));

fout.write((char\*)&q,sizeof(q));

}

fin.close();

fout.close();

int g=noquestion2();

fstream f1("multicor.dat", ios::binary | ios::out);

fstream f2("Temp.dat",ios::binary | ios::in);

for(int y=1;y<=(g-1);y++)

{

questionmc q;

f2.read((char\*)&q,sizeof(q));

f1.write((char\*)&q,sizeof(q));

}

f1.close();

f2.close();

cout<<"SUCCESSFULLY DELETED"<<endl;

}

delay(100);

cout<<"Do you wish to delete any other question(Y/N)? ";

cin>>k;

}

else

k='n';

}

}while(k=='Y' || k=='y');

}

void removestudent()

{

clrscr();

Student1 w1,s;

int a=0;

int h=0;

while(a==0)

{

fstream fin("Student.dat",ios::binary |ios::in);

gotoxy(1,1);

char user[21];

cout<<"Enter username of student: ";

cin>>user;

user[strlen(user)]='\0';

int l=strlen(user);

int k=0;

while(h<nostudent() && k==0)

{

fin.read((char\*)&s,sizeof(s));

if(strcmp(user,s.username)==0)

k=1;

h++;

}

if(k==0)

{

clrscr();

cout<<endl<<"Invalid username!!";

l=strlen(user);

gotoxy(l+16,1);

backspace(l);

a=0;

}

else

a=1;

}

clrscr();

s.display();

cout<<endl<<endl;

cout<<"Is this the student you wish to remove from the list(Y/N): ";

char d;

cin>>d;

if(d=='Y' || d=='y')

{

fstream fin1("Student.dat",ios::binary | ios::in);

fstream fout1("Temp.dat",ios::binary | ios::out);

for(int y=1;y<h;y++)

{

Student1 s1;

fin1.read((char\*)&s1,sizeof(s1));

fout1.write((char\*)&s1,sizeof(s1));

}

fin1.read((char\*)&w1,sizeof(w1));

for(int j1=(h+1);j1<=nostudent();j1++)

{

Student1 s2;

fin1.read((char\*)&s2,sizeof(s2));

fout1.write((char\*)&s2,sizeof(s2));

}

fin1.close();

fout1.close();

int ti=nostudent();

fstream f1("Student.dat",ios::binary | ios::out);

fstream f2("Temp.dat",ios::binary | ios::in);

for(int h=1;h<=ti;h++)

{

Student1 s3;

f2.read((char\*)&s3,sizeof(s3));

f1.write((char\*)&s3,sizeof(s3));

}

f1.close();

f2.close();

}

}

void altertime()

{

fstream fout("Time.dat",ios::binary | ios::out);

clrscr();

cout<<"Enter the new duration of the exam(in seconds): ";

long d;

cin>>d;

fout<<d;

}

void adminfunction()

{

clrscr();

char g;

do

{

clrscr();

char c;

cout<<"Enter q-alter any question"<<endl

<<" a-change your password"<<endl

<<" i- to include a new question"<<endl

<<" v-view marks of a student"<<endl

<<" d-delete a question"<<endl

<<" s-remove a student from the list"<<endl

<<" t-alter duration of the exam"<<endl;

cin>>c;

if(c=='q' || c=='Q')

alter();

else if(c=='a' || c=='A')

passchange();

else if(c=='i' || c=='I')

addquestion();

else if(c=='v' || c=='V')

viewmarks();

else if(c=='d' || c=='D')

deletequestion();

else if(c=='s' || c=='S')

removestudent();

else if(c=='t' || c=='T')

altertime();

else

cout<<"Invalid operation!";

clrscr();

cout<<"Do you wish to perform any other operation(Y/N): ";

cin>>g;

}while(g=='Y' || g=='Y');

}

void admin()

{

clrscr();

cout<<"Username:admin"<<endl;

cout<<"Please enter your password: ";

char k[21];

getpassword(k);

char t[21];

fstream fin("Admin.dat",ios::binary |ios::in);

fin>>t;

t[strlen(t)]='\0';

int j=0;

for(int i=0;i<5 && j==0;i++)

{

if(strcmp(k,t)==0)

j++;

else

{

insertspace(10);

cout<<" Wrong password";

back(28);

backspace(strlen(k));

getpassword(k);

}

}

if(i==5)

{

cout<<"You have exceeded maximum number of attempts"

<<endl<<"Press any key to exit";

exit(0);

}

else

adminfunction();

}

2)LIBRARY.cpp

This file contains functions which do simple operations like printing a given number of lines/spaces etc.

#include<iostream.h>

void insertline(int s)

{

for(int t=0;t<s;t++)

cout<<endl;

}

void insertspace(int d)

{

for(int i=0;i<d;i++)

cout<<" ";

}

void backspace(int y)

{

for(int d=0;d<y;d++)

cout<<"\b"<<" "<<"\b";

}

void back(int u)

{

for(int h=0;h<u;h++)

cout<<"\b";

}

3)MOUSEFUN.cpp

This file contains functions which enables us to use simple mouse programming.

#include<dos.h>

union REGS in,out;

int callmouse()

{

in.x.ax=1;

int86(51,&in,&out);

return 1;

}

void mouseposi(int \*a,int \*b,int \*cl)

{

int xpos,ypos,click;

in.x.ax=3;

int86(51,&in,&out);

click=out.x.bx;

\*cl=click;

xpos=out.x.cx;

\*a=xpos;

ypos=out.x.dx;

\*b=ypos;

}

4)STUDENT.cpp

This program is the one which contains the main function and contains the functions which the student can use.

#include<iostream.h>

#include<fstream.h>

#include<stdlib.h>

#include<conio.h>

#include<string.h>

#include<stdio.h>

#include<dos.h>

extern insertline(int),getpassword(char\*),back(int),backspace(int),markdisp(char\*);

extern admin(),markdisp2(char b[21]),insertspace(int),read(char\*,long,int a),read2(char\*,unsigned char\*,int \*a);

extern noquestion1(),noquestion2(),nostudent(),mouseposi(int\*,int\*,int\*);

struct Student

{

char name[21],username[21],pass[21],contactno[10],gender;

int age,marks[3],std,rno,attendance;

Student();

void getpass();

void accept();

void display();

void comprusername(char A[]);

void checkpass(char D[]);

};

Student::Student()

{

for(int y=0;y<strlen(name);y++)

name[y]='\0';

for(int i=0;i<10;i++)

{

gender='M';

contactno[i]=0;

}

for(int u=0;u<3;u++)

marks[u]=0;

attendance=rno=0;

}

void Student::getpass()

{

char ch;

int i=1;

ch=getch();

pass[0]=ch;

while(ch!=13)

{

cout<<"\*";

ch=getch();

while(ch=='\b')

{

cout<<"\b"<<" "<<"\b";

i--;

ch=getch();

}

pass[i]=ch;

i++;

}

pass[i-1]='\0';

}

void Student::accept()

{

cout<<"Enter your name: ";

gets(name);

cout<<"Enter your age: ";

cin>>age;

cout<<"Enter your gender: ";

cin>>gender;

cout<<"Enter your class: ";

cin>>std;

cout<<"Enter your contactno:";

cin>>contactno;

cout<<"Enter your username: ";

cin>>username;

int l=strlen(username);

username[l]='\0';

int h=0;

int k=1;

while(k!=0)

{

h=0;

fstream file("Student.dat",ios::binary | ios::in);

for(int i=0;i<nostudent() && h!=1;i++)

{

Student s1;

file.read((char\*)&s1,sizeof(s1));

if(strcmp(username,s1.username)==0)

h=1;

}

if(h==1)

{

k=1;

cout<<"Username already taken";

gotoxy(22+l,6);

backspace(l);

cin>>username;

l=strlen(username);

username[l]='\0';

}

else

k=0;

file.close();

}

gotoxy(0,7);

insertspace(22);

gotoxy(0,7);

cout<<endl<<"Enter your password: ";

getpass();

char r[21];

cout<<endl<<"Re-enter password: ";

getpassword(r);

r[strlen(r)]='\0';

while(strcmp(r,pass)!=0)

{

insertspace(20);

cout<<"password dosen't match";

back(22);

backspace(20+strlen(r));

getpassword(r);

r[strlen(r)]='\0';

}

}

void Student:: display()

{

cout<<"Name: "<< name <<endl

<<"Gender: m"<< gender <<endl

<<"Age: "<< age <<endl

<<"Username: "<< username <<endl;

