NATIONAL P.G. COLLEGE

C++ PROJECT FILE

Submitted By:

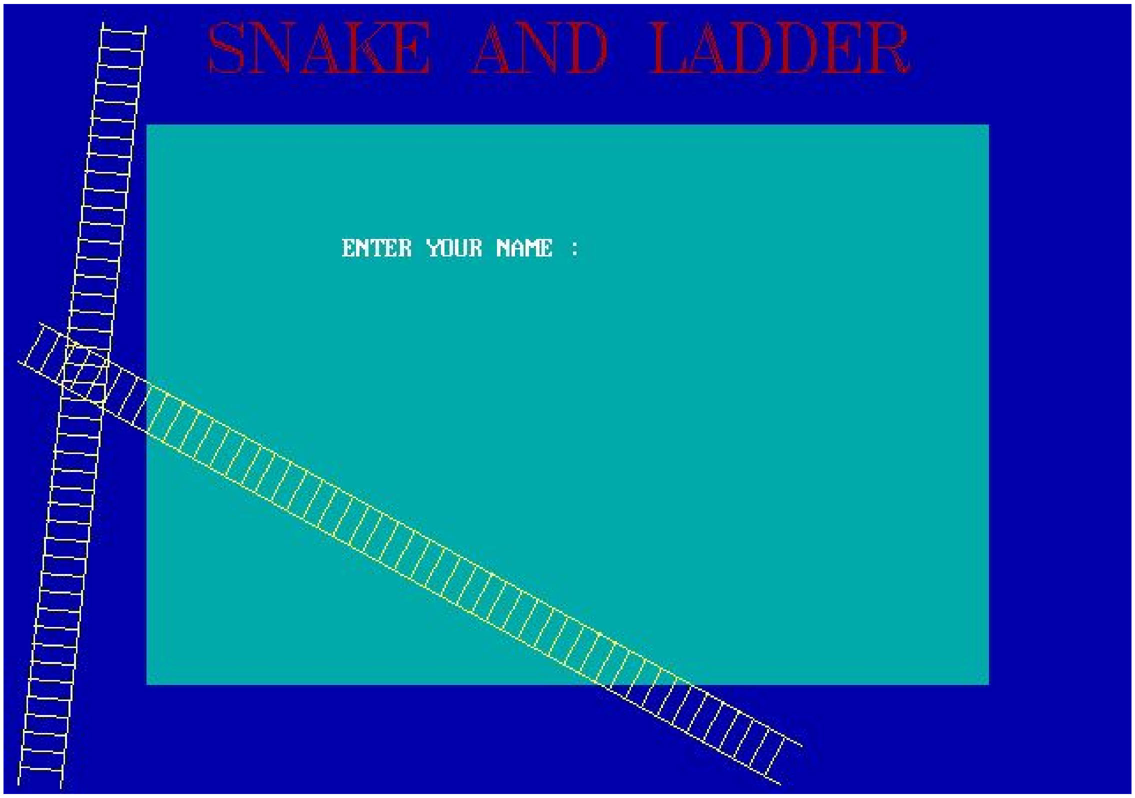
Bharat Singh Rajput
ID 42
B.C.A II Semester

Submitted To:

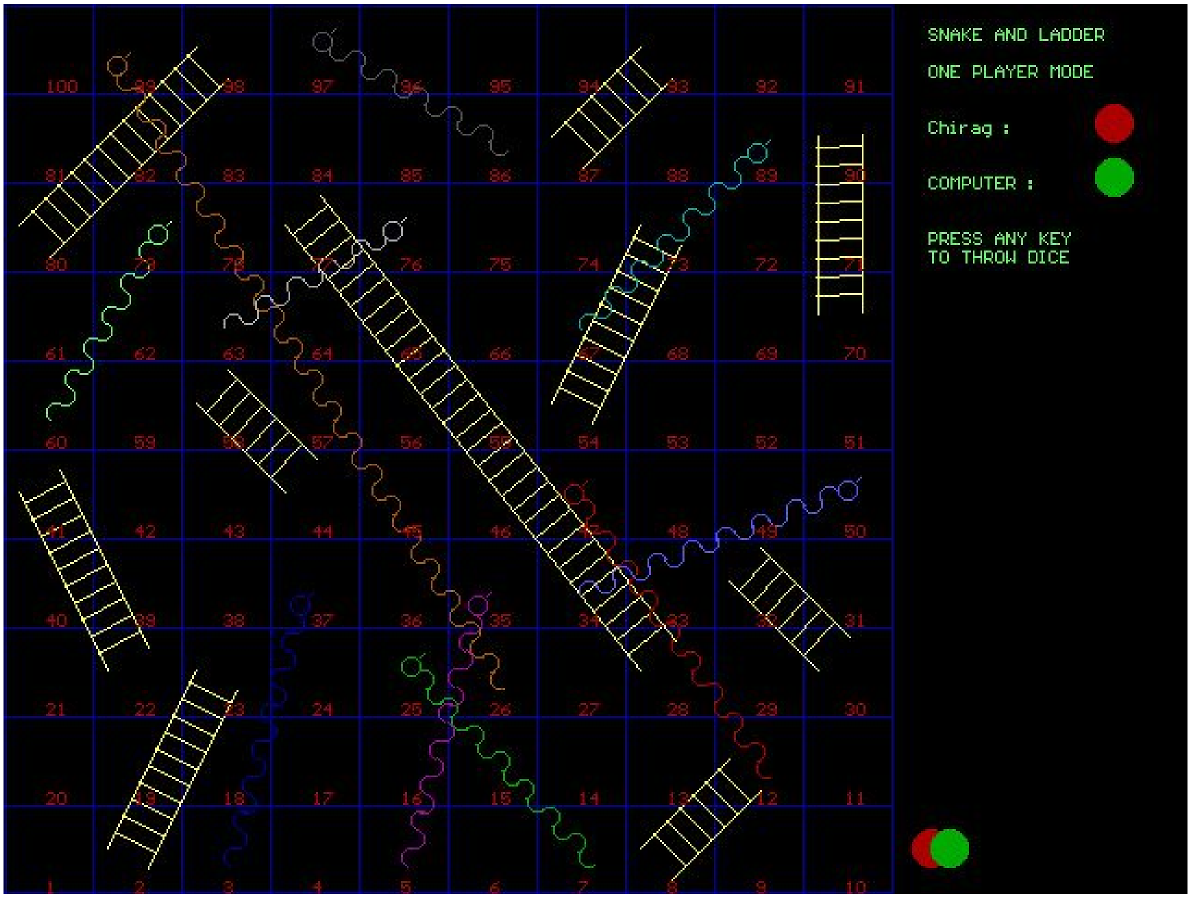
Mr. Amit Kumar Srivastava
Department Of Computer Science
National P.G. College, Lucknow.

INDEX

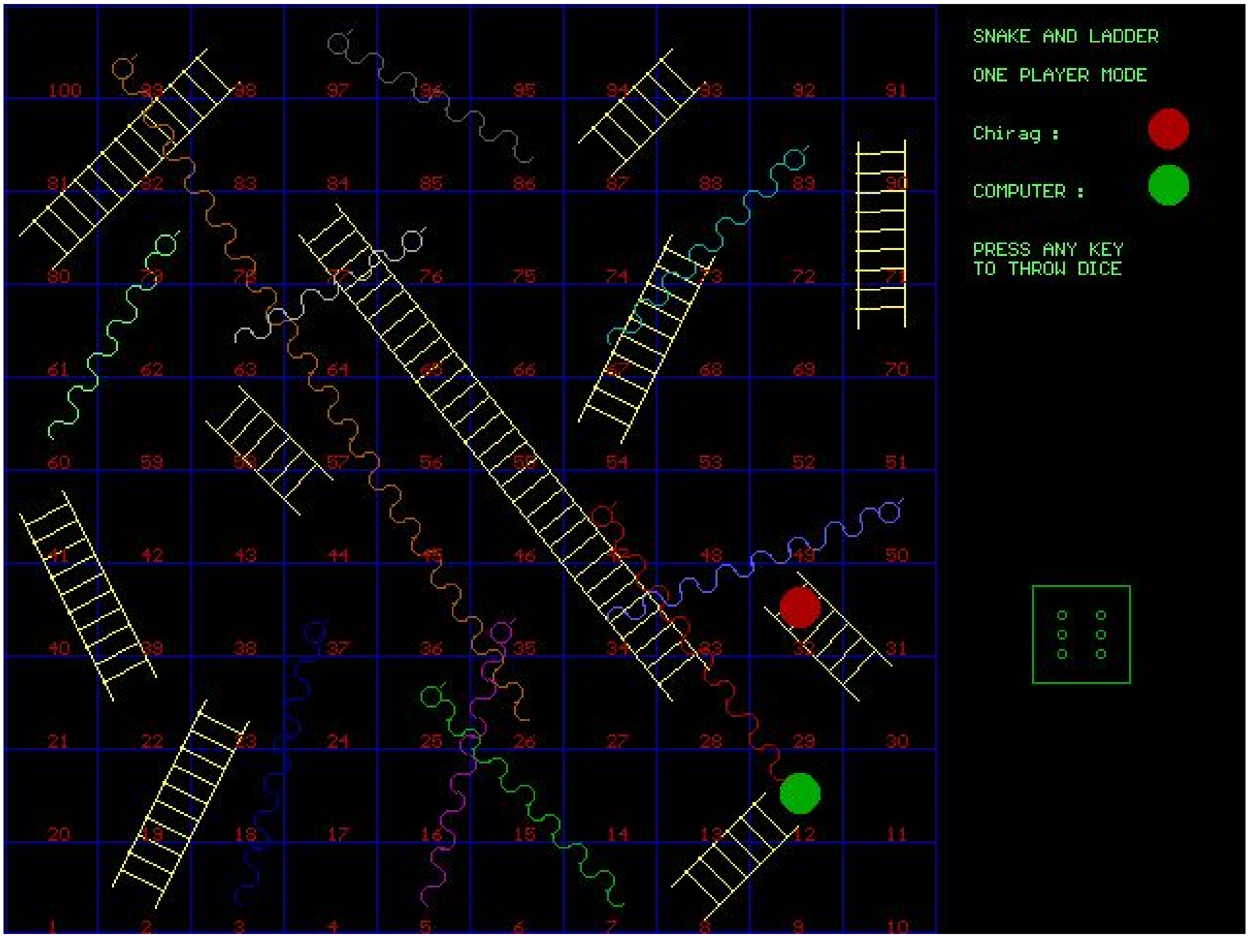
Q1. Write a program to create Snake and Ladder Game.



```
SNAKE AND LADDER

ENTER YOUR NAME :
```



```
SNAKE AND LADDER

ONE PLAYER MODE

Chirag :

COMPUTER :

PRESS ANY KEY
TO THROW DICE
```



