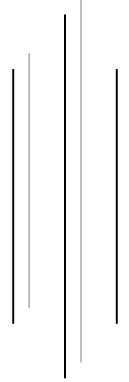
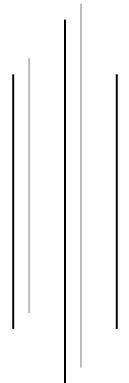




TRIBHUWAN UNIVERSITY
INSTITUTE OF ENGINEERING
Pulchowk Campus, Lalitpur



Report on the mini project work of computer
on the topic
“Death Race”



SUBMITTED BY:

- Prabin Bhandari (070/BCT-525)
- Sagar Adhikari (070/BCT-533)
- Sagar Aryal (070/BCT-534)

SUBMITTED TO:

Department of Electronics and
Computer

ACKNOWLEDGEMENT

We would like to express our special thanks of gratitude to our teacher Mr. Baburam Dawadi Sir as well as our department of electronics and computer engineering who gave us the golden opportunity to do this wonderful project of C programming on the topic of Death Race, which also helped us in doing a lot of Research and study about c-programming language and we came to know about so many new things. Secondly, we would also like to thank our friends and seniors who helped us a lot in finishing this project within the limited time.

Finally, we would like to provide our sincere gratitude to all those who helped us directly and indirectly in completing this project in an efficient way.

TABLE OF CONTENTS

	<u>Topics</u>	<u>Page No.</u>
1.	Abstract of the Project.....	1
2.	Background.....	2
3.	Problem Analysis.....	3
4.	Algorithm.....	4
5.	Flowchart.....	14
6.	Code.....	29
7.	Compilation, Testing and Debugging.....	59
8.	Output Snapshots.....	60
9.	Discussion and Conclusions.....	65

1. Abstract of the Project

This project is a mini concept of the car racing game. Even though high quality graphics is not used (i.e. turbo c graphics is used), this game would be interesting to play and this project (i.e. game) mainly focuses using the core 'C language' (i.e. turbo c). This game has quite similar features that the other car racing games we play. Users would enjoy playing this game. In this game, users just tries to avoid the collision with oncoming cars. The score which is displayed on the screen, increases when game progresses and the game is over if it hits with any in coming cars.

And, talking about the core about the project, this project features modular programming. In other words, this project has been divided into different files and linked each other. This project is divided into different modules and functions so that the debugging process becomes easier and the program features readability.

Also to include, scores secured by the user while playing is stored in another data file. The user just have to enter his/her name. In addition to it, anyone can view the scores secured by the played users in the game.

Hence, we have utilized out full efforts in building this project efficiently and to resemble with the real life playing games.

2. Background

This project was primarily designed for a semester project for “Computer Programming”. This project, as is a semester project, we did not try to make it for real purpose or to be sold in the market. This project is just an imitation of the real world project which is created utilizing our programming skills in C and make it resemble a lot and gain the similar experience as we play the real-life games.

So, Our Project titled “DEATH RACE”, is a simple car racing game developed using C language. As we already mentioned above, we utilized our programming knowledge and skills that we have learned so far in C from our teacher and seniors to create this game.

Our team will never state that this is total and complete game and bug free by itself at its present version. But, our team believes that this game can be a good way of presenting and utilizing those programming skills and hopes that our juniors would get a helpful idea of creating qualitative programs and improving this version of game in future.

We, our team, decided to create a game with graphics and sound in terms that we could use our programming skills attractively. Well, to talk about potential to create games, we were well acknowledged that it is difficult to create the real-world games (as we play in our computers or psps, etc.), however we determined to create a game a lot simpler than the real-world games but resemble more to those so that we could have fun playing the game.

And, thus, we hope that this game can be a good demonstration of utilizing the programming skills and it can also be helpful and entertaining who plays or want to gain some knowledge and analyze the code.

Anyone wishing to improve and upgrade the program is very welcome to do so.

3. Problem Analysis

To create a successful project (or, a program), a precise analysis of various elements of the problem must be done. Here, to develop a game, we must have the knowledge of various obstacles to create it.

In this program, to develop a racing game, the inputs are mostly taken in character form, i.e. the arrow keys (up, down, left, right keys).