}

int nostudent()

{

fstream fg("Student.dat",ios::binary | ios::in);

Student s1;

int y=0;

while(!fg.eof())

{

fg.read((char\*)&s1,sizeof(s1));

y++;

}

return y;

}

int main()

{

clrscr();

char t;

char A[21];

cout<<"Enter user type a-admin, s-student: ";

cin>>t;

int y;

Student q;

int no=0;

if(t=='s' || t=='S')

{

char c;

cout<<"Enter i to sign in and u to sign up: ";

cin>>c;

clrscr();

if(c=='u' || c=='U')

{

char c='n';

while(c=='n' || c=='N')

{

q.accept();

cout<<endl<<"Please check whether the entered details are correct if no(press n) if yes(press any other key)";

cin>>c;

cout<<endl;

}

q.rno=nostudent()+1000;

fstream fout("Student.dat",ios::binary | ios::out | ios::app);

fout.write((char\*)&q,sizeof(q));

clrscr();

cout<<"Registration successful"<<endl;

cout<<"Your roll number: "<< q.rno <<endl;

fout.close();

}

clrscr();

Student s;

if(c=='i' || c=='I')

{

int a=0;

int l=0;

while(a==0)

{

fstream fin("Student.dat",ios::binary |ios::in);

gotoxy(1,1);

char user[21];

cout<<"Enter username: ";

gotoxy(l+16,1);

backspace(l);

cin>>user;

user[strlen(user)]='\0';

int k=0;

no=0;

while(!fin.eof() && k==0)

{

fin.read((char\*)&s,sizeof(s));

if(strcmp(user,s.username)==0)

k++;

no++;

}

if(k==0)

{

clrscr();

cout<<endl<<"Invalid username!!";

l=strlen(user);

a=0;

}

else

a=1;

fin.close();

}

char r[21];

y=0;

int i=0;

int u=0;

while(y==0 && i<5)

{

gotoxy(30,2);

backspace(30);

cout<<"Enter password: ";

gotoxy(u+16,2);

backspace(u);

getpassword(r);

if(strcmp(r,s.pass)!=0)

{

cout<<endl<<"Incorrect Password";

u=strlen(r);

i++;

}

else

y++;

if(i==5)

{

cout<<"You have exceeded maximum number of attempts "

<<"Press any key to exit";

getch();

exit(0);

}

}

}

for(int i=0;i<21;i++)

A[i]='\0';

unsigned char B[21];

for(int j=0;j<21;j++)

B[j]=0;

int k;

fstream ti("Time.dat",ios::binary | ios::in);

ti>>k;

int m1,m2;

if(y==1)

{

clrscr();

s.display();

insertline(3);

cout<<"Do you wish to take up the test or "

<<"modify any of your details(t/m)";

char g;

cin>>g;

if(g=='m' || g=='M')

{

If(s.attendance==0)

{

clrscr();

cout<<"You cannot alter your username!!"<<endl

<<"Enter n-to alter your name"

<<" p-alter password"

<<" c-alter contact number";

char v;

cin>>v;

clrscr();

if(v=='n'|| v=='N')

{

char name[21];

cout<<"Enter your name: ";

cin>>name;

strcpy(s.name,name);

}

else if(v=='p' || v=='P')

{

cout<<"Enter your new password: ";

char p[21];

getpassword(p);

strcpy(s.pass,p);

}

else if(v=='c' || v=='C')

{

cout<<"Enter your new contact number: ";

char co[21];

cin>>co;

strcpy(s.contactno,co);

}

else

cout<<"Invalid operation";

Student w1;

fstream fin1("Student.dat",ios::binary | ios::in);

fstream fout1("Temp.dat",ios::binary | ios::out);

for(int y=1;y<no;y++)

{

Student s1;

fin1.read((char\*)&s1,sizeof(s1));

fout1.write((char\*)&s1,sizeof(s1));

}

fin1.read((char\*)&w1,sizeof(w1));

fout1.write((char\*)&s,sizeof(s));

for(int j1=(no+1);j1<=nostudent();j1++)

{

Student s2;

fin1.read((char\*)&s2,sizeof(s2));

fout1.write((char\*)&s2,sizeof(s2));

}

fin1.close();

fout1.close();

int ti=nostudent();

fstream f1("Student.dat",ios::binary | ios::out);

fstream f2("Temp.dat",ios::binary | ios::in);

for(int h=1;h<=ti;h++)

{

Student s3;

f2.read((char\*)&s3,sizeof(s3));

f1.write((char\*)&s3,sizeof(s3));

}

f1.close();

f2.close();

}

else

{

cout<<"You cannot alter!You have already written the test":

}

}

else if(g=='t' || g=='T')

{

clrscr();

if(s.attendance==1)

{

cout<<"You have already taken the test";

gotoxy(13,9);

cout<<"Marks in single option correct: "<< s.marks[0] <<endl;

insertspace(12);

cout<<"Marks in multiple option correct: "<< s.marks[1] <<endl;

insertspace(12);

cout<<"Total marks obtained: "<< s.marks[0]+s.marks[1] <<endl;

insertspace(12);

cout<<"Maximum marks: "<<noquestion1()\*3+noquestion2()\*4;

insertline(3);

cerr<<"Press any key to exit";

getch();

exit(0);

}

else

{

ifstream fd("instruct.txt",ios::in);

char g[1000];

while(!fd.eof())

{

fd.getline(g,1000);

cout<<g<<endl;

}

int q;

getch();

cout<<q; //to skip the warning 'q' is never used//

clrscr();

q=read(A,y,k);

read2(A,B,&q);

int a1=0,b1=0,c1=0;

int x1=1,y1=0,z1=0,yr=0;

int end=0;

int l=0,n=0;

while(!end)

{

if(x1)

{

m1=markdisp(A);

gotoxy(20,6);

insertline(12);

insertspace(62);

cout<<"NEXT";

}

else if(y1)

{

m2=markdisp2(B);

gotoxy(20,6);

insertline(12);

insertspace(14);

cout<<"PREVIOUS";

insertspace(40);

cout<<"NEXT";

}

else if(z1)

{

s.marks[0]=m1;

s.marks[1]=m2;

s.marks[2]=m1+m2;

clrscr();

insertspace(15);

cout<<"Your final report: ";

gotoxy(13,9);

cout<<"Marks in single option correct: "<<m1<<endl;

insertspace(12);

cout<<"Marks in multiple option correct: "<<m2<<endl;

insertspace(12);

cout<<"Total marks obtained: "<<m1+m2<<endl;

insertspace(12);

cout<<"Maximum marks: "<<noquestion1()\*3+noquestion2()\*4;

insertline(3);

s.attendance=1;

gotoxy(20,6);

insertline(12);

insertspace(14);

cout<<"PREVIOUS";

insertspace(40);

cout<<"STOP";

if(n && !yr)

end=1;

}

else;

l=((a1>111 && a1<176) && (b1>135 && b1<144));

n=((a1>496 && a1<528) && (b1>135 && b1<144));

c1=0;

while(c1!=1)

{

delay(100);

mouseposi(&a1,&b1,&c1);

l=((a1>111 && a1<176) && (b1>135 && b1<144));

n=((a1>496 && a1<528) && (b1>135 && b1<144));

if(!l && !n)

c1=0;

}

int x2=x1;

yr=y1;

x1=(l&&y1);

y1=((n&&x2) || (l&&z1));

z1=((n && yr) || ( z1 && !l));

}

Student w1;

fstream fin1("Student.dat",ios::binary | ios::in);

fstream fout1("Temp.dat",ios::binary | ios::out);

for(int y=1;y<no;y++)

{

Student s1;

fin1.read((char\*)&s1,sizeof(s1));

fout1.write((char\*)&s1,sizeof(s1));

}

fin1.read((char\*)&w1,sizeof(w1));

fout1.write((char\*)&s,sizeof(s));

clrscr();

for(int j1=(no+1);j1<=nostudent();j1++)

{

Student s2;

fin1.read((char\*)&s2,sizeof(s2));

fout1.write((char\*)&s2,sizeof(s2));

}

fin1.close();

fout1.close();

int ti=nostudent();

fstream f1("Student.dat",ios::binary | ios::out);

fstream f2("Temp.dat",ios::binary | ios::in);

for(int h=1;h<=ti;h++)

{

Student s3;

f2.read((char\*)&s3,sizeof(s3));

f1.write((char\*)&s3,sizeof(s3));

}

f1.close();

f2.close();

}

}

}

else;

}

else if(t=='a' || t=='A')

admin();

else;

return 0;

}

4)WORK1.cpp

This file governs flow of control in the 'single correct' type questions. It is stores the answers entered by student and at last print the result for this section.