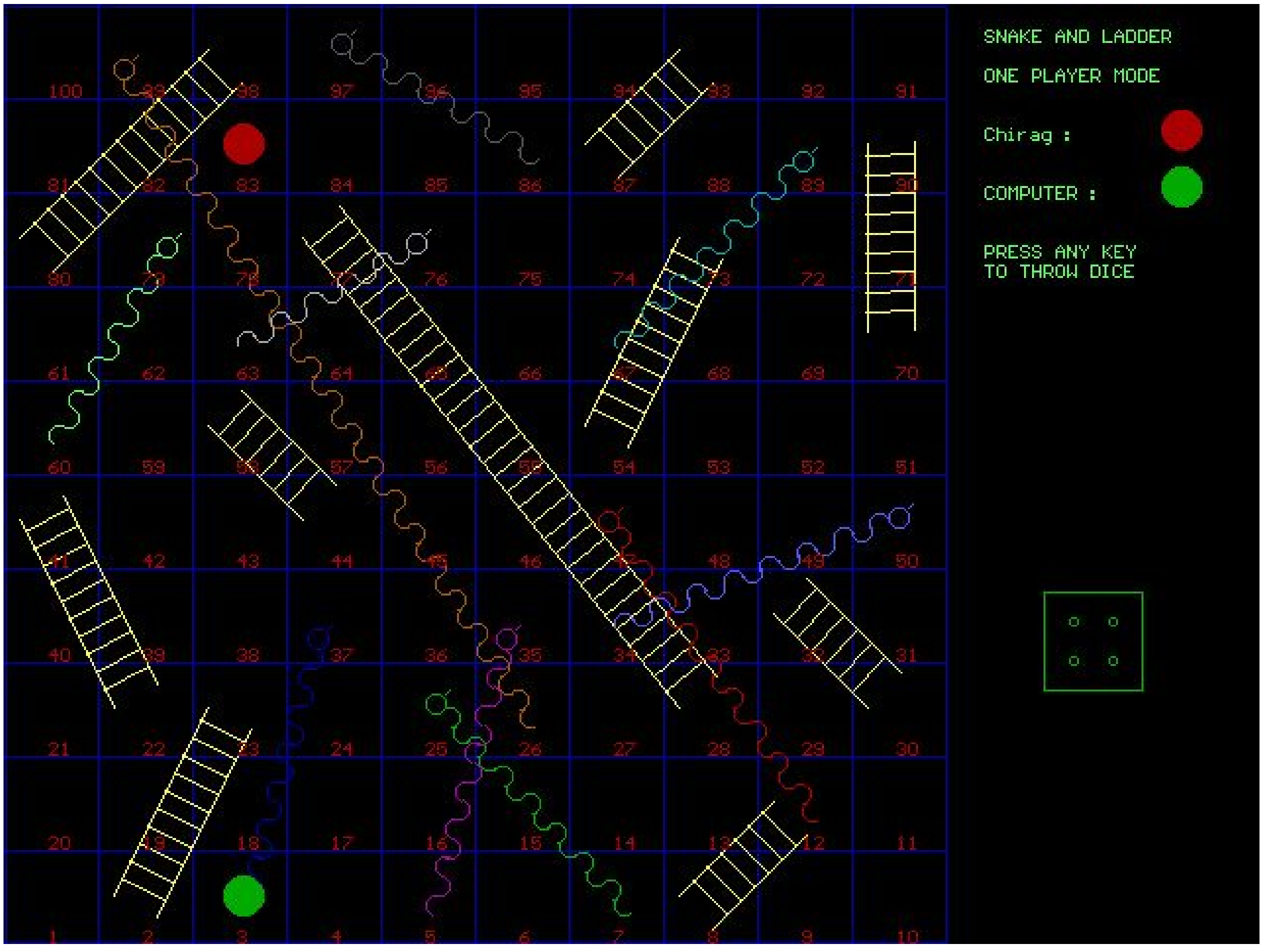
SNAKE AND LADDER

ONE PLAYER MODE

Chirag :

COMPUTER :

PRESS ANY KEY
TO THROW DICE



SNAKE AND LADDER

ONE PLAYER MODE

Chirag :

COMPUTER :

PRESS ANY KEY
TO THROW DICE

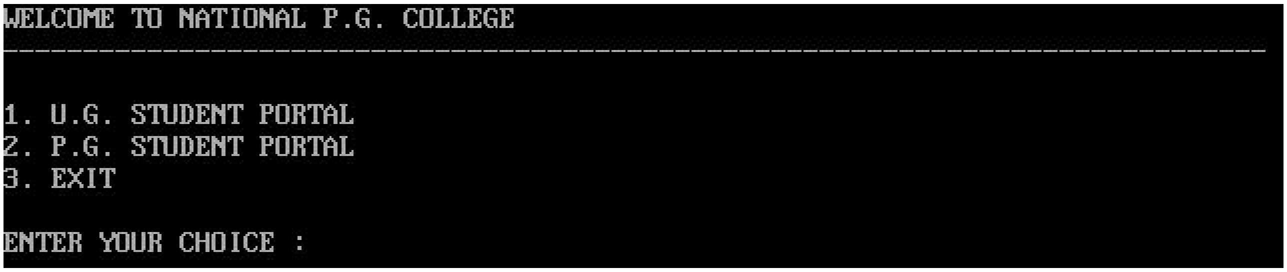
PROGRAM

```
/*
********************
* SNAKE AND LADDER *
********************
*/
#include<stdlib.h>
#include<conio.h>
#include<stdio.h>
#include<graphics.h>
#include<dos.h>
#include<string.h>
#include<math.h>
#define SNAKES 10
#define LADDERS 10
char p1Name[20],s[3];
struct ladders{
           int p1;
           int p2;
           }ladder[LADDERS] = {
                            { 2,23  },
                            { 22,41 },
                            {  8,12 },
                            { 30,32 },
                            { 44,58 },
                            { 28,77 },
                            { 54,73 },
                            { 70,90 },
                            { 80,98 },
                            { 87,93 }
                           };
struct snakes {
            int p1;
            int p2;
            int len;
            int angle;
            }snakes[SNAKES] = {
                            { 3 ,37,15,75 },
                            { 7 ,25,15,133 },
                            { 67,89,14,45 },
                            { 12,47,19,125 },
                            { 5 ,35,15,75  },
                            { 26,99,40,122 },
                            { 63,76,11,30  },
                            { 86,97,12,150 },
                            { 34,50,16,21  },
                            { 60,79,12,60  },
                            };

double rad(int deg) {
 return (deg*3.14159)/180;
}

char * inttostr(int n) {
 int  i=0;
 while(n > 0) {
      s[i++] = n%10 + 48;
      n /= 10;
 }
 s[i] = '\0';
 strrev(s);
 return s;
}

void getchords(int p1, int p2,int *x1,int *y1, int *x2, int *y2) {
if(p1 > 100) {
 p1 = 100;
 }
if(p2 > 100) {
 p2 = 100;
 }
     if((p1/10)%2) {
     if((p1%10) == 0) {
           *x1 = 480 - 48*((p1)%10);
           *y1 = 480 - 48*((p1-1)/10);
     }
 else {
           *x1 = 480 - 48*((p1-1)%10);
           *y1 = 480 - 48*(p1/10);
 }
}
     else {
           if((p1%10) == 0) {
                *x1 = 480 - 48*((p1-1)%10);
                *y1 = 480 - 48*((p1-1)/10);
     }
           else {
                *x1 = 48*(p1%10);
                *y1 = 480-48*((p1-1)/10);
           }

}
     if(x2 != NULL && y2 != NULL) {
        if((p2/10)%2)   {
        if((p2%10) == 0) {
            *x2 = 480 - 48*((p2)%10);
            *y2 = 480 - 48*((p2-1)/10);
        }
        else {
            *x2 = 480 - 48*((p2-1)%10);
            *y2 = 480 - 48*(p2/10);
        }
    }
    else {
        if((p2%10) == 0) {
            *x2 = 480 - 48*((p2-1)%10);
            *y2 = 480 - 48*((p2-1)/10);
    }
    else {
        *x2 = 48*(p2%10);
        *y2 = 480-48*((p2-1)/10);
        }
        }
    }
}

void snake(int p, int angle, int len) {
    int i,xt,yt;
    double x,y;
    getchords(p,0,&xt,&yt,NULL,NULL);
    x = xt-20;
    y = yt-20;
    for(i = 1 ; i < len ; i++) {
        if(i%2) {
            arc(x, y, angle, angle-180, 5);
        }
        else {
            arc(x, y, angle-200, angle, 5);
        }
        x = x + 10*cos(rad(angle));
        y = y - 10*sin(rad(angle));
    }
    circle(x,y,5);
    putpixel(x-2,y-2,BLUE);
    putpixel(x+2,y+2,BLUE);
    line(x+5,y-5,x+7,y-7);
}

void drawladder(int p1,int p2) {
    int x1,y1,x2,y2,x3,y3,sign,tempx,tempy,tempx2,tempy2,i;
    float p,b,angle,angle2,ladderlength,temp;
```