The outputs of the program is rather displayed using the graphical elements.

So, to proceed specifically, into each modules, we analyze each steps thoroughly.

As, in the main page, a series of short animation are to be displayed using graphical elements. And, a menu is displayed where various options are displayed to proceed to the game. And for input, up, down keys are used to select individual options and enter key to select that option which are prompt as a character.

After that, proceeding to the game, we should prompt user a key to start the game which is taken in character form. Then, for the game, a car shape is to be drawn, and is made using the graphical elements (like, line, rectangle, colors). Then, the construction of the road is to be made using rectangle and filling it the color which are all in graphical forms. After that, arrow keys are used to move the car but only sideways. And to do so, we should prompt the user a character with the ASCII values of the arrow keys. Then, we call other cars (i.e. objects made using graphical elements) whose y-coordinates are increased continuously. After that, when the user crashes, we prompt the user to enter his name to store the record. And we prompt the user a character whether to continue the game or not. If yes, the game is continued from the beginning or else, the menu page (i.e. function) is to be called.

Besides, playing the game, we want the user to view their records. So, to do it, a file is created to store the records. And, to view it, a function named (for, e.g. Highscore) is called. In this functions no inputs are taken. Rather the score and name obtained from the game is used to display. It means the name is displayed in character form and score in integer form on the screen.

Also, the program features the instructions to play the game. A function named help is called to see the instructions. Like above function, this function too does not take any inputs from the user, and instructions and graphical elements are displayed on the screen.

Then, to exit from the program, user has the option 'exit'. When exit option is selected using return key, an exit function is called which exits the program.

4. Algorithm

Algorithm for main()

- Step 1: Start
- Step 2: Initialize the graphics.
- Step 3: Call the function sound().
- Step 4: Delay 100 milliseconds
- Step 5: Call the function display().
- Step 6: Call the function startpage1().
- Step 7: Stop Graphics.
- Step 8: End the program.

Algorithm for function startpage1()

- Step 1: Start
- Step 2: Declare ch1 and a as character.
- Step 3: Set color white and create a basic rectangle throughout the screen to display the contents of the page.
- Step 4: Display Start, High Scores, Help and exit with font size 5 and on the center of screen with all them displayed in the different lines.
- Step 5: Display a line under start.
- Step 6: Read a character from the user a.
- Step 7: convert a to its ascii value
- Step 8: If a=13 go to step 9 else if a=0 go to step 10 else go to step 12
- Step 9: Call function game();

- Step 10: Read ch1
- Step 11: If ch1=80 call function startpage2()
Else If ch1=72 call function statpage4()
Else if ch1=27 exit the program
Else call the function startpage1()
- Step 12: Call function startpage1()
- Step 13: End the Program

Algorithm for function startpage2()

- Step 1: Start
- Step 2: Declare ch1 and a as character.
- Step 3: Set color white and create a basic rectangle throughout the screen to display the contents of the page.
- Step 4: Display Start, High Scores, Help and exit with font size 5 and on the center of screen with all them displayed in the different lines.
- Step 5: Display a line under High Scores.
- Step 6: Read a character from the user ch1.
- Step 7: convert ch1 to its ascii value
- Step 8: If a=13 go to step 9 else if a=0 go to step 10 else go to step 12
- Step 9: Call function Drecord();
- Step 10: Read ch1
- Step 11: If ch1=80 call function startpage3()
Else If ch1=72 call function statpage1()
Else if ch1=27 exit the program

- Else call the function startpage2()
Step 12: Call function startpage2()
Step 13: End the Program

Algorithm for function startpage3()

- Step 1: Start
- Step 2: Declare ch1 and a as character.
- Step 3: Set color white and create a basic rectangle throughout the screen to display the contents of the page.
- Step 4: Display Start, High Scores, Help and exit with font size 5 and on the center of screen with all them displayed in the different lines.
- Step 5: Display a line under Help.
- Step 6: Read a character from the user ch1.
- Step 7: convert ch1 to its ascii value
- Step 8: If a=13 go to step 9 else if a=0 go to step 11 else go to step 12
- Step 9: Clear the screen and display game tips. Then display Press P to pause at any time and press any key to resume. Enter Esc key to exit from the game. The speed of your car is increased as you proceed. So greater difficulty is on the way as you proceed. PRESS THE ARROW KEYS TO MOVE THE CAR. Press enter key to return to the main menu.
- Step 10: Take and character input and then call function startpage3()
- Step 11: Read ch1
- Step 12: If ch1=80 call function startpage4()
Else If ch1=72 call function statpage2()
Else if ch1=27 exit the program
Else call the function startpage3()

Step 12: Call function startpage3()

Step 13: End the Program

Algorithm for function startpage4()

Step 1: Start

Step 2: Declare ch1 and a as character.