#include<iostream.h>

#include<conio.h>

#include<string.h>

#include<stdio.h>

#include<fstream.h>

#include<stdlib.h>

#include<graphics.h>

#include<dos.h>

extern insertspace(int),insertline(int),back(int),backspace(int);

extern callmouse(),mouseposi(int \*a,int \*b,int \*c);

class question

{

public:

char q[1000],a[20],b[20],c[20],d[20],ans;

public:

question();

void display();

void accept();

char ansret();

};

void question::question()

{

for(int i=0;i<1000;i++)

q[i]=32;

for(int y=0;y<20;y++)

a[y]=b[y]=c[y]=d[y]=32;

ans=32;

}

void question::accept()

{

cout<<endl<<"Enter the question: ";

gets(q);

cout<<endl<<"Enter option A: ";

gets(a);

cout<<endl<<"Enter option B: ";

gets(b);

cout<<endl<<"Enter option C: ";

cin>>c;

cout<<endl<<"Enter option D: ";

gets(d);

delay(100);

cout<<endl<<"Enter correct answer: ";

cin>>ans;

}

void question::display()

{

gotoxy(20,6);

cout<<q;

insertline(3);

insertspace(13);

cout<<"A)( )"<<a <<endl;

insertspace(13);

cout<<"B)( )"<<b <<endl;

insertspace(13);

cout<<"C)( )"<<c<<endl;

insertspace(13);

cout<<"D)( )"<<d<<endl;

insertspace(13);

cout<<" ( )Clear response";

}

char question:: ansret()

{

return ans;

}

void next(int rec,int m);

void getpassword(char p[21])

{

char ch=getch();

p[0]=ch;

int i=1;

while(ch!=13)

{

cout<<"\*";

ch=getch();

while(ch=='\b')

{

backspace(1);

ch=getch();

i--;

}

p[i]=ch;

i++;

}

p[i-1]='\0';

}

long noquestion1()

{

fstream fin("Question.dat",ios::binary | ios::in);

question q;

fin.seekg(0L,ios::end);

long lastbyte=fin.tellg();

return (lastbyte/sizeof(q));

}

char getstans(int d,int &ti)

{

int a,b,cl=0;

callmouse();

char ans='\0';

char c='\0';

int l=0,n=0;

struct dostime\_t t;

int ts=t.second;

while(cl!=1)

{

delay(100);

mouseposi(&a,&b,&cl);

\_dos\_gettime(&t);

gotoxy(40,1);

int x=ti/3600;

int h=ti%3600;

int w=h/60;

int z=h%60;

printf("Time left: %2d:%02d:%02d\n",x,w,z);

\_dos\_gettime(&t);

if(t.second!=ts)

{

ti--;

ts=t.second;

}

if(ti<0)

{

cl=1;

a=0;

b=0;

}

}

l=((a>111 && a<176) && (b>135 && b<144));

n=((a>496 && a<528) && (b>135 && b<144));

cl=0;

if(a>127 && a<136)

{

if(b>63 && b<72)

{

gotoxy(13,10);

cout<<" ";

gotoxy(13,11);

cout<<" ";

gotoxy(13,12);

cout<<" ";

gotoxy(13,9);

cout<<"\*";

ans='a';

}

else if(b>71 && b<80)

{

gotoxy(13,9);

cout<<" ";

gotoxy(13,11);

cout<<" ";

gotoxy(13,12);

cout<<" ";

gotoxy(13,10);

cout<<"\*";

ans='b';

}

else if(b>79 && b<88)

{

gotoxy(13,9);

cout<<" ";

gotoxy(13,10);

cout<<" ";

gotoxy(13,12);

cout<<" ";

gotoxy(13,11);

cout<<"\*";

ans='c';

}

else if(b>87 && b<96)

{

gotoxy(13,9);

cout<<" ";

gotoxy(13,10);

cout<<" ";

gotoxy(13,11);

cout<<" ";

gotoxy(13,12);

cout<<"\*";

ans='d';

}

else if(b>95 && b<104)

{

gotoxy(13,9);

cout<<" ";

gotoxy(13,10);

cout<<" ";

gotoxy(13,11);

cout<<" ";

gotoxy(13,12);

cout<<" ";

ans='\0';

}

else;

c=ans;

}

else if(n)

c='n';

else if(l && d!=1)

c='p';

else;

return c;

}

int read(char \*A,long y,int a)

{

int m=0;

long i=noquestion1();

next(y,i);

char l='\0';

while(m==0 && a>0)

{

struct dostime\_t t;

\_dos\_gettime(&t);

int b=t.second;

gotoxy(40,1);

int x=(a)/3600;

int h=(a)%3600;

int w=h/60;

int z=h%60;

printf("Time left: %2d:%02d:%02d\n",x,w,z);

\_dos\_gettime(&t);

if(t.second!=b)

a--;

b=t.second;

if(l=='n')

{

y++;

next(y,i);

}

else if(l=='p')

{

y--;

next(y,i);

}

else;

if(A[y]=='a')

{

gotoxy(13,9);

cout<<"\*";

}

else if(A[y]=='b')

{

gotoxy(13,10);

cout<<"\*";

}

else if(A[y]=='c')

{

gotoxy(13,11);

cout<<"\*";

}

else if(A[y]=='d')

{

gotoxy(13,12);

cout<<"\*";

}

else;

l=getstans(y,a);

if(l!='n' && l!='p'&& l!='s')

A[y]=l;

if((y==i) && (l=='n'))

m++;

}

return a;

}

void ansdisp(char B[][5]);

void next(int y,int h)

{

if(y<=h && y>0)

{

question q;

fstream f("Question.dat",ios::binary | ios::in);

f.seekg((sizeof(q))\*(y-1));

f.read((char\*)&q,sizeof(q));

clrscr();

gotoxy(20,20);

clrscr();

gotoxy(15,6);

cout<<"Q "<< y <<")";

q.display();

f.close();

if(y==1)

{

gotoxy(20,6);

insertline(12);

insertspace(62);

cout<<"NEXT";

}

if(y!=1)

{

gotoxy(20,6);

insertline(12);

insertspace(14);

cout<<"PREVIOUS";

insertspace(40);

cout<<"NEXT";

}

}

else;

}

int markdisp(char A[21])

{

int sum=0;

fstream f1("Question.dat",ios::binary | ios::in);

int i=noquestion1();

question s;

char g[21];

question q;

for(int j=1;j<=i;j++)

{

f1.read((char\*)&q,sizeof(q));

if(q.ansret()==A[j])

sum=sum+3;

else if(A[j]=='\0');

else if(q.ansret()!=A[j])

sum=sum-1;

else;

}

f1.close();

clrscr();

insertspace(15);

cout<<"YOUR SECTION 1 REPORT";

cout<<endl<<endl;

fstream f("Question.dat",ios::binary | ios::in);

question d;

insertspace(5);

gotoxy(10,5);

for(int c=1;c<=i;c++)

{

f.read((char\*)&d,sizeof(d)) ;

cout<<"Your answer: "<< A[c] <<" Correct answer: "<< d.ansret()<<endl;

insertspace(9);

}

f.close();

insertline(7);

insertspace(15);

cout<<"Marks obtained: "<< i\*3 <<endl;

insertspace(15);

cout<<"Maximum marks: "<< sum <<endl;

return sum;

}

4)WORK2.cpp

This file governs flow of control in the 'multi correct' type questions. It is stores the answers entered by student and at last print the result for this section.