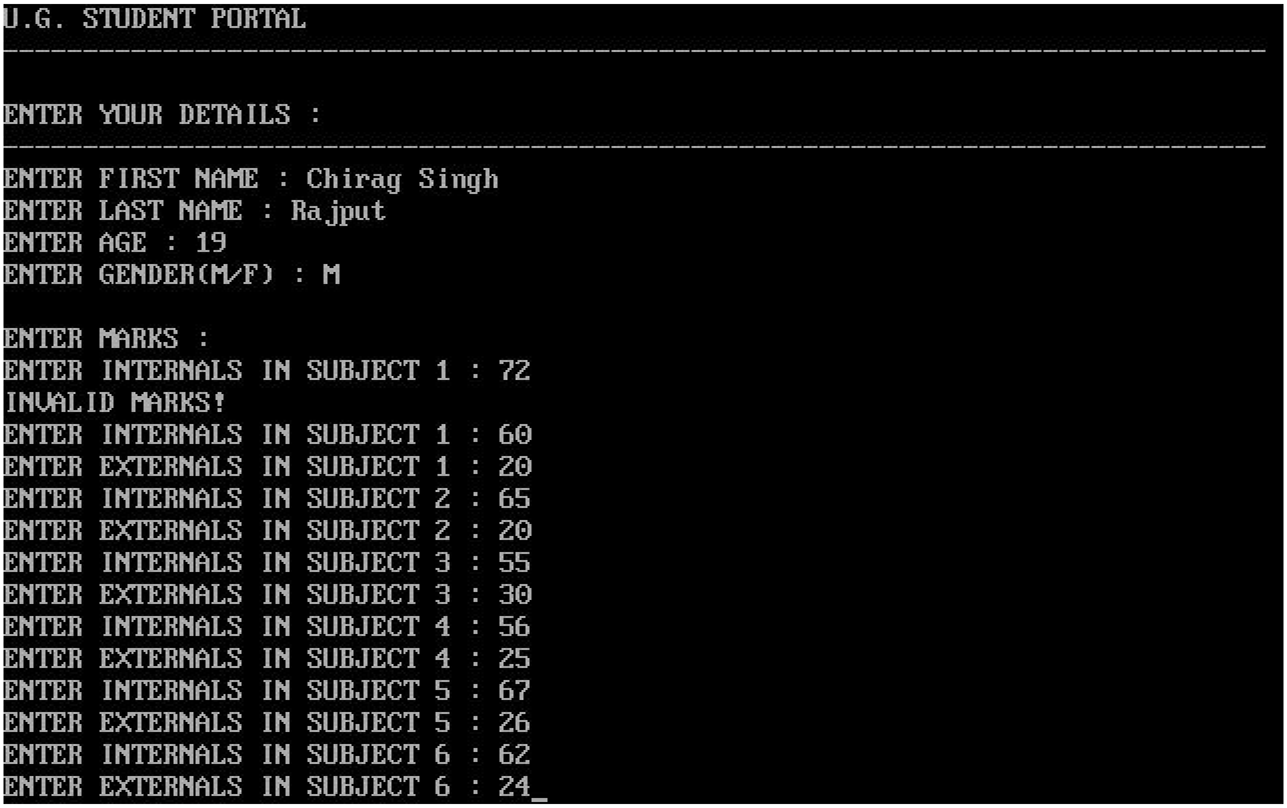
```
     setcolor(YELLOW);
     getchords(p1,p2,&x1,&y1,&x2,&y2);
     x1 -= 40;
     y1 -= 25;
     x2 -= 40;
     y2 -= 25;
     line(x1,y1,x2,y2);
     p = y2 - y1;
     b = x2 - x1;
     sign   = (x2 >= x1) ? 1 : -1;
     angle  = atan(p/b) + rad(90);
     angle2 = atan(p/b);
     temp   = cos(angle2);
     ladderlength = (temp <= 0) ? -p : b / temp;
     for(i = 1 ; i < ladderlength/10 * sign ; i++) {
      tempx  = x1 + 10 * i * cos(angle2) * sign;
      tempy  = y1 + 10 * i * sin(angle2) * sign;
      tempx2 = tempx + 25  * cos(angle)  * sign;
      tempy2 = tempy + 25  * sin(angle)  * sign;
      line(tempx,tempy,tempx2,tempy2);
     }
     x3 = x1 + 25 * cos(angle) * sign;
     y3 = y1 + 25 * sin(angle) * sign;
     x2 = x2 + 25 * cos(angle) * sign;
      y2 = y2 + 25 * sin(angle) * sign;
     line(x3,y3,x2,y2);
     setcolor(RED);
}

void drawdice(int x, int y,int face) {
rectangle(x,y,x+50,y+50);
switch(face) {
 case 1 :
       circle(x+25,y+25,2);
     break;
 case 2 :
       circle(x+15,y+15,2);
     circle(x+35,y+35,2);
     break;
 case 3 :
       circle(x+15,y+15,2);
     circle(x+25,y+25,2);
     circle(x+35,y+35,2);
     break;
 case 4 :
       circle(x+15,y+15,2);
     circle(x+35,y+15,2);
     circle(x+35,y+35,2);
     circle(x+15,y+35,2);
     break;
 case 5 :
       circle(x+15,y+15,2);
     circle(x+35,y+15,2);
     circle(x+35,y+35,2);
     circle(x+15,y+35,2);
     circle(x+25,y+25,2);
     break;
 case 6 :
       circle(x+15,y+15,2);
     circle(x+35,y+15,2);
     circle(x+35,y+35,2);
     circle(x+15,y+35,2);
     circle(x+15,y+25,2);
     circle(x+35,y+25,2);
     break;
 }
}

void drawmap() {
int i,x,y,j,x1,y1;
setcolor(BLUE);
rectangle(1,1,480,479);
for(i = 1 ; i < 10 ; i++) {
 line(48*i,0,48*i,479);
 line(0,48*i,480,48*i);
 }
 for(i = 0 ; i < LADDERS; i++) {
 drawladder(ladder[i].p1,ladder[i].p2);
 }
     for(i = 0 ; i < SNAKES ; i++) {
           setcolor(i%10+1);
           snake(snakes[i].p1,snakes[i].angle,snakes[i].len);
 }
 setcolor(RED);
 for(i = 1 , j = 100 ; i <= 50 ; i++,j--) {
     getchords(i,j,&x,&y,&x1,&y1);
     outtextxy(x-25,y-10,inttostr(i));
     outtextxy(x1-25,y1-10,inttostr(j));
     }
 setcolor(10);
outtextxy(500,10,"SNAKE AND LADDER");
}

void drawPlayerPos(int p1,int p2) {
 int x1,y1,x2,y2;
 getchords(p1,p2,&x1,&y1,&x2,&y2);
 if(p1 == p2) {
     x1 -= 5;
     x2 += 5;
     }
 setcolor(RED);
 circle(x1-22,y1-25,10);
 setfillstyle(1,RED);
 floodfill(x1-22,y1-25,RED);
 circle(600,64,10);
 floodfill(600,64,RED);
 setfillstyle(1,GREEN);
 setcolor(GREEN);
 circle(x2-22,y2-25,10);
 floodfill(x2-22,y2-25,GREEN);
 circle(600,93,10);
 floodfill(600,93,GREEN);
}

void p1Draw(int p1,int p2) {
cleardevice();
     drawmap();
     outtextxy(500,30,"ONE PLAYER MODE");
     outtextxy(500,60,p1Name);
     outtextxy(500,90,"COMPUTER : ");
     drawPlayerPos(p1,p2);
}

int check(int *pos) {
int i;
if(*pos >= 100) {
 return 1;
 }
for(i = 0 ; i < SNAKES ; i++) {
 if(*pos == snakes[i].p2) {
     *pos = snakes[i].p1;
     return 0;
     }
 }
 for(i = 0 ; i < LADDERS ; i++) {
 if(*pos == ladder[i].p1) {
     *pos = ladder[i].p2;
     return 0;
     }
 }
}

void p1() {
```

```
int player_pos=0,comp_pos=0,temp;
char ch;
strcat(p1Name," : ");
while(player_pos <= 100 && comp_pos <= 100) {
cleardevice();
drawmap();
outtextxy(500,30,"ONE PLAYER MODE");
outtextxy(500,60,p1Name);
outtextxy(500,90,"COMPUTER : ");
outtextxy(500,120,"PRESS ANY KEY");
outtextxy(500,130,"TO THROW DICE");
drawPlayerPos(player_pos,comp_pos);
ch = getch();
if(ch!=27) {
 temp = rand()%6+1;
 drawdice(530,300,temp);
 player_pos += temp;
 if(check(&player_pos) == 1) {
     settextstyle(1,0,5);
     outtextxy(getmaxx()/2 - textwidth("YOU WIN!")/2,getmaxy()/2,"YOU WIN!");
     break;
     }
     } else {
     exit(0);
     }
     getch();
     p1Draw(player_pos,comp_pos);
     outtextxy(500,120,"COMPUTER'S TURN");
     temp = rand()%6+1;
     cleardevice();
     drawmap();
     p1Draw(player_pos,comp_pos);
     outtextxy(500,120,"COMPUTER'S TURN");
     drawdice(530,300,temp);
     comp_pos += temp;
     delay(2000);
     if(check(&comp_pos) == 1) {
     settextstyle(1,0,5);
     outtextxy(getmaxx()/2 - textwidth("YOU LOSE!")/2,getmaxy()/2,"YOU LOSE!");
     break;
     }
     cleardevice();
     outtextxy(500,120,"COMPUTER'S TURN");
     drawmap();
     drawdice(530,300,temp);
     p1Draw(player_pos,comp_pos);
     delay(1000);
 }
 getch();
 delay(3000);
}
void run() {
 setbkcolor(BLACK);
 settextstyle(2,0,4);
 drawmap();
 p1();
}

void intro() {
 settextstyle(1,0,5);
 setfillstyle(1,1);
 bar(0,0,getmaxx(),80);
 bar(0,0,80,getmaxy());
 bar(0,getmaxy(),getmaxx(),getmaxy()-80);
 bar(getmaxx(),getmaxy(),getmaxx()-80,0);
 drawladder(1,99);
 drawladder(10,60);
 setbkcolor(CYAN);
 outtextxy(getmaxx()/2-textwidth("SNAKE AND LADDER")/2,10,"SNAKE AND LADDER");
 gotoxy(25,10);
 printf("ENTER YOUR NAME : ");
 gets(p1Name);
 }

void main() {
 int gd=DETECT,gm;
 initgraph(&gd,&gm,"C:\\TC\\BGI");
 randomize();
 intro();
 cleardevice();
 run();
 getch();
}
```