Step 3: Set color white and create a basic rectangle throughout the screen to display the contents of the page.

Step 4: Display Start, High Scores, Help and exit with font size 5 and on the center of screen with all them displayed in the different lines.

Step 5: Display a line under exit.

Step 6: Read a character from the user ch1.

Step 7: convert ch1 to its ascii value

Step 8: If a=13 go to step 9 else if a=0 go to step 11 else go to step 12

Step 9: Clear the screen and display Thanks for playing. Press any key to exit.

Step 10: Take and character input and then exit the function.

Step 11: Read ch1

Step 12: If ch1=80 call function startpage1()

Else If ch1=72 call function statpage3()

Else if ch1=27 exit the program

Else call the function startpage4()

Step 13: Call function startpage4();

Step 14: End the Program

Algorithm for function loading()

- Step 1: Start
- Step 2: Declare I, a and per=0 as integer
- Step 3: For I= 0 to 200 go to step 4 else go to step 13
- Step 4: Set the background color of 9 and set color of 7 for other things to displayed.
- Step 5: Display loading at the center of screen.
- Step 6: Draw a rectangle around loading.
- Step 7: Delay for 25 milli second.
- Step 8: Set color red
- Step 9: if I mod 2 =0 go to step 10 else go to 12
- Step 10: Display Per
- Step 11: Per=Per+1
- Step 12: i=i+1 and go to step 3
- Step 13: Display Press and key to start the game.
- Step 14: Take a character input from the user
- Step 15: End

Algorithm for function soundd()

- Step 1: Start
- Step 2: Declare i as integer.
- Step 3: Set color 13 and draw a basic rectangle with background white to display the contents.
- Step 4: for i = 1 to 4 go to step 5 else go to step 16.

- Step 5: put sound of $i*50$
- Step 6: Set color of i
- Step 7: Set font size of i
- Step 8: Display PSS on the co-ordinate 50,100
- Step 9: Display Games on the co-ordinate 100,200
- Step 10: Display Presents on the co-ordinate 150,300
- Step 11: Set Color white
- Step 12: Delay 1 second
- Step 13: Display PSS on the co-ordinate 50,100
- Step 14: Display Games on the co-ordinate 100,200
- Step 15: Display Presents on the co-ordinate 150,300
- Step 16: $i=i+1$ and go to step 4
- Step 17: End the sound
- Step 18: End the Program.

Algorithm for function display()

- Step 1: Start
- Step 2: declare i , $x=0$ and $s=1$ as integers.
- Step 3: for $i = 1$ to 7 go to step 4 else go to step 15
- Step 4: Set background color white.
- Step 5: Display sound of frequency $i*500$.
- Step 6: Set color to i .
- Step 7: Display Death at co-ordinate 50,100.
- Step 8: Display Race at co-ordinate 150,300.

- Step 9: Display rectangle at co-ordinates $0+x, 200, 100+x, 250$
- Step 10: Display Circle with radius 10 and center at $25+x, 260$.
- Step 11: Display Circle with radius 10 and center at $70+x, 260$.
- Step 12: $x=x+100$
- Step 13: Delay 250 millisecond.
- Step 14: $s=s+1$ and $i=i+1$ and go to step 2
- Step 15: Stop the sound.
- Step 16: set back ground color of black.
- Step 17: End the Program.

Algorithm for function game()

- Step 1: Start
- Step 2: Call function Loading
- Step 3: Call function animation after a key is pressed
- Step 4: End

Algorithm for function animation()

- Step 1: Start
- Step 2: Store image of car in memory using getimage function.
- Step 3: Clear the image of car from screen.
- Step 4: Draw white strips on black road on the screen and store this image to the memory.
- Step 5: Clear the image of road from screen.
- Step 6: Get two different random values among 1,2,3.