#include<iostream.h>

#include<stdlib.h>

#include<fstream.h>

#include<string.h>

#include<stdio.h>

#include<ctype.h>

#include<conio.h>

#include<dos.h>

extern callmouse(),setposi(int,int,int),mouseposi(int \*a,int \*b,int \*c),insertspace(int),insertline(int);

extern back(int),backspace(int),read(char\*,long,int),noquestion1();

class question1

{

char q[1000],a[20],b[20],c[20],d[20];

unsigned char ans;

public:

question1();

void accept();

void display();

char ansret();

};

void question1::question1()

{

for(int i=0;i<1000;i++)

q[i]=' ';

for(int y=0;y<20;y++)

a[y]=b[y]=c[y]=d[y]=' ';

ans=0;

}

void question1::accept()

{

cout<<endl<<"Enter the question: ";

gets(q);

cout<<endl<<"Enter option A: ";

gets(a);

cout<<endl<<"Enter option B: ";

gets(b);

cout<<endl<<"Enter option C: ";

gets(c);

cout<<endl<<"Enter option D: ";

gets(d);

char c='\0';

int x[5];

for(int h=0;h<5;h++)

x[h]=0;

cout<<"Is A a correct answer to this question?(Y/N)"<<endl;

cin>>c;

if(c=='y' ||c=='Y')

x[1]=1;

else;

cout<<"Is B a correct answer to this question?(Y/N)"<<endl;

cin>>c;

if(c=='y' ||c=='Y')

x[2]=1;

else;

cout<<"Is C a correct answer to this question?(Y/N)"<<endl;

cin>>c;

if(c=='y' ||c=='Y')

x[3]=1;

else;

cout<<"Is D a correct answer to this question?(Y/N)"<<endl;

cin>>c;

if(c=='y' ||c=='Y')

x[4]=1;

else;

ans=(x[1]\*8)+(x[2]\*4)+(x[3]\*2)+(x[4]\*1);

}

void question1::display()

{

gotoxy(20,6);

puts(q);

insertline(2);

insertspace(13);

cout<<"A)( )"<<a <<" "<<endl;

insertspace(13);

cout<<"B)( )"<<b <<" "<<endl;

insertspace(13);

cout<<"C)( )"<<c<<" "<<endl;

insertspace(13);

cout<<"D)( )"<<d<<" "<<endl;

insertspace(13);

cout<<" ( )Clear response";

}

char question1:: ansret()

{

return ans;

}

long noquestion2()

{

fstream fin("multicor.dat",ios::binary | ios::in);

question1 q;

fin.seekg(0L,ios::end);

long lastbyte=fin.tellg();

return (lastbyte/sizeof(q));

}

char getstans2(unsigned char &ans,int f,int v,int &ti)

{

delay(100);

int a,b,cl=0;

char e='\0';

gotoxy(20,20);

unsigned char anstemp;

anstemp=ans;

gotoxy(4,1);

int d[5];

d[1]=int(anstemp>>3);

gotoxy(20,19);

anstemp=ans;

anstemp=anstemp<<5;

anstemp=anstemp>>7;

d[2]=int(anstemp);

anstemp=ans;

anstemp=anstemp<<6;

anstemp=anstemp>>7;

d[3]=int(anstemp);

anstemp=ans;

anstemp=anstemp<<7;

anstemp=anstemp>>7;

d[4]=int(anstemp);

anstemp=ans;

callmouse();

int l=0,m=0,n=0;

struct dostime\_t t;

\_dos\_gettime(&t);

int ts=t.second;

while(cl!=1)

{

delay(100);

mouseposi(&a,&b,&cl);

gotoxy(40,1);

int x=ti/3600;

int h=ti%3600;

int w=h/60;

int z=h%60;

printf("Time left: %2d:%02d:%02d\n",x,w,z);

\_dos\_gettime(&t);

if(t.second!=ts)

{

ti--;

ts=t.second;

}

if(ti<0)

{

cl=1;

a=0;

b=0;

}

}

l=((a>111 && a<176) && (b>135 && b<144));

m=((a>303 && a<352) && (b>167 && b<175));

n=((a>496 && a<528) && (b>135 && b<144));

cl=0;

if(a>127 && a<136)

{

if(b>63 && b<72)

{

gotoxy(13,9);

cout<<"\*";

d[1]=1;

}

else if(b>71 && b<80)

{

gotoxy(13,10);

cout<<"\*";

d[2]=1;

}

else if(b>79 && b<88)

{

gotoxy(13,11);

cout<<"\*";

d[3]=1;

}

else if(b>87 && b<96)

{

gotoxy(13,12);

cout<<"\*";

d[4]=1;

}

else if(b>95 && b<104)

{

gotoxy(13,9);

cout<<" ";

gotoxy(13,10);

cout<<" ";

gotoxy(13,11);

cout<<" ";

gotoxy(13,12);

cout<<" ";

for(int y=0;y<5;y++)

d[y]=0;

}

else;

}

else if(n && f!=v)

e='n';

else if(m && f==v)

e='s';

else if(l)

e='p';

else;

ans=(((d[1])\*8)+((d[2])\*4)+((d[3])\*2)+((d[4])\*1));

return e;

}

void next2(int i,int y);

void read2(char \*A,unsigned char ans[21],int \*a)

{

int y=1;

for(int g=0;g<21;g++)

ans[g]=0;

fstream fin("multicor.dat",ios::binary | ios::in);

int c=noquestion2();

fin.close();

clrscr();

cout<<"This section consists of "<< c <<" questions";

getch();

next2(1,c);

char f='\0';

while(f!='s' && \*a>0)

{

struct dostime\_t t;

\_dos\_gettime(&t);

int b=t.second;

int s=\*a;

gotoxy(40,1);

int x=(\*a)/3600;

s=(\*a)%3600;

int w=s/60;

int z=s%60;

printf("Time left: %2d:%02d:%02d\n",x,w,z);

\_dos\_gettime(&t);

if(t.second!=b)

{

(\*a)--;

b=t.second;

}

if(f=='n')

{

y++;

next2(y,c);

}

else if(f=='p')

{

if(y==1)

{

\*a=read(A,noquestion1(),\*a);

next2(1,c);

}

else

{

y--;

next2(y,c);

}

}

else;

unsigned char d[5],anstemp=ans[y];

d[1]=(anstemp>>3);

anstemp=anstemp<<5;

anstemp=anstemp>>7;

d[2]=anstemp;

anstemp=ans[y];

anstemp=anstemp<<6;

anstemp=anstemp>>7;

d[3]=anstemp;

anstemp=ans[y];

anstemp=anstemp<<7;

anstemp=anstemp>>7;

d[4]=(anstemp);

if((d[1])==1)

{

gotoxy(13,9);

cout<<"\*";

}

if((d[2])==1)

{

gotoxy(13,10);

cout<<"\*";

}

if((d[3])==1)

{

gotoxy(13,11);

cout<<"\*";

}

if((d[4])==1)

{

gotoxy(13,12);

cout<<"\*";

}

else;

f=getstans2(ans[y],y,c,\*a);

}

}

void next2(int y,int h)

{

if(y<=h && y>0)

{

fstream f("multicor.dat",ios::binary | ios::in);

question1 l;

f.seekg((sizeof(l))\*(y-1));

f.read((char\*)&l,sizeof(l));

clrscr();

gotoxy(15,6);

cout<<"Q"<< y <<")";

l.display();

f.close();

if(y==h)

{

gotoxy(20,6);

insertline(12);

insertspace(14);

cout<<"PREVIOUS";

insertline(4);

insertspace(38);

cout<<"SUBMIT";

}

if(y!=h)

{

gotoxy(20,6);

insertline(12);

insertspace(14);

cout<<"PREVIOUS";

insertspace(40);

cout<<"NEXT";

}

}

else;

}

int markdisp2(char ans[21])

{

clrscr();

int u=noquestion2();

fstream fin("multicor.dat",ios::binary | ios::in);

int sum=0;

insertspace(15);

cout<<"YOUR SECTION 2 REPORT ";

gotoxy(10,5);

for(int y=1;y<=u;y++)

{

question1 q;

fin.read((char\*)&q,sizeof(q));

if(ans[y]==q.ansret())

sum+=4;

else

{

if(ans[y]==0);

else

sum-=1;