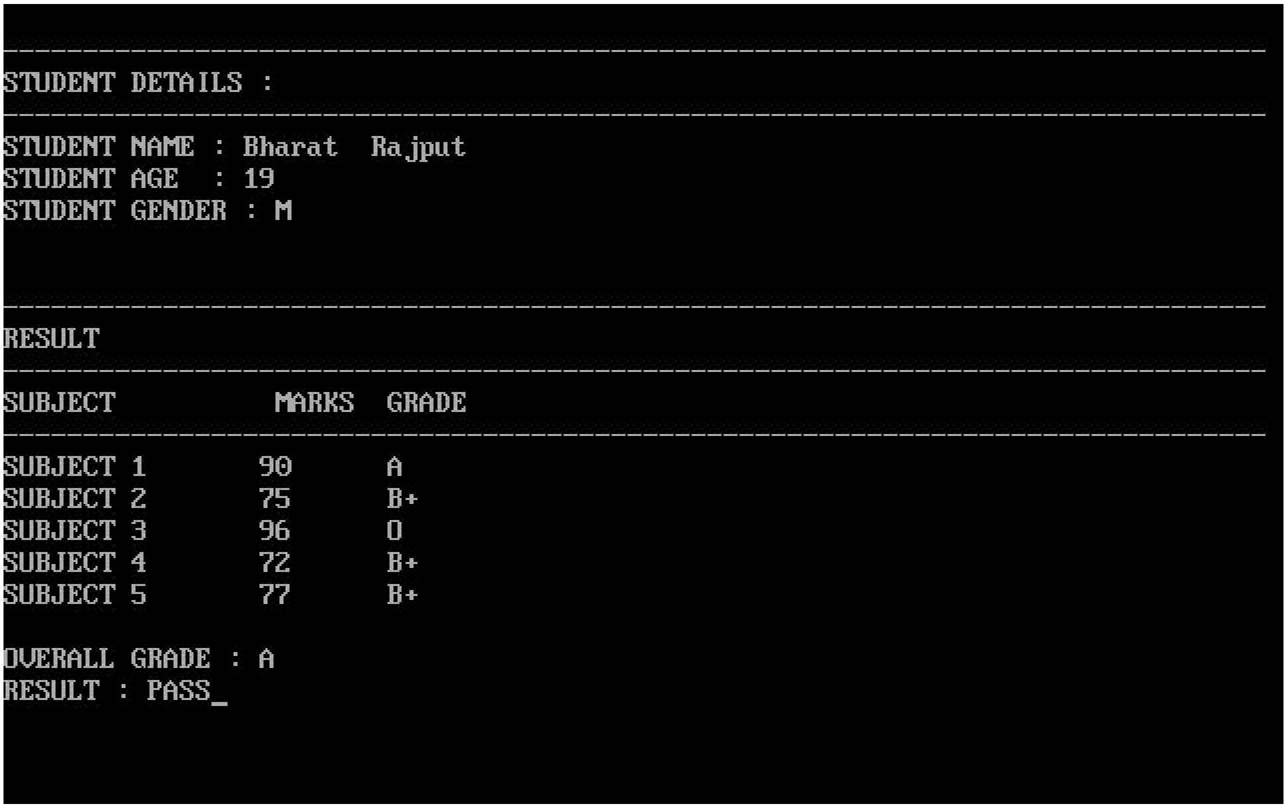
Q2. Write a program to input student details and print its result according to U.G. or P.G.



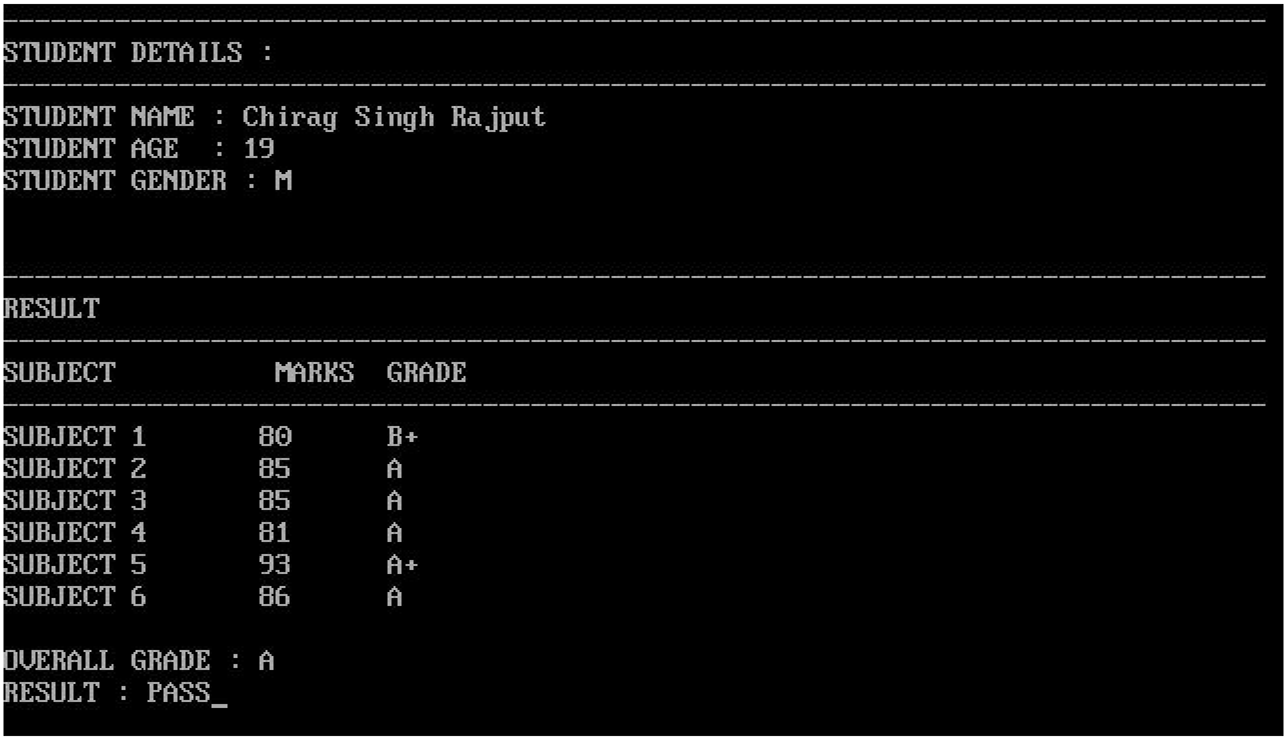
```
WELCOME TO NATIONAL P.G. COLLEGE
-------------------------------------------------------------------------------


1. U.G. STUDENT PORTAL
2. P.G. STUDENT PORTAL
3. EXIT

ENTER YOUR CHOICE :
```



```
U.G. STUDENT PORTAL
-------------------------------------------------------------------------------


ENTER YOUR DETAILS :
-------------------------------------------------------------------------------
ENTER FIRST NAME : Chirag Singh
ENTER LAST NAME : Rajput
ENTER AGE : 19
ENTER GENDER(M/F) : M

ENTER MARKS :
ENTER INTERNALS IN SUBJECT 1 : 72
INVALID MARKS!
ENTER INTERNALS IN SUBJECT 1 : 60
ENTER EXTERNALS IN SUBJECT 1 : 20
ENTER INTERNALS IN SUBJECT 2 : 65
ENTER EXTERNALS IN SUBJECT 2 : 20
ENTER INTERNALS IN SUBJECT 3 : 55
ENTER EXTERNALS IN SUBJECT 3 : 30
ENTER INTERNALS IN SUBJECT 4 : 56
ENTER EXTERNALS IN SUBJECT 4 : 25
ENTER INTERNALS IN SUBJECT 5 : 67
ENTER EXTERNALS IN SUBJECT 5 : 26
ENTER INTERNALS IN SUBJECT 6 : 62
ENTER EXTERNALS IN SUBJECT 6 : 24_
```



```
-------------------------------------------------------------------------------
STUDENT DETAILS :
-------------------------------------------------------------------------------
STUDENT NAME : Bharat  Rajput
STUDENT AGE  : 19
STUDENT GENDER : M


-------------------------------------------------------------------------------
RESULT
-------------------------------------------------------------------------------
SUBJECT          MARKS  GRADE
-------------------------------------------------------------------------------
SUBJECT 1       90      A
SUBJECT 2       75      B+
SUBJECT 3       96      O
SUBJECT 4       72      B+
SUBJECT 5       77      B+

OVERALL GRADE : A
RESULT : PASS_
```