- Step 7: Create an infinite loop which ends only for certain conditions
Inside the infinite loop
- Step 8: Display score at the left side of the screen.
- Step 9: Check current score and assign speed to the car as per the current score (so that difficulty increases continuously when the score increases).
- Step 10: Restore the image of white strips on black road from the memory and display the screen.
- Step 11: Scroll down white strips with time with delay 50 ms to create sensation of road moving.
- Step 12: If the random values obtained are 2 and 3.
- a. Display the other car on road at lane 2 and 3 with continuously increasing coordinates (as per the variable that shows speed of the car
 - b. If coordinates of other car reaches the coordinates of player's car on the same lane (2 and 3), then show signal of crash
- Call function record with argument as current score
- Ask for player to continue or quit
- If user responds yes goto start else call startpage());
- c. If user's car on lane 1, and the other car's coordinates became out of the screen
- Increase the score variable used in the program to increase the speed
- d. Ask for two different value among 1,2,3.
- Step 13: If the random values obtained are 1 and 3, or 1 and 2, follow the same procedure as done above.
- Step 14: Display the position of player's car on lane 1,2 or 3 according to variable 'posn' in the given program.
- Value of 'posn' changes with user pressing left or right arrow

- Step 15: Use kbhit function to respond to the left or right arrow key press and change the position of player's
Car position according to it. Using (posn=posn+1 for right arrow key press and posn=posn-1 for left arrow key press)
- Step 16: If kbhit() notices the user pressed 'Esc' key then ask if sure to exit the game
- Step 17: If kbhit() notices the user pressed 'P' then pause the screen until any other character is pressed
- Step 18: If value of 'posn' becomes greater than 3 , assign posn=3 , and if value of 'posn' becomes less than 1 assign posn=1
- Step 19: Increase score of player by 1
- Step 20: Continue the loop.
- Step 21: End the Program.

Algorithm for function Drecord()

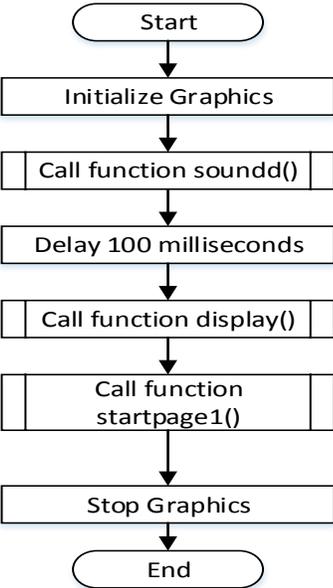
- Step 1: Start
- Step 2: Initialize a File pointer.
- Step 3: Open the record file for only reading purpose.
- Step 4: Check if the file opens successfully or not.
- Step 5: If file open fails, then print error message and exit the function. Else go to Step 6.
- Step 6: Check if there are any contents. If 'yes' print the content of the file using while statement until the End of the file. Else print no any records.
- Step 7: Close the file associated with the stream and disassociate it.
- Step 8: End

Algorithm for function Record ()

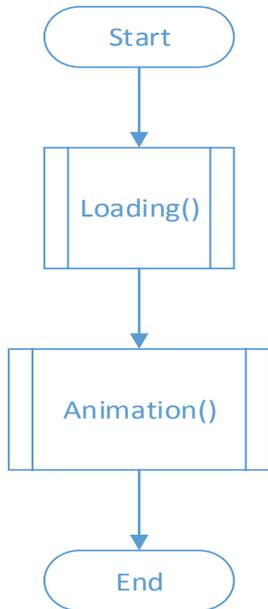
- Step 1: Start
- Step 2: Initialize the file pointer.
- Step 3: Open the file for appending purpose i.e. read/ write.
- Step 4: Check if the file opens successfully or not.
- Step 5: If file open fails, then print error message and exit the function.
- Step 6: Set the color into "WHITE" and change the text style using 'settextstyle' function.
- Step 7: Print "CRASH" on the screen.
- Step 8: Delay it for 2 seconds and clear the screen.
- Step 9: Then, ask the user for their name.
- Step 10: Change the first letter of the word to Uppercase if necessary.
- Step 11: Then if the string is valid, write it to the file using fputs() and also their respective scores using fprintf() function.
- Step 12: Close the file associated with the stream and disassociate it.
- Step 13: Clear the screen, and print "RECORD SAVED" on the screen.
- Step 14: Delay it for 1 second.
- Step 15: End