}

int d[5];

int c[5];

unsigned char anstemp=ans[y];

d[1]=(anstemp>>3);

anstemp=anstemp<<5;

anstemp=anstemp>>7;

d[2]=anstemp;

anstemp=ans[y];

anstemp=anstemp<<6;

anstemp=anstemp>>7;

d[3]=anstemp;

anstemp=ans[y];

anstemp=anstemp<<7;

anstemp=anstemp>>7;

d[4]=anstemp;

anstemp=q.ansret();

c[1]=(anstemp>>3);

anstemp=anstemp<<5;

anstemp=anstemp>>7;

c[2]=anstemp;

anstemp=q.ansret();

anstemp=anstemp<<6;

anstemp=anstemp>>7;

c[3]=anstemp;

anstemp=q.ansret();

anstemp=anstemp<<7;

anstemp=anstemp>>7;

c[4]=anstemp;

cout<<"Your response/s for question "<< y <<" : ";

if(d[1]==1)

cout<<"A";

if(d[2]==1)

cout<<"B";

if(d[3]==1)

cout<<"C";

if(d[4]==1)

cout<<"D";

cout<<" Correct answer/s: ";

if(c[1]==1)

cout<<"A";

if(c[2]==1)

cout<<"B";

if(c[3]==1)

cout<<"C";

if(c[4]==1)

cout<<"D";

cout<<endl;

insertspace(9);

}

insertline(5);

fin.close();

insertspace(15);

cout<<"Marks obtained: "<< sum << endl;

insertspace(15);

cout<<"Maximum marks: "<< u\*4 <<endl;

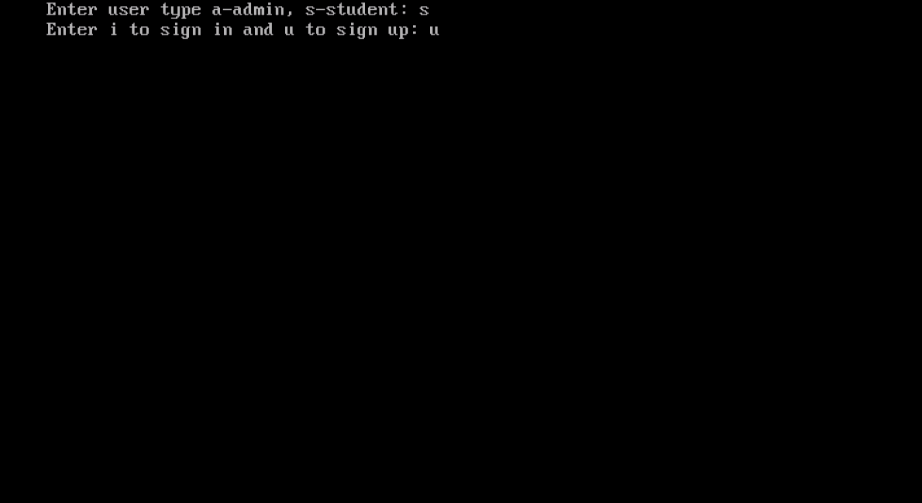
return sum;