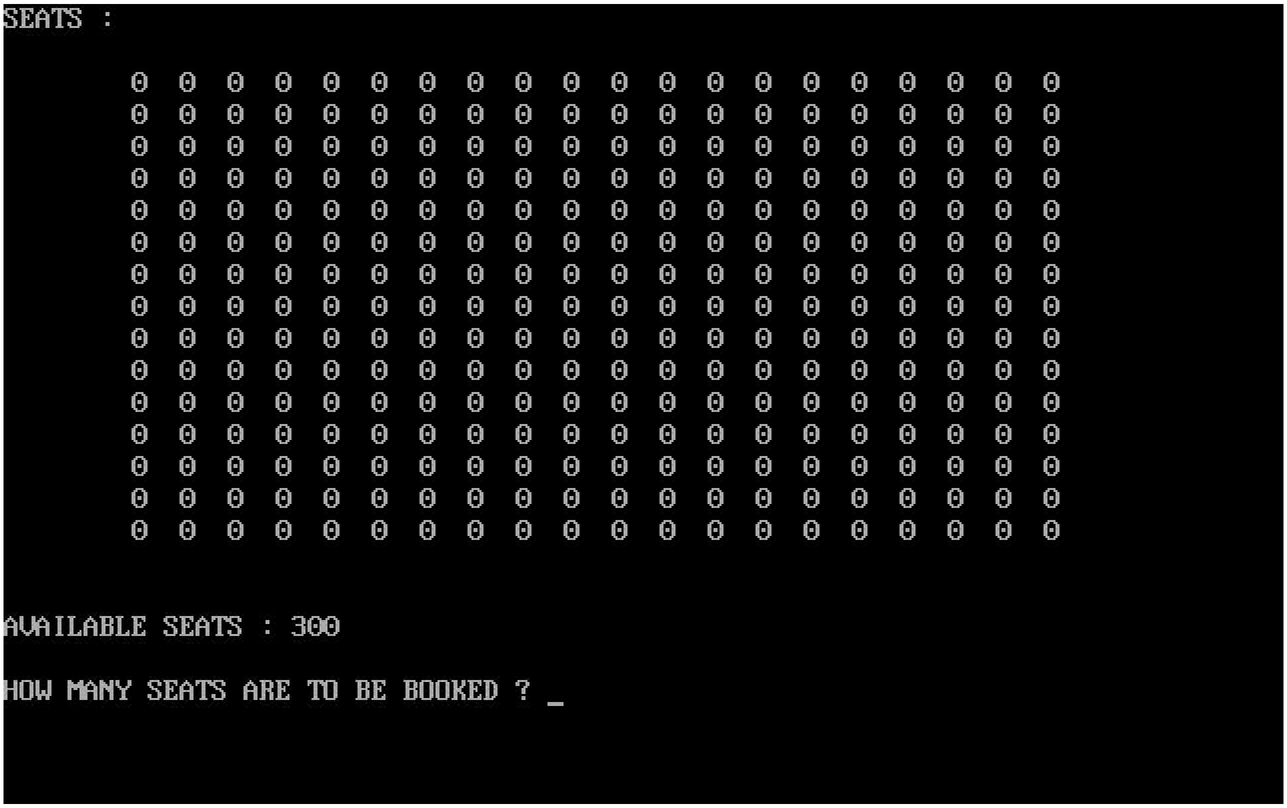


```
-------------------------------------------------------------------------------
STUDENT DETAILS :
-------------------------------------------------------------------------------
STUDENT NAME : Chirag Singh Rajput
STUDENT AGE  : 19
STUDENT GENDER : M


-------------------------------------------------------------------------------
RESULT
-------------------------------------------------------------------------------
SUBJECT          MARKS  GRADE
-------------------------------------------------------------------------------
SUBJECT 1       80      B+
SUBJECT 2       85      A
SUBJECT 3       85      A
SUBJECT 4       81      A
SUBJECT 5       93      A+
SUBJECT 6       86      A

OVERALL GRADE : A
RESULT : PASS_
```

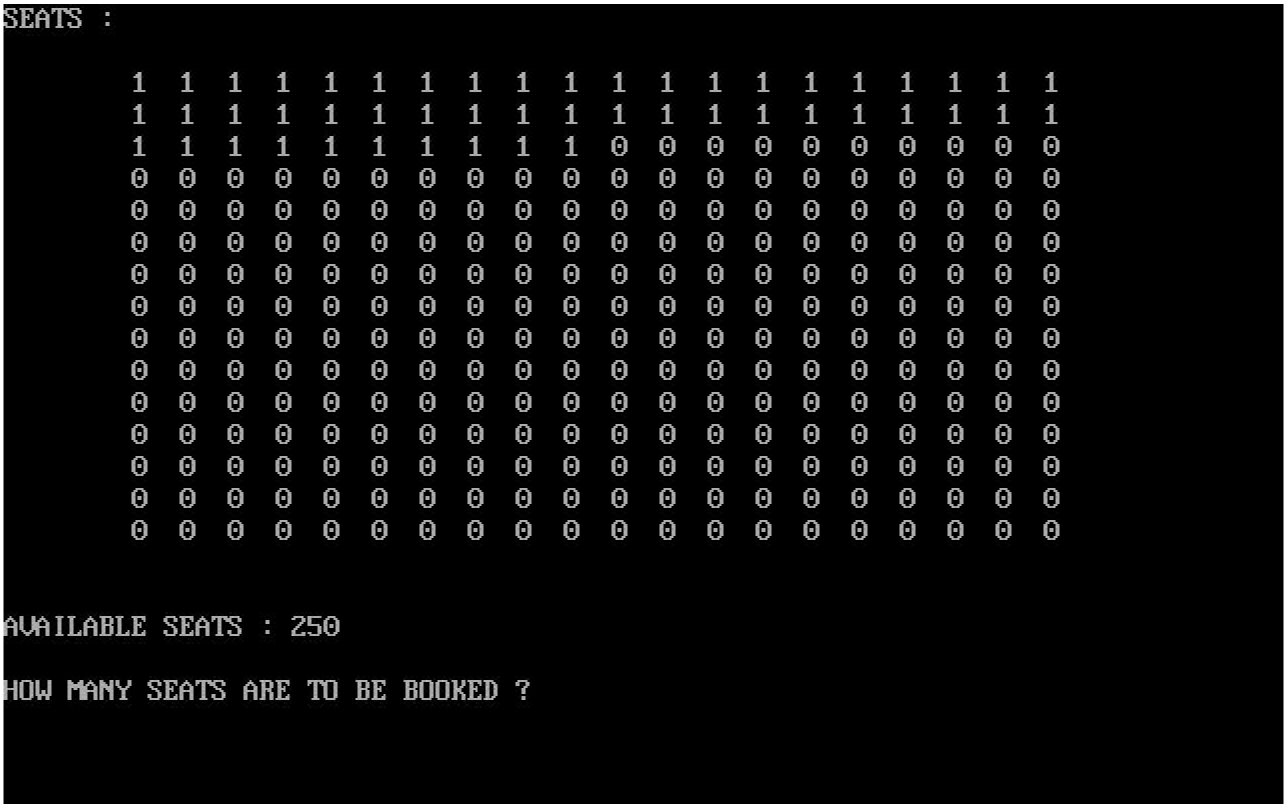
PROGRAM

```
#include <iostream.h>
#include <conio.h>
#include <stdlib.h>

void drawLine() {
 cout << endl;
 for(int i = 0 ; i < 79 ; i++) {
     cout << "-";
     }
     cout << endl;
 }

char * grades[] = {"O", "A+", "A" , "B+", "B", "C", "D", "F"};
int getGrades(int marks) {
 if(marks > 95) {
     return 0;
     } else if(marks > 90) {
      return 1;
           } else if(marks > 80) {
            return 2;
            } else if(marks > 70) {
                return 3;
                } else if(marks > 60) {
                 return 4;
                 } else if(marks > 50) {
                 return 5;
                 } else if(marks > 40) {
                     return 6;
                     } else return 7;
      }


class NPGC {
 protected :
           char  firstName[20];
           char  lastName[20];
           int age;
           char gender;
            int * internals;
           int * externals;
 public :
           void getDetails() {
           cout << "ENTER YOUR DETAILS : ";
           drawLine();
           cout << "ENTER FIRST NAME : ";
           cin.ignore();
           cin.getline(firstName,20);
           cout << "ENTER LAST NAME : ";
           cin.getline(lastName,20);
           cout << "ENTER AGE : ";
           cin  >> age;
           cout << "ENTER GENDER(M/F) : ";
           cin  >> gender;
           }
     void putDetails() {
                drawLine();
                cout << "STUDENT DETAILS : ";
                drawLine();
                cout << "STUDENT NAME : " << firstName << " " <<lastName <<
endl;
                cout << "STUDENT AGE  : " << age << endl;
                cout << "STUDENT GENDER : " << gender << endl << endl;
                      }
      };

     class UG : public NPGC {
      public  : void getMarks();
                void displayResult();
      };

      void UG :: getMarks() {
                internals = new int[6];
                externals = new int[6];
                cout << endl << "ENTER MARKS : " << endl;
                for(int i = 0 ; i < 6 ; i++) {
                cout << "ENTER INTERNALS IN SUBJECT " << i+1 << " : ";
                cin  >> internals[i];
                if(internals[i] > 70 || internals[i] < 0) {
                cout << "INVALID MARKS!" << endl;
                i--;
                continue;
                 }
                cout << "ENTER EXTERNALS IN SUBJECT " << i+1 << " : ";
                 cin  >> externals[i];
                 if(externals[i] > 30 || externals[i] < 0) {
                           cout << "INVALID MARKS!";
                           i--;
                           continue;
                      }
                }
           }

      void UG :: displayResult()  {
        int total = 0;
        drawLine();
        cout << "RESULT";
        drawLine();
        cout << "SUBJECT \t MARKS\tGRADE";
        drawLine();
        for(int i = 0 ; i < 6 ; i++) {
        int sum = internals[i] + externals[i];
         cout<<"SUBJECT "<<i+1<<"\t"<<sum<<"\t"<<grades[(getGrades(sum))]
<<endl;
                total += sum;
           }
           cout << endl << "OVERALL GRADE : " << grades[(getGrades(total/6))];
           cout << endl << "RESULT : ";
           (total/6 > 40)? cout << "PASS" : cout << "FAIL";
      }

      class PG : public NPGC {
      public  : void getMarks();
                void displayResult();
      };
      void PG :: getMarks() {
           internals = new int[5];
           externals = new int[5];
          cout << endl << "ENTER MARKS : " << endl;
           for(int i = 0 ; i < 5 ; i++) {
          cout << "ENTER INTERNALS IN SUBJECT " << i+1 << " : ";
           cin  >> internals[i];
           if(internals[i] > 80 || internals[i] < 0) {
            cout << "INVALID MARKS!" << endl;
            i--;
            continue;
           }
           cout << "ENTER EXTERNALS IN SUBJECT " << i+1 << " : ";
           cin  >> externals[i];
           if(externals[i] > 20 || externals[i] < 0) {
             cout << "INVALID MARKS!";
             i--;
             continue;
           }
        }
     }

      void PG :: displayResult()  {
        int total = 0;
        drawLine();
        cout << "RESULT";
        drawLine();
        cout << "SUBJECT \t MARKS\tGRADE";
        drawLine();
        for(int i = 0 ; i < 5 ; i++) {
          int sum = internals[i] + externals[i];

cout<<"SUBJECT"<<i+1<<"\t"<<sum<<"\t"<<grades[(getGrades(sum))]<<endl;
                total += sum;
           }
           cout << endl << "OVERALL GRADE : " << grades[(getGrades(total/5))];
     cout << endl << "RESULT : ";
     (total/5 > 40)? cout << "PASS" : cout << "FAIL";
 }

 void main() {
     short ch;
     while(1) {
      clrscr();
      cout << "WELCOME TO NATIONAL P.G. COLLEGE";
      drawLine();
      cout << endl << "1. U.G. STUDENT PORTAL" << endl;
      cout << "2. P.G. STUDENT PORTAL" << endl;
      cout << "3. EXIT" << endl << endl;
      cout << "ENTER YOUR CHOICE : ";
      cin  >> ch;
     switch(ch) {
      case 1 : UG student;
               clrscr();
               cout << "U.G. STUDENT PORTAL";
                drawLine();
               cout << endl;
                student.getDetails();
                student.getMarks();
                cout << endl << "PRESS ANY KEY TO DISPLAY RESULT..";
                getch();
                clrscr();
                student.putDetails();
                student.displayResult();
                break;
      case 2 :  PG studentPG;
                clrscr();
                cout << "P.G. STUDENT PORTAL";
                drawLine();
                cout << endl;
                studentPG.getDetails();
                studentPG.getMarks();
               cout << endl << "PRESS ANY KEY TO DISPLAY RESULT..";
                getch();
               clrscr();
                studentPG.putDetails();
                studentPG.displayResult();
                break;
     case 3 : exit(0);
     default : cout << "WRONG CHOICE!";
                                break;
     }
     getch();
 }
}
```