5. Flowchart

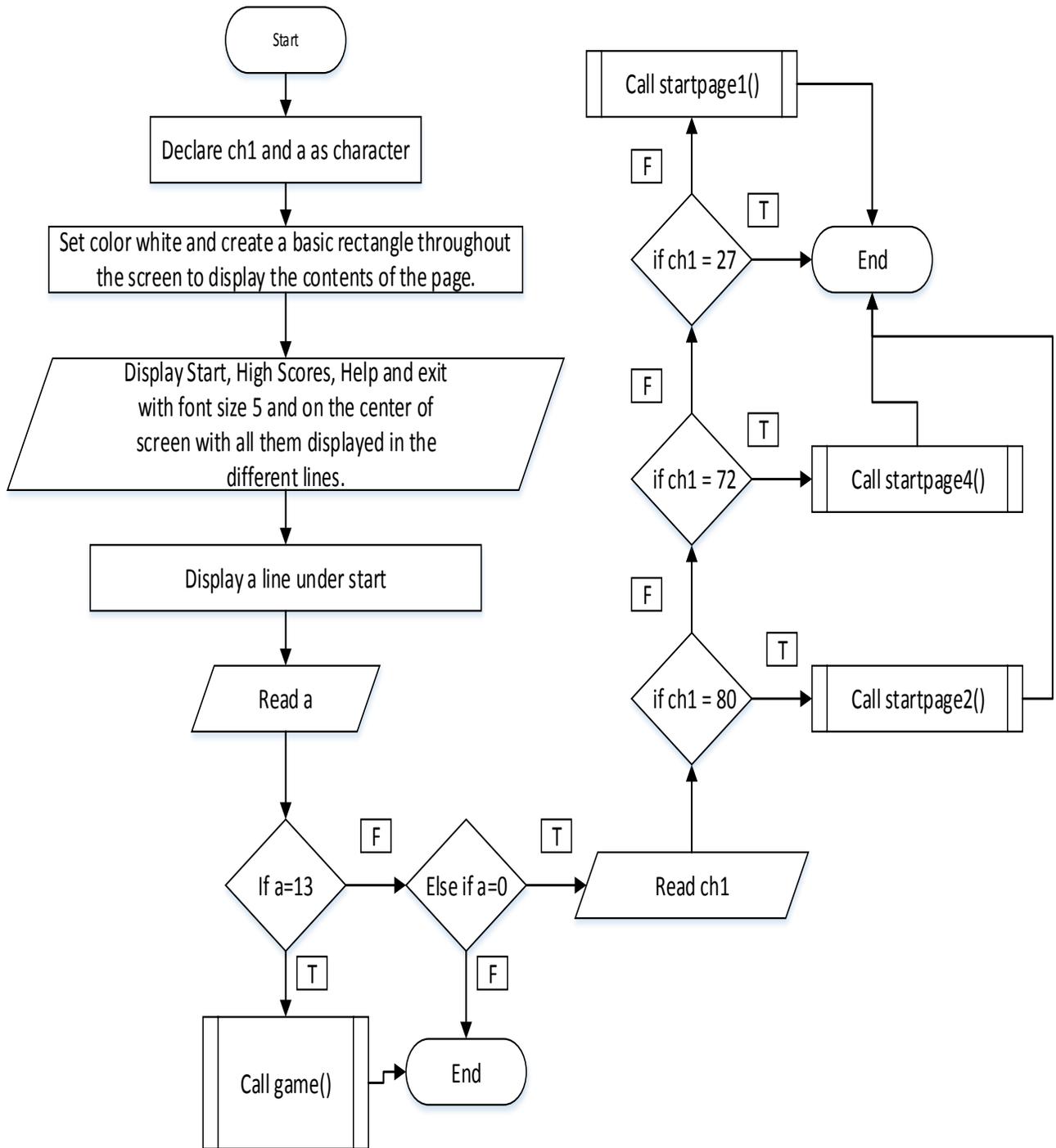
Flowchart for main()