}

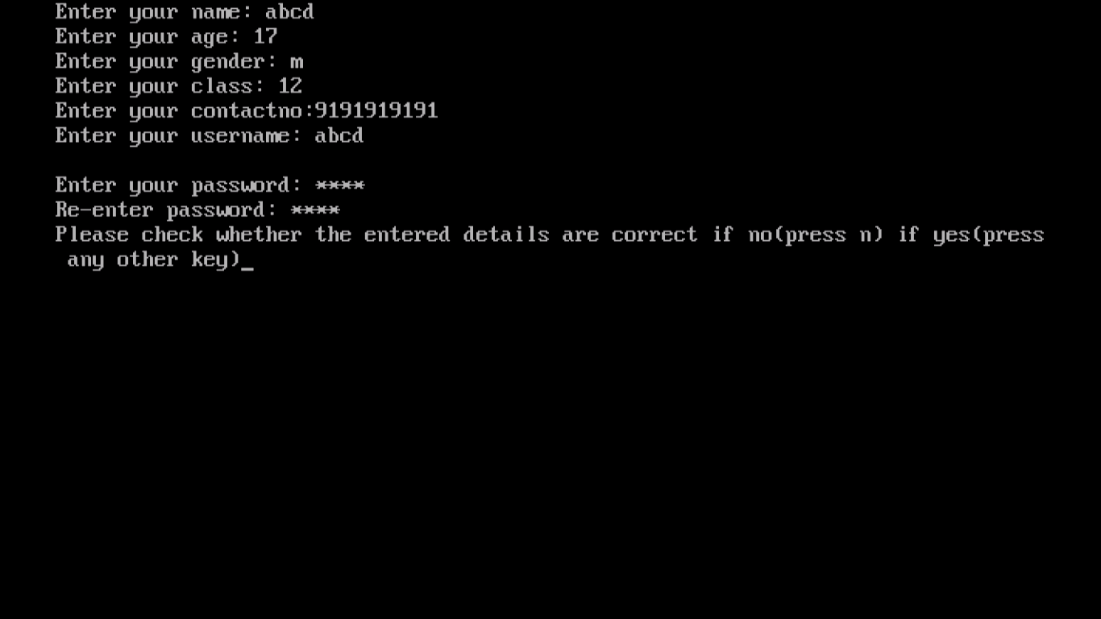
1. **Start screen**



1. **Student sign in/up**



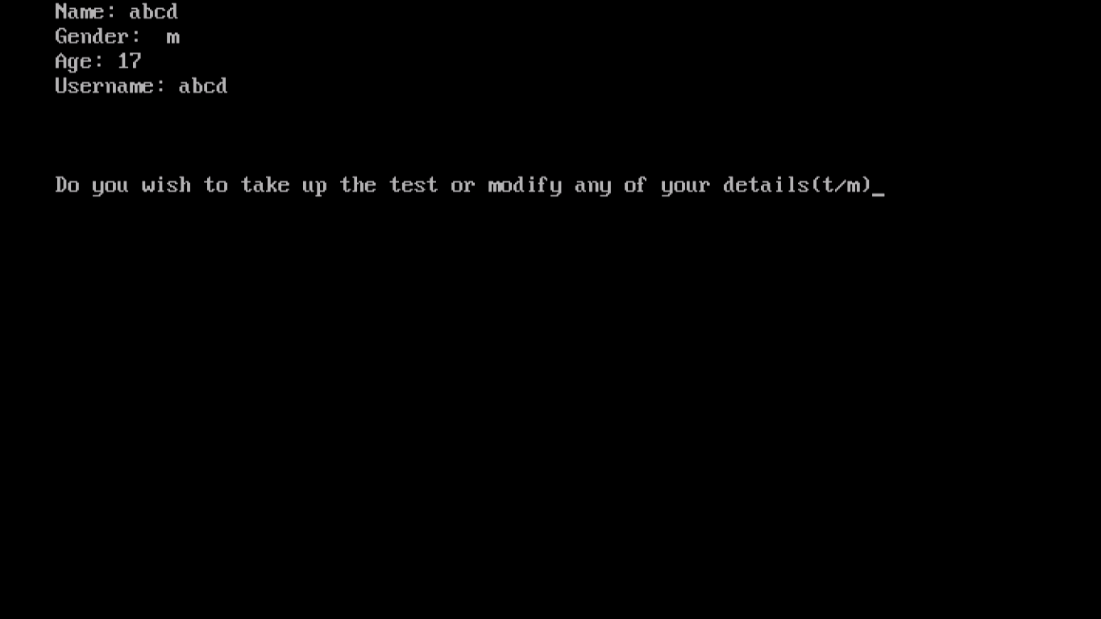
1. **Student sign up**

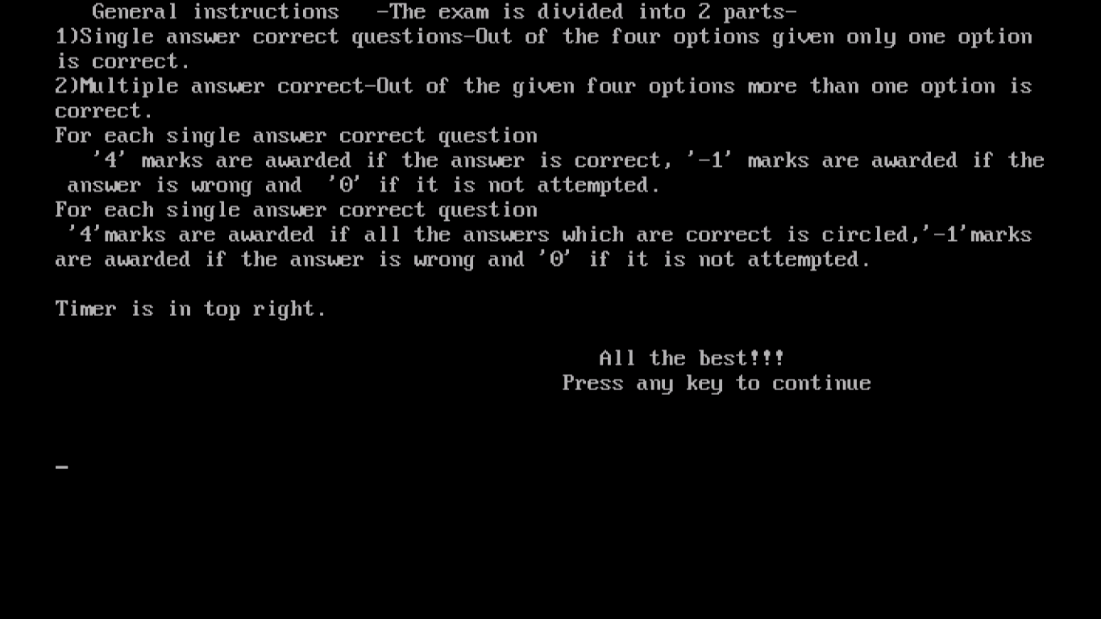


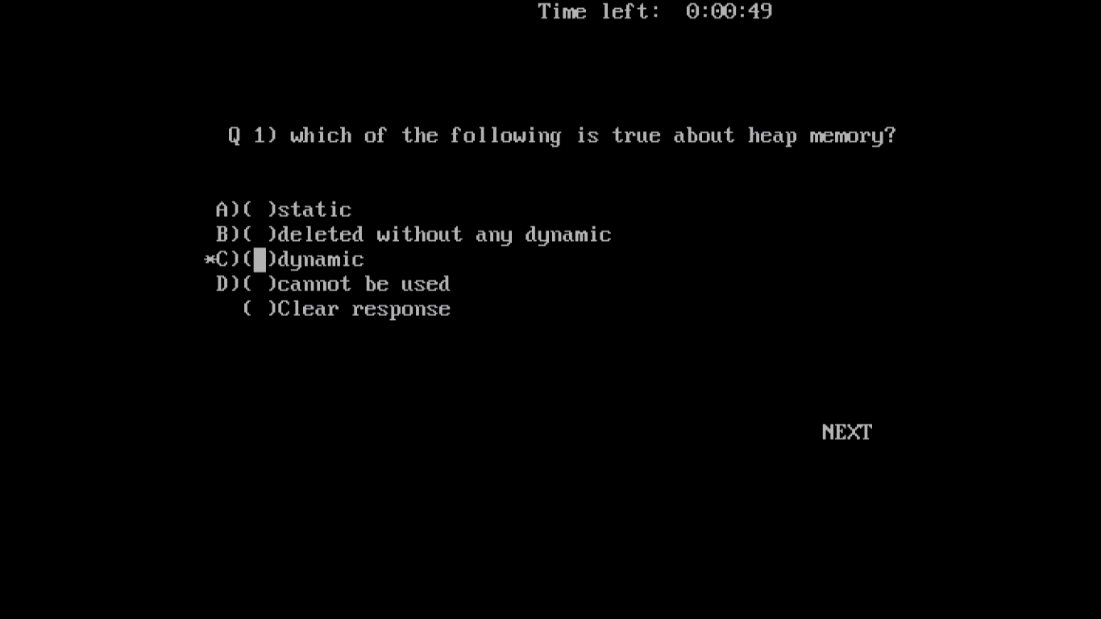
1. **Student signing in**



1. **Student can take up the test or modify their details**



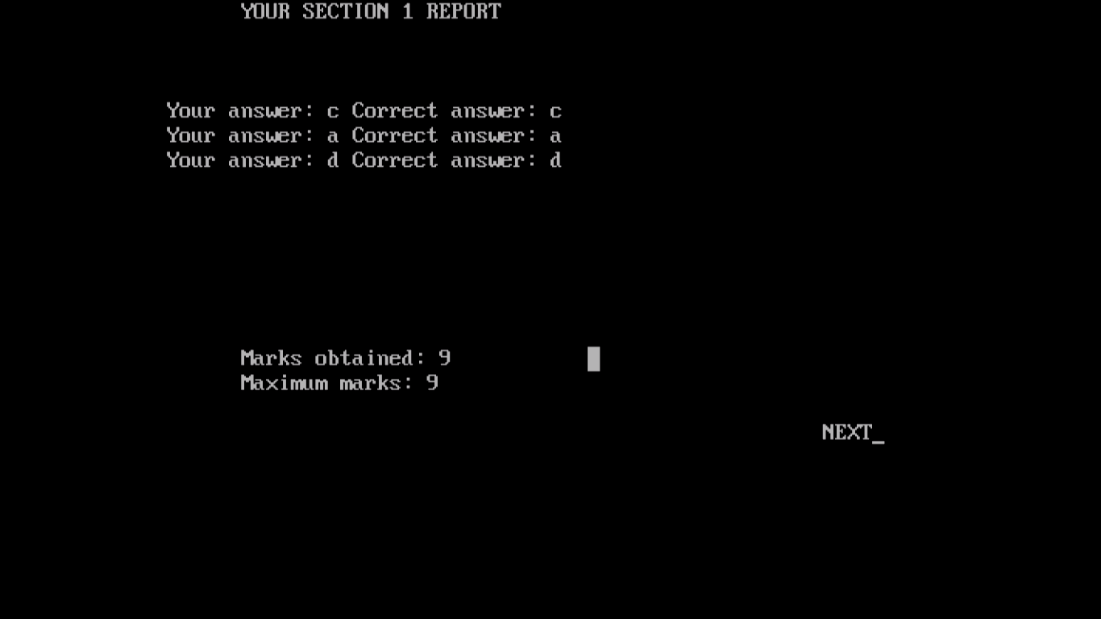
1. **General instructions for the student taking up the test.**
2. **A single-option correct question**



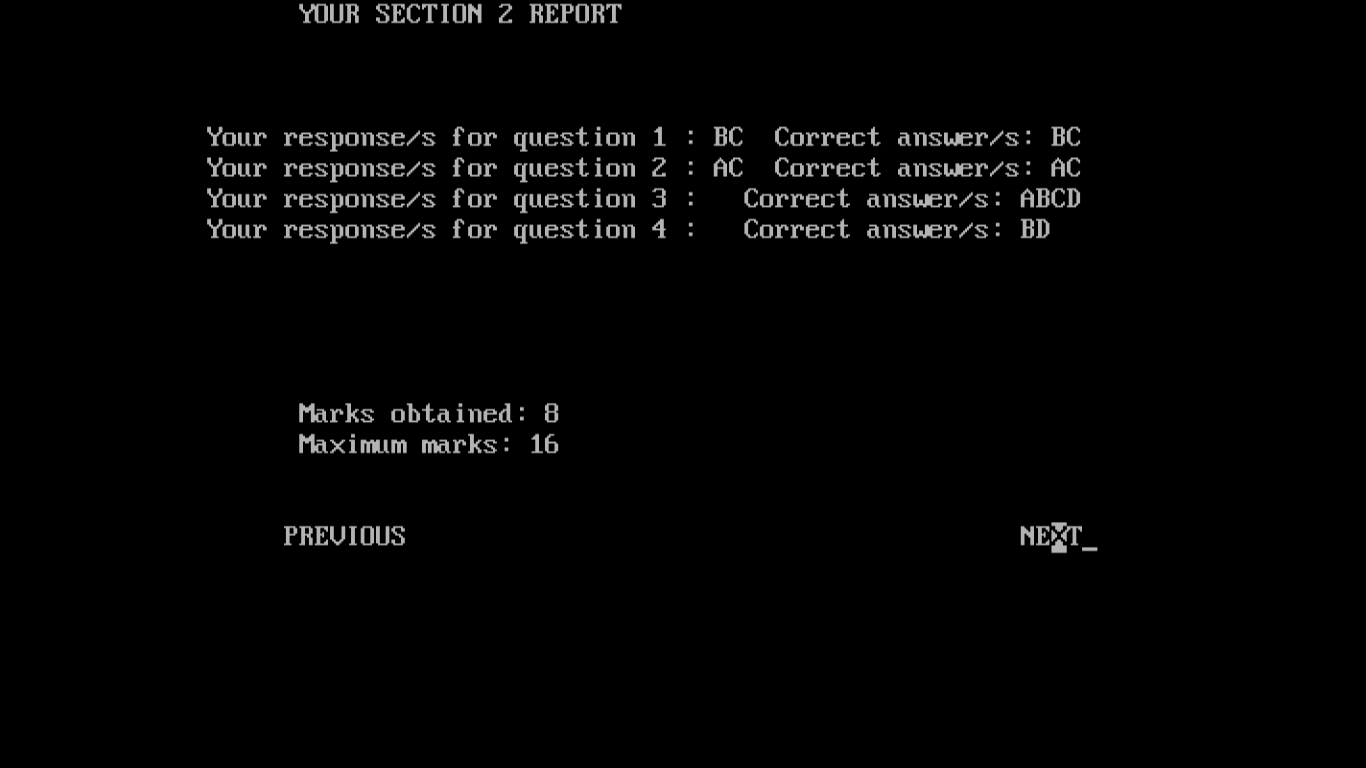
1. **A multiple-option correct question**