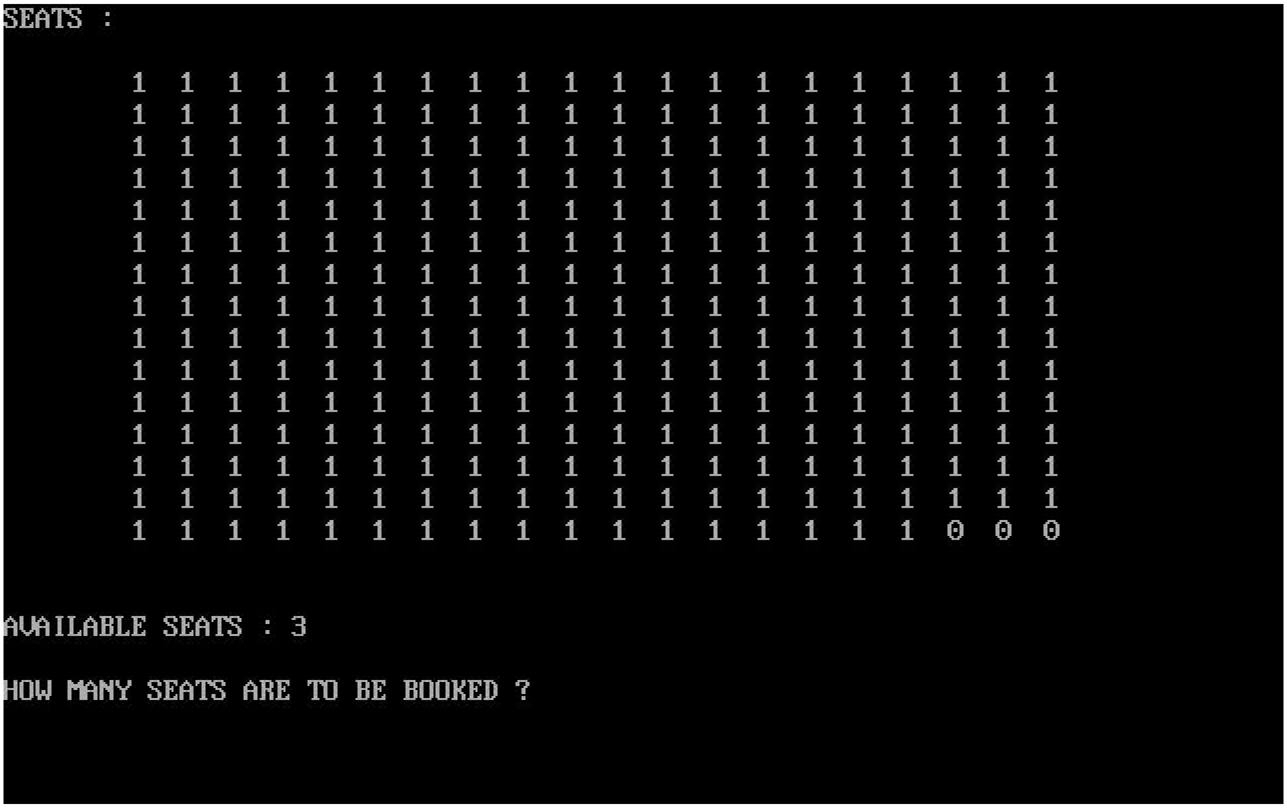
Q3. Write a program to reserve seats in a bus.



```
SEATS :

        0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
        0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
        0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
        0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
        0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
        0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
        0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
        0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
        0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
        0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
        0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
        0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
        0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
        0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
        0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0


AVAILABLE SEATS : 300

HOW MANY SEATS ARE TO BE BOOKED ? _
```



```
SEATS :

        1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
        1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
        1  1  1  1  1  1  1  1  1  1  0  0  0  0  0  0  0  0  0  0
        0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
        0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
        0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
        0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
        0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
        0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
        0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
        0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
        0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
        0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
        0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
        0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0


AVAILABLE SEATS : 250

HOW MANY SEATS ARE TO BE BOOKED ?
```



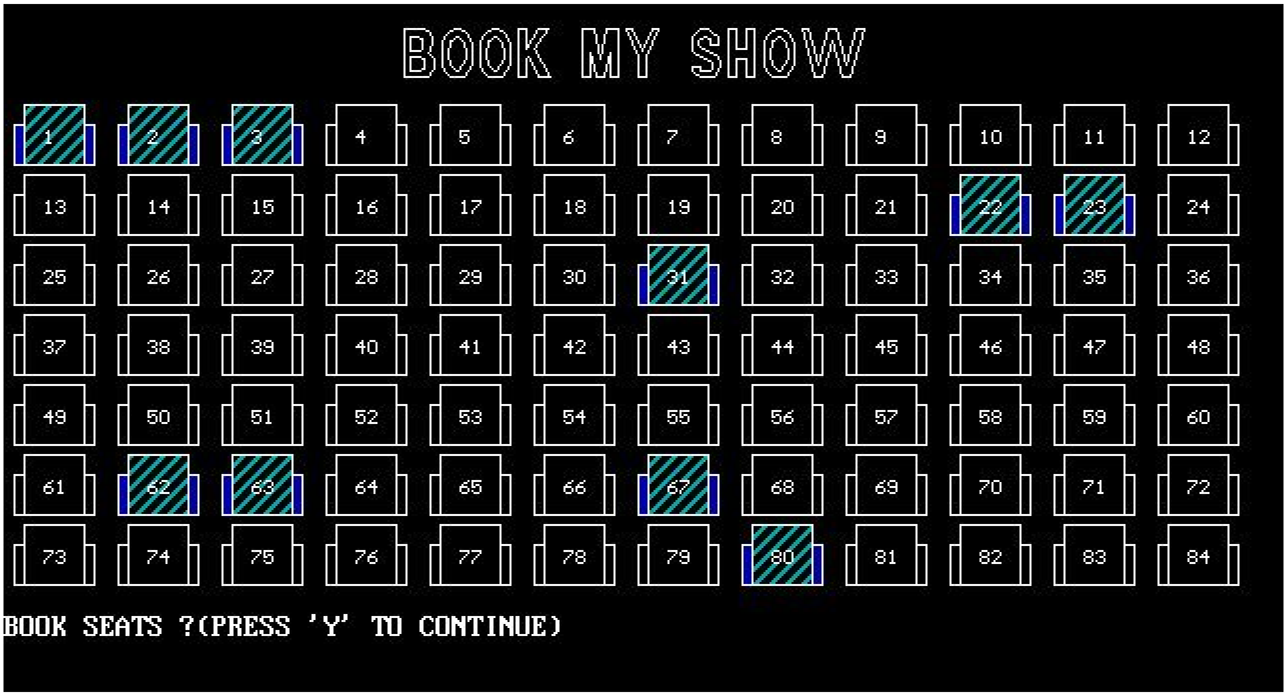
```
SEATS :

       1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
       1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
       1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
       1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
       1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
       1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
       1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
       1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
       1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
       1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
       1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
       1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
       1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
       1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
       1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  0  0


AVAILABLE SEATS : 3

HOW MANY SEATS ARE TO BE BOOKED ?
```

PROGRAM

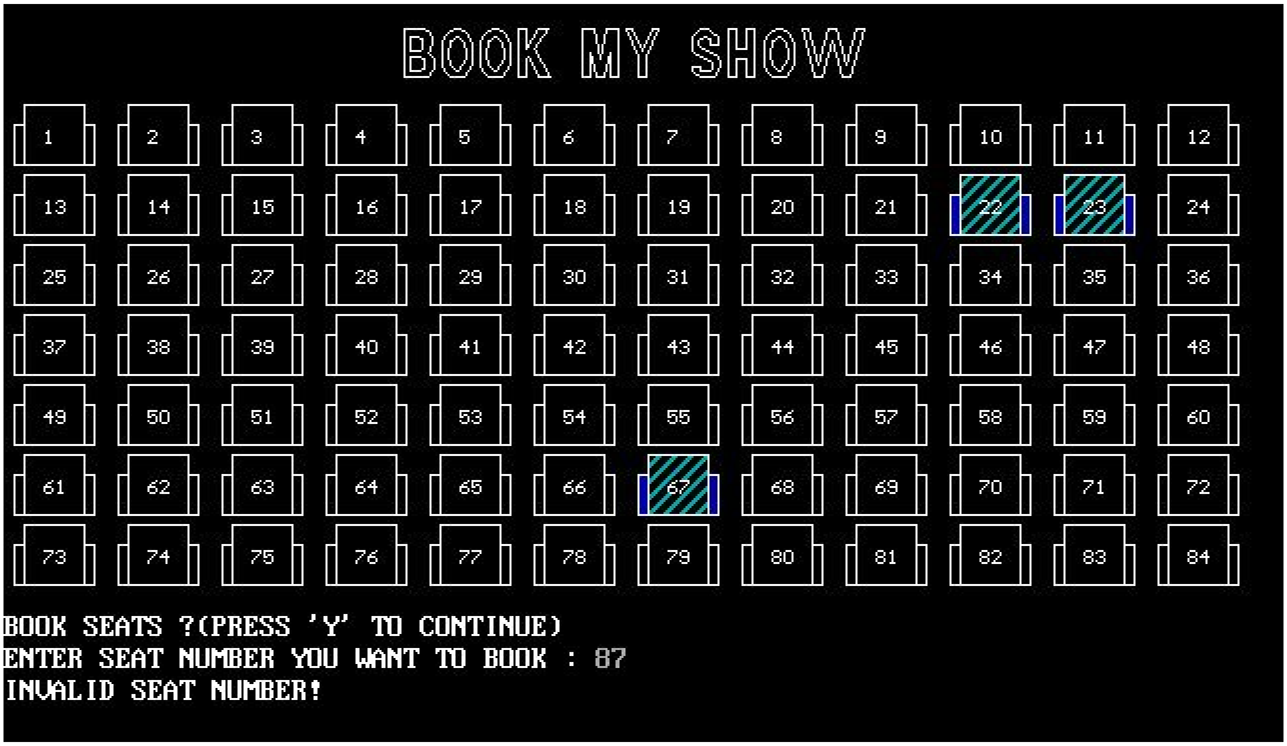
```
#include<iostream.h>
#include<conio.h>
#define ROWS 15
#define COLS 10
class Bus {
      int seats[ROWS][COLS];
      int booked;
      public :
      void initSeats();
      void drawSeats();
      void bookSeats();
      };

void Bus :: bookSeats() {
 int count = ROWS * COLS,booking,i = booked/ROWS,j=booked%COLS;
 cout << "\n\nAVAILABLE SEATS : "<< count - booked << endl;
 cout<< "\nHOW MANY SEATS ARE TO BE BOOKED ? ";
 cin >> booking;
 if(booking < 0 || booking > count - booked) {
     cout << "INVALID ENTRY!" ;
     return;
 } else {
```

```
           booked += booking;
           while(booking--) {
           seats[i][j%COLS] = 1;
           if(++j == COLS) {
            j = 0;
            i++;
            }
           }
      }
     }

void Bus :: initSeats() {
 for(int i = 0 ; i < ROWS ; i++) {
     for(int j = 0 ; j < COLS ; j++) {
      seats[i][j] = 0;
      }
     }
 booked = 0;
}

void Bus :: drawSeats() {
clrscr();
 cout << "SEATS : " << endl<<endl;
 for(int i = 0 ; i < ROWS ; i++) {
     cout << "        ";
     for(int j = 0 ; j < COLS ; j++) {
      cout << seats[i][j] << "  ";
      }
      cout << endl;
     }
 }
void main() {
 Bus b;
 int gd = 0 , gm;
 b.initSeats();
 do {
   clrscr();
   b.drawSeats();
   b.bookSeats();
   cout << "BOOK MORE? (PRESS Y TO CONTINUE)...";
  while(getch() == 'y');

}
```

Q4. Write a program to book seats in a theater



```
BOOK MY SHOW

 1   2   3   4   5   6   7   8   9  10  11  12
13  14  15  16  17  18  19  20  21  22  23  24
25  26  27  28  29  30  31  32  33  34  35  36
37  38  39  40  41  42  43  44  45  46  47  48
49  50  51  52  53  54  55  56  57  58  59  60
61  62  63  64  65  66  67  68  69  70  71  72
73  74  75  76  77  78  79  80  81  82  83  84

BOOK SEATS ?(PRESS 'Y' TO CONTINUE)
```

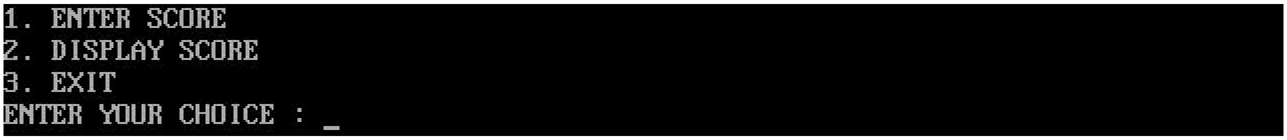


```
BOOK MY SHOW

 1   2   3   4   5   6   7   8   9  10  11  12
13  14  15  16  17  18  19  20  21  22  23  24
25  26  27  28  29  30  31  32  33  34  35  36
37  38  39  40  41  42  43  44  45  46  47  48
49  50  51  52  53  54  55  56  57  58  59  60
61  62  63  64  65  66  67  68  69  70  71  72
73  74  75  76  77  78  79  80  81  82  83  84

BOOK SEATS ?(PRESS 'Y' TO CONTINUE)
ENTER SEAT NUMBER YOU WANT TO BOOK : 87
INVALID SEAT NUMBER!
```

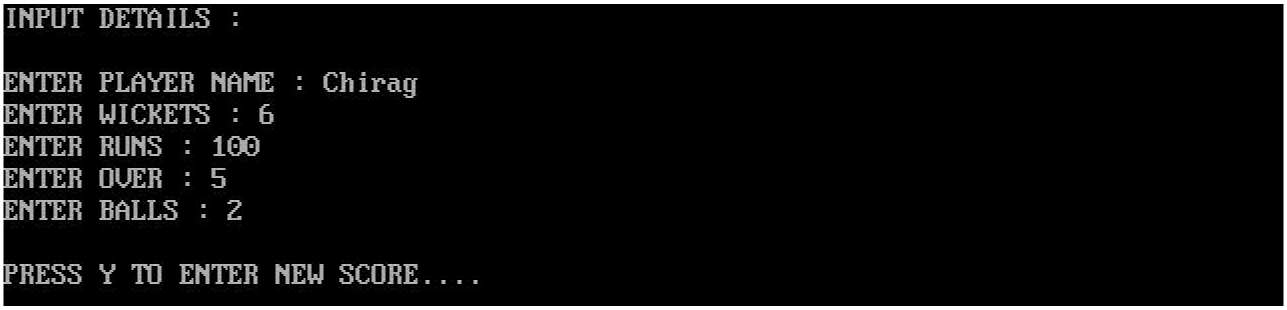
PROGRAM

```
#include<iostream.h>
#include<graphics.h>
#include<conio.h>
#define SEATCOUNT 84
#include<dos.h>
#include<stdlib.h>
class seats {
            int seat[SEATCOUNT];
            public : void drawSeats();
                     void bookSeats();
                     void initSeats();
                     void seatDig(int, int, int);
           };

void seats :: initSeats() {
 for(int i = 0 ; i < SEATCOUNT ; i++) {
     seat[i] = 0;
     }
 }

void seats :: bookSeats() {
 int booking,ch;
 gotoxy(1, 20);
 cout<<"BOOK SEATS ?(PRESS 'Y' TO CONTINUE)";
 ch = getch();
 if(ch == 'y' || ch == 'Y') {
 cout<<endl<<"ENTER SEAT NUMBER YOU WANT TO BOOK : ";
 cin>>booking;
 if(seat[booking-1] != 0) {
      cout<<"SEAT NOT AVAILABLE!";
      } else if(booking > SEATCOUNT) {
            cout <<"INVALID SEAT NUMBER!";
            } else {
                 cout<<"SEAT BOOKED SUCCESSFULLY!";
                 seat[booking-1] = 1;
                 }
            } else {
                exit(0);
                }

 }
 void seats :: seatDig(int x, int y, int count) {
     char number[5];
     itoa(count+1,number,10);
     rectangle(x, y ,x + 30, y + 30);
     rectangle(x-5,y+10,x,y+30);
     rectangle(x+30,y+10,x+35,y+30);
     outtextxy(x+10,y+10,number);
     if(seat[count] == 1) {
      setfillstyle(4,3);
      floodfill(x+1,y+1,WHITE);
      setfillstyle(1,1);
      floodfill(x-4,y+11,WHITE);
      floodfill(x+34,y+11,WHITE);
      }
     }

void seats :: drawSeats() {
 int x , y = 50, counter = 0;
 settextstyle(10,0,1);
 outtextxy(getmaxx()/2-textwidth("BOOK MY SHOW")/2,0,"BOOK MY SHOW");
 settextstyle(2,0,0);
 for(int i = 0 ; i < SEATCOUNT ; i++) {
 delay(10);
 x = 10  + counter++*52;
 if(i % 12 == 0 && i != 0)  {
     y += 35;
     counter = 1;
     x = 10;
     }
     seatDig(x,y,i);
 }
}

int main() {
 seats obj;
 int gd = 0,gm;
 initgraph(&gd,&gm,"C:\\TC\\BGI");
 obj.initSeats();
 while(1) {
 cleardevice();
 obj.drawSeats();
 obj.bookSeats();
 getch();
 }
 getch();
 }
```