1. **Report for single answer correct questions.**



1. **Report for multiple-option correct questions.**



1. **Final report**



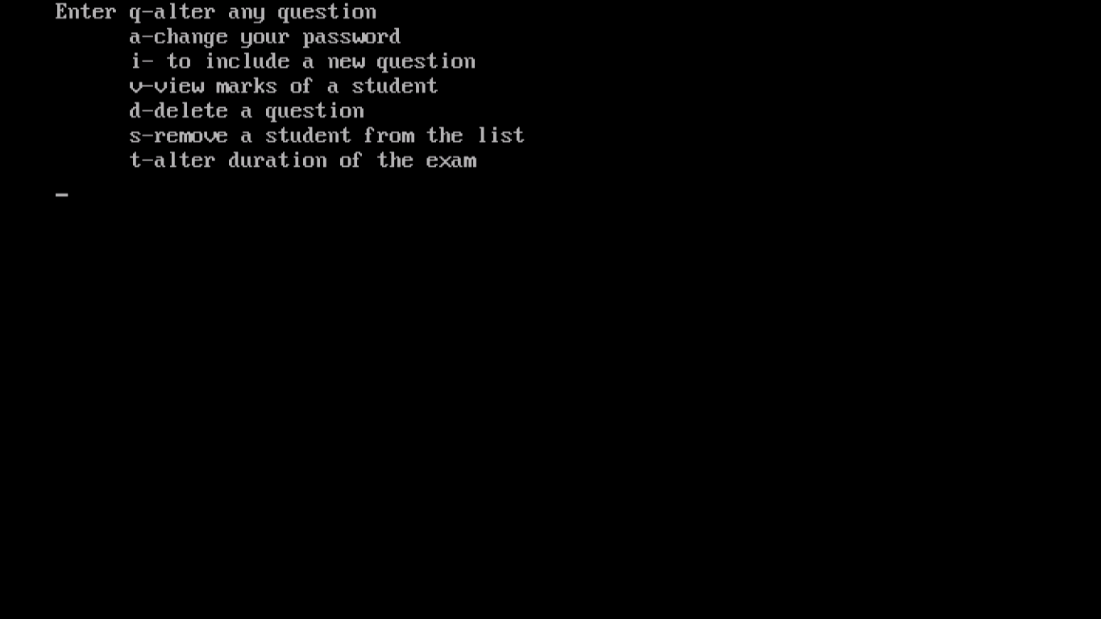
1. **Student modifying details.**



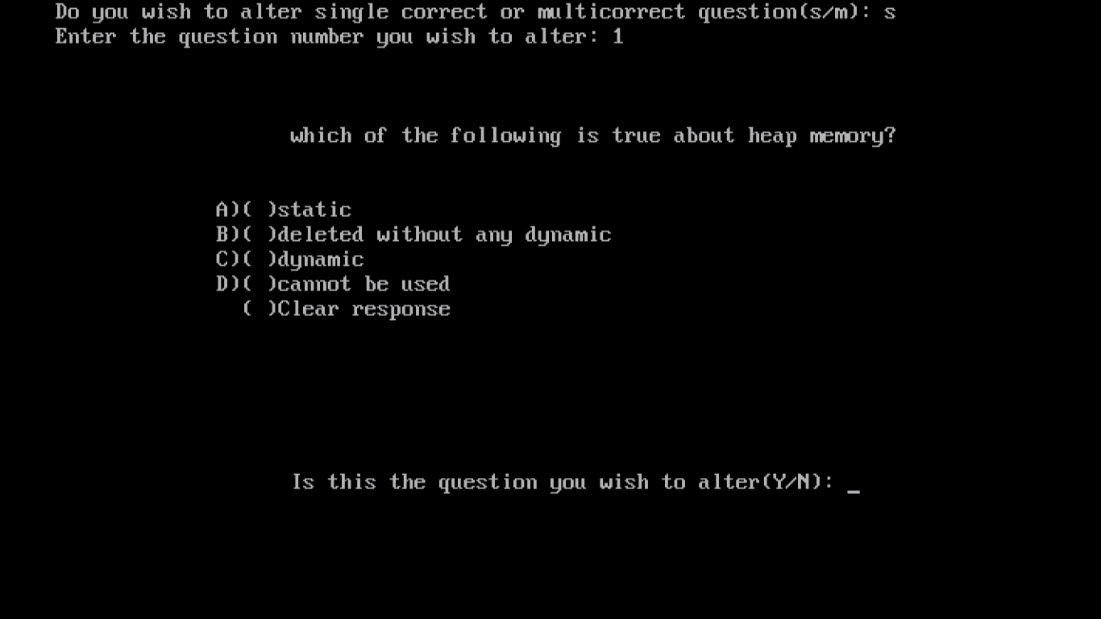
1. **Admin logging in**



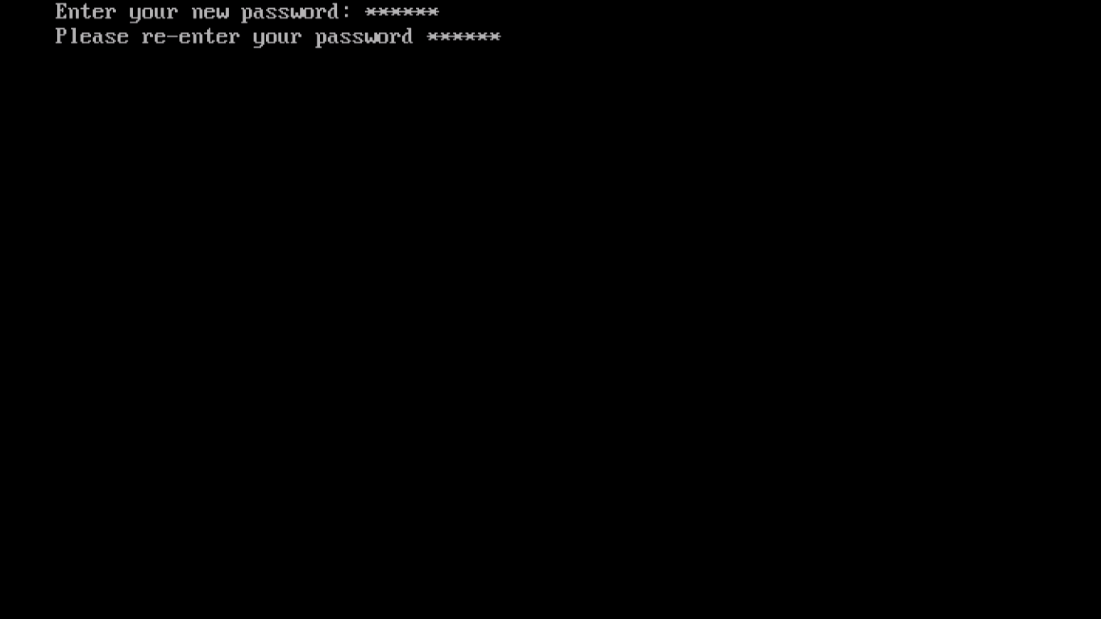
1. **Functions of admin**



1. **Admin altering a question.**



1. **Admin changing password**



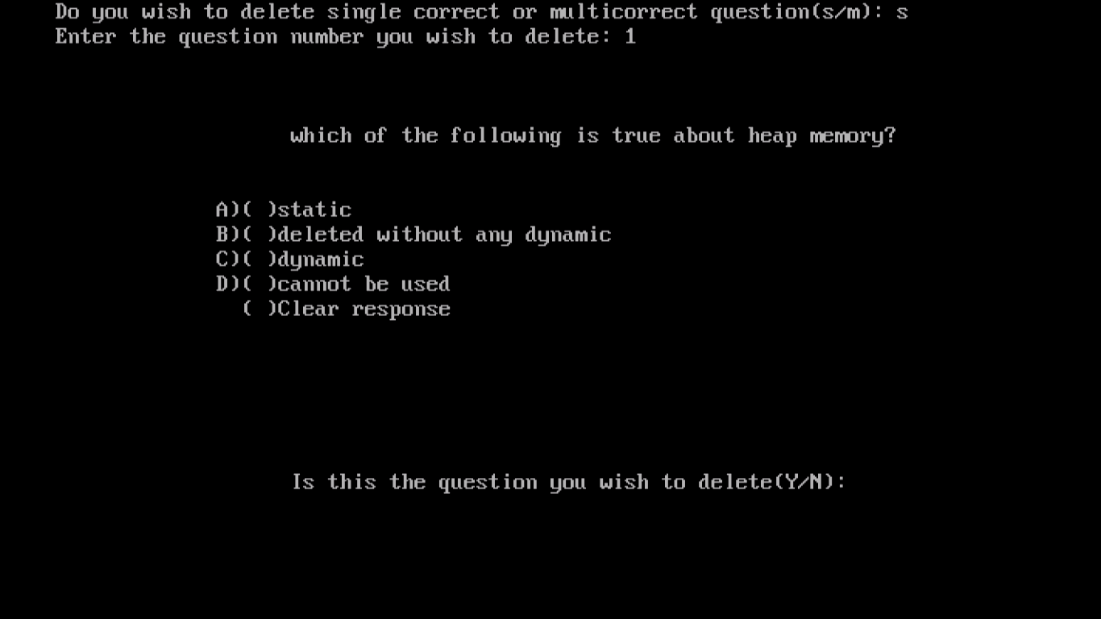
1. **Admin adding question**



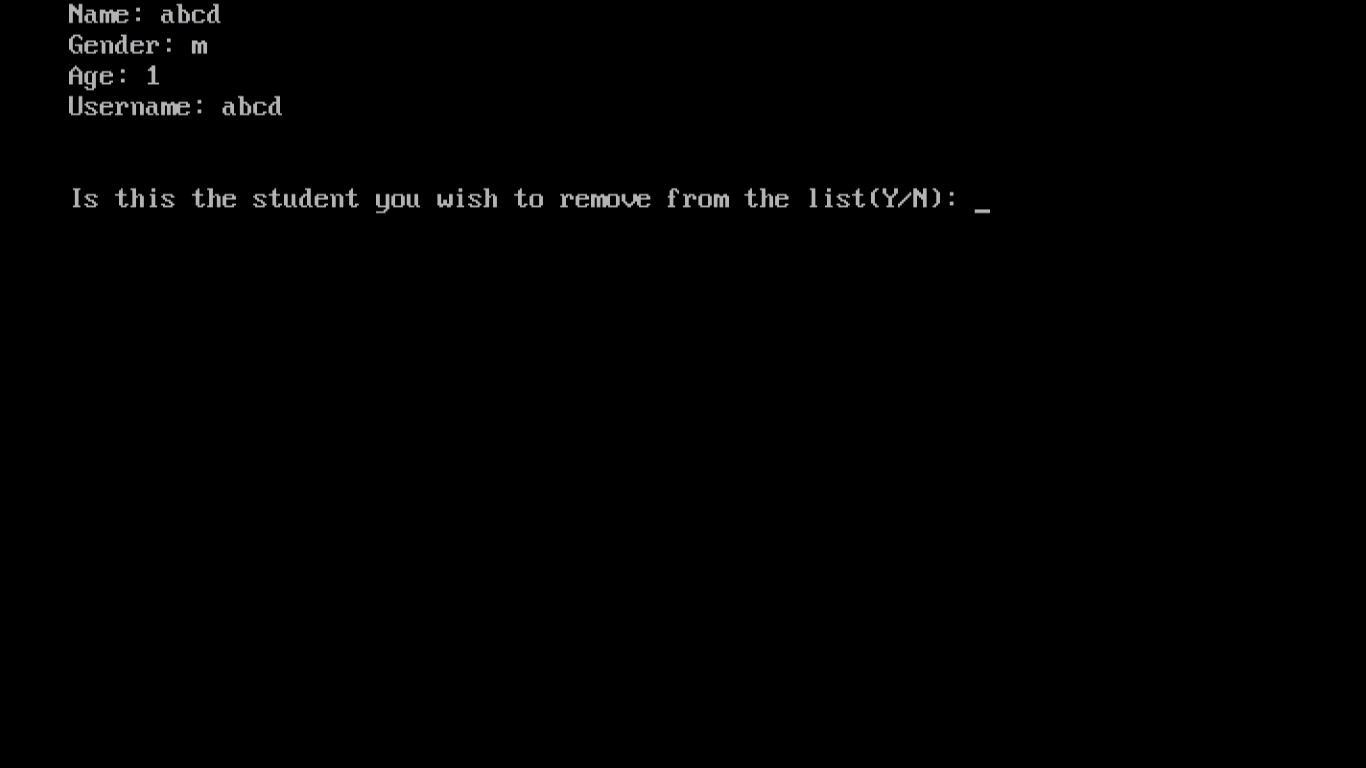
1. **Admin viewing a student's result.**



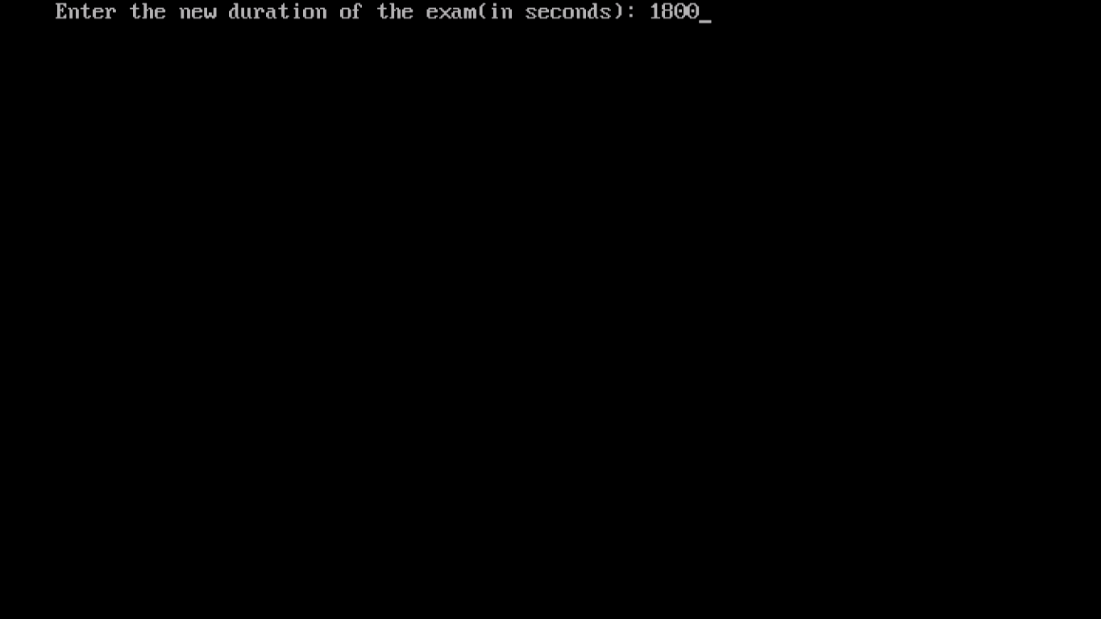
1. **Admin deleting a question.**



1. **Admin can remove any student.**



1. **Admin can alter the test timings.**



Limitations and Shortcoming

1. The application works only in "Turbo c++" as it involves older standard of mouse programming.
2. The timer used in the program works on the assumption that the loops involved in the program take only a few milliseconds to run.
3. There is no random access to questions.
4. The percentage marks or the percentile of a student is not calculated or displayed. Only their marks and the maximum marks are displayed.
5. The student cannot view their answers in future i.e. after finishing their test since it is not stored permanently. They can however view his results anytime.

References/Bibliography

electrosofts.com/cgraphics/mouse.html

Conclusion

The project titled "**ONLINE EXAMINATION MANAGEMENT SYSTEM**" done by **VIGNESH** for the academic year **2017-18** has been compiled, tested and executed.