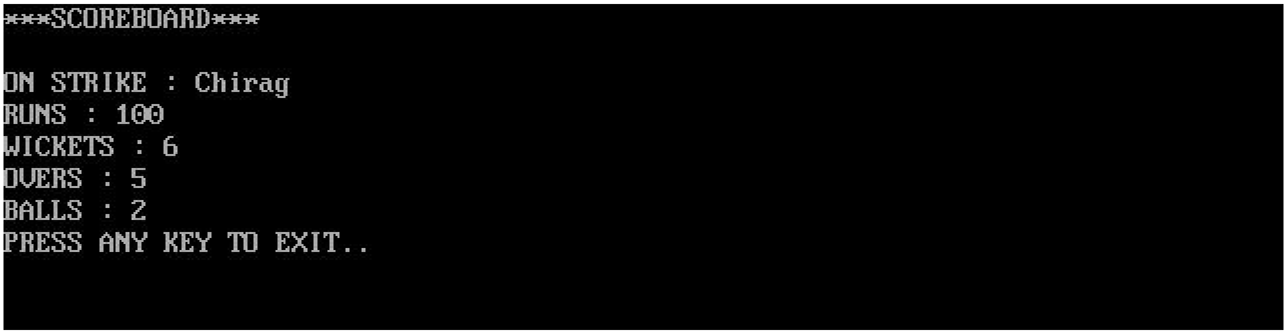
Q.5 Write a program to create a scoreboard and update it.



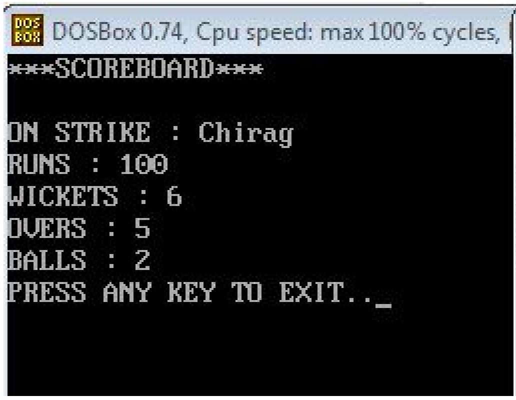
```
1. ENTER SCORE
2. DISPLAY SCORE
3. EXIT
ENTER YOUR CHOICE : _
```



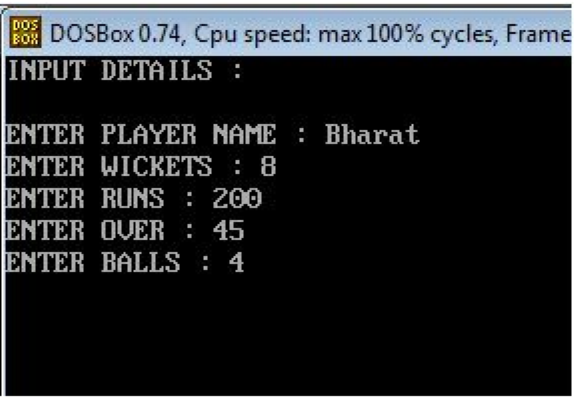
```
INPUT DETAILS :

ENTER PLAYER NAME : Chirag
ENTER WICKETS : 6
ENTER RUNS : 100
ENTER OVER : 5
ENTER BALLS : 2

PRESS Y TO ENTER NEW SCORE....
```



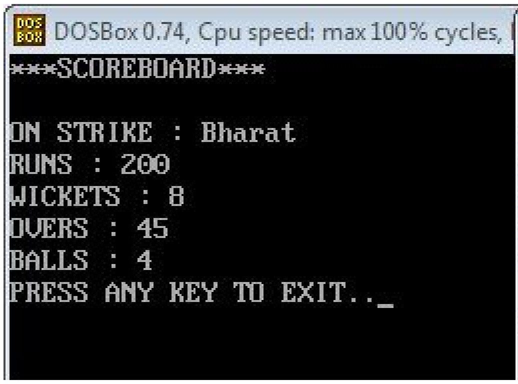
```
***SCOREBOARD***

ON STRIKE : Chirag
RUNS : 100
WICKETS : 6
OVERS : 5
BALLS : 2
PRESS ANY KEY TO EXIT..
```



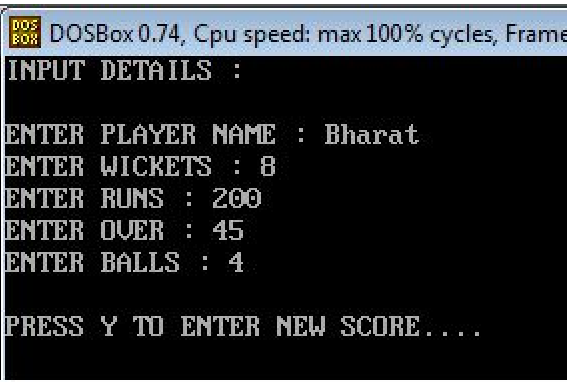
```
***SCOREBOARD***

ON STRIKE : Chirag
RUNS : 100
WICKETS : 6
OVERS : 5
BALLS : 2
PRESS ANY KEY TO EXIT.._
```



```
INPUT DETAILS :

ENTER PLAYER NAME : Bharat
ENTER WICKETS : 8
ENTER RUNS : 200
ENTER OVER : 45
ENTER BALLS : 4
```



```
***SCOREBOARD***

ON STRIKE : Bharat
RUNS : 200
WICKETS : 8
OVERS : 45
BALLS : 4
PRESS ANY KEY TO EXIT.._
```



```
INPUT DETAILS :

ENTER PLAYER NAME : Bharat
ENTER WICKETS : 8
ENTER RUNS : 200
ENTER OVER : 45
ENTER BALLS : 4

PRESS Y TO ENTER NEW SCORE....
```

PROGRAM

```
#include<fstream.h>
#include<iostream.h>
#include<conio.h>
#include<stdlib.h>
#include<dos.h>
class GAME {
                      char name[20];
                      int wickets;
                      int runs;
                      int over;
                      int balls;
                      public :
                      void getDetails();
                      void updateDetails();
                      void printDetails();
                      };

void GAME :: printDetails() {
 fstream file;
 file.open("GAME.txt", ios :: in);
 char ch,buffer[100],i = 0;
 if(file.fail()) {
     return;
     }
 file.get(ch);
 while(ch != EOF) {
     buffer[i++] = ch;
     file.get(ch);
 }
 buffer[i] = '\0';
 cout << buffer;
 file.close();
}

void GAME :: getDetails() {
 cout << "INPUT DETAILS : " << endl << endl;
 cout << "ENTER PLAYER NAME : ";
 cin.getline(name,20);
 cout << "ENTER WICKETS : ";
 cin  >> wickets;
 cout << "ENTER RUNS : ";
 cin  >> runs;
 cout << "ENTER OVER : ";
 cin  >> over;
 cout << "ENTER BALLS : ";
 cin  >> balls;
 }

void GAME :: updateDetails() {
```

```
fstream fp;
fp.open("GAME.TXT", ios :: out | ios :: trunc);
fp << "***SCOREBOARD***" << endl << endl;
fp << "ON STRIKE : " << name << endl;
fp << "RUNS : "<< runs <<  endl;
fp << "WICKETS : " << wickets << endl;
fp << "OVERS : " << over << endl;
fp << "BALLS : " << balls << endl;
fp.close();
}

void main() {
    GAME obj;
    int ch;
    clrscr();
    cout << "1. ENTER SCORE" << endl;
    cout << "2. DISPLAY SCORE" << endl;
    cout << "3. EXIT" << endl;
    cout << "ENTER YOUR CHOICE :";
    cin >> ch;
    switch(ch) {
     case 1 : do {
               clrscr();
               obj.getDetails();
               obj.updateDetails();
               cout << "PRESS Y TO ENTER NEW SCORE";
               } while(getch() == 'y');
               break;
    case 2 :  while(!kbhit()) {
               clrscr();
                obj.printDetails();
               cout << "PRESS ANY KEY TO EXIT..";
               delay(1000);
                               }
    case 3 :
    default : exit(0);
    }
}
```