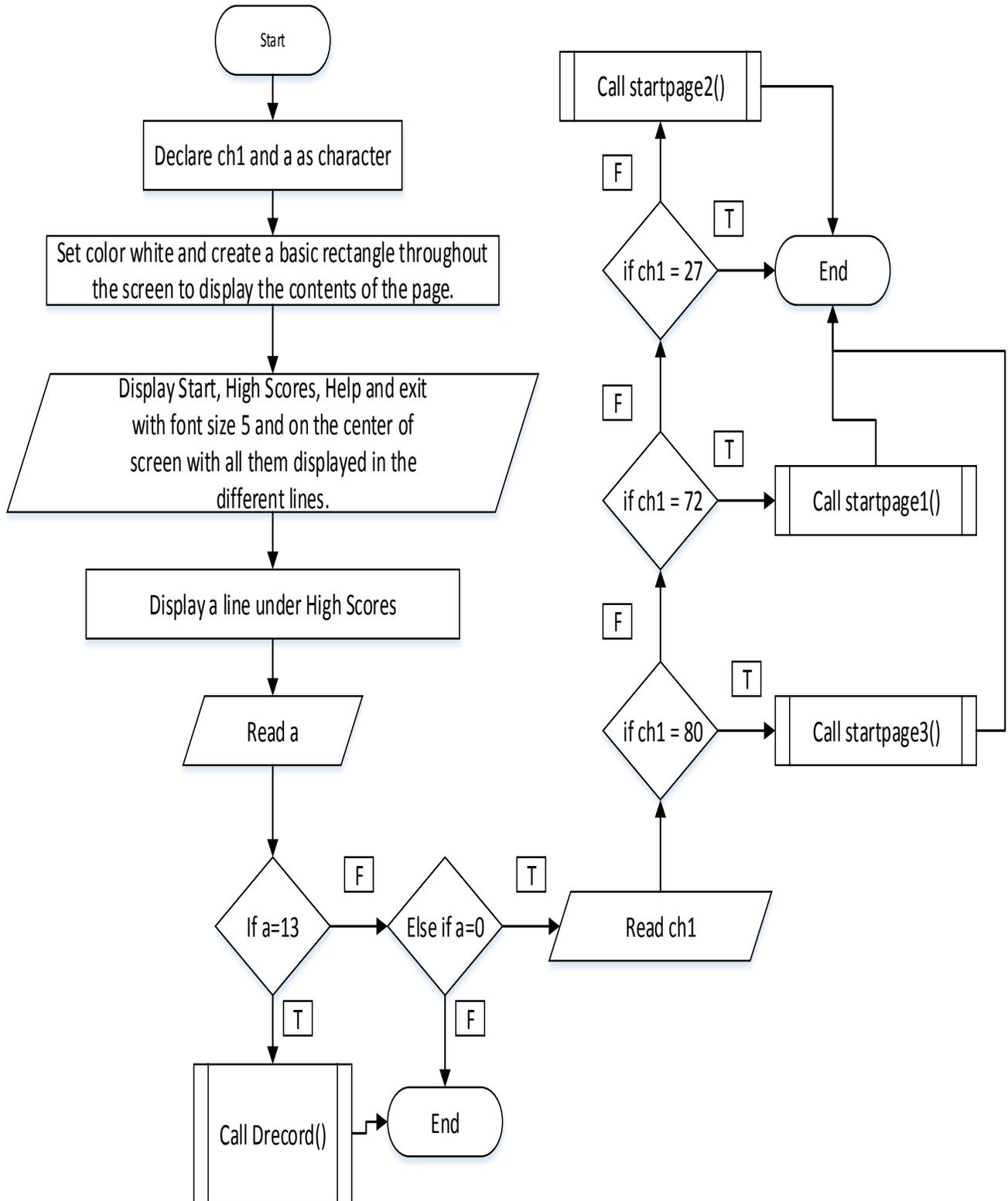
Flowchart for function game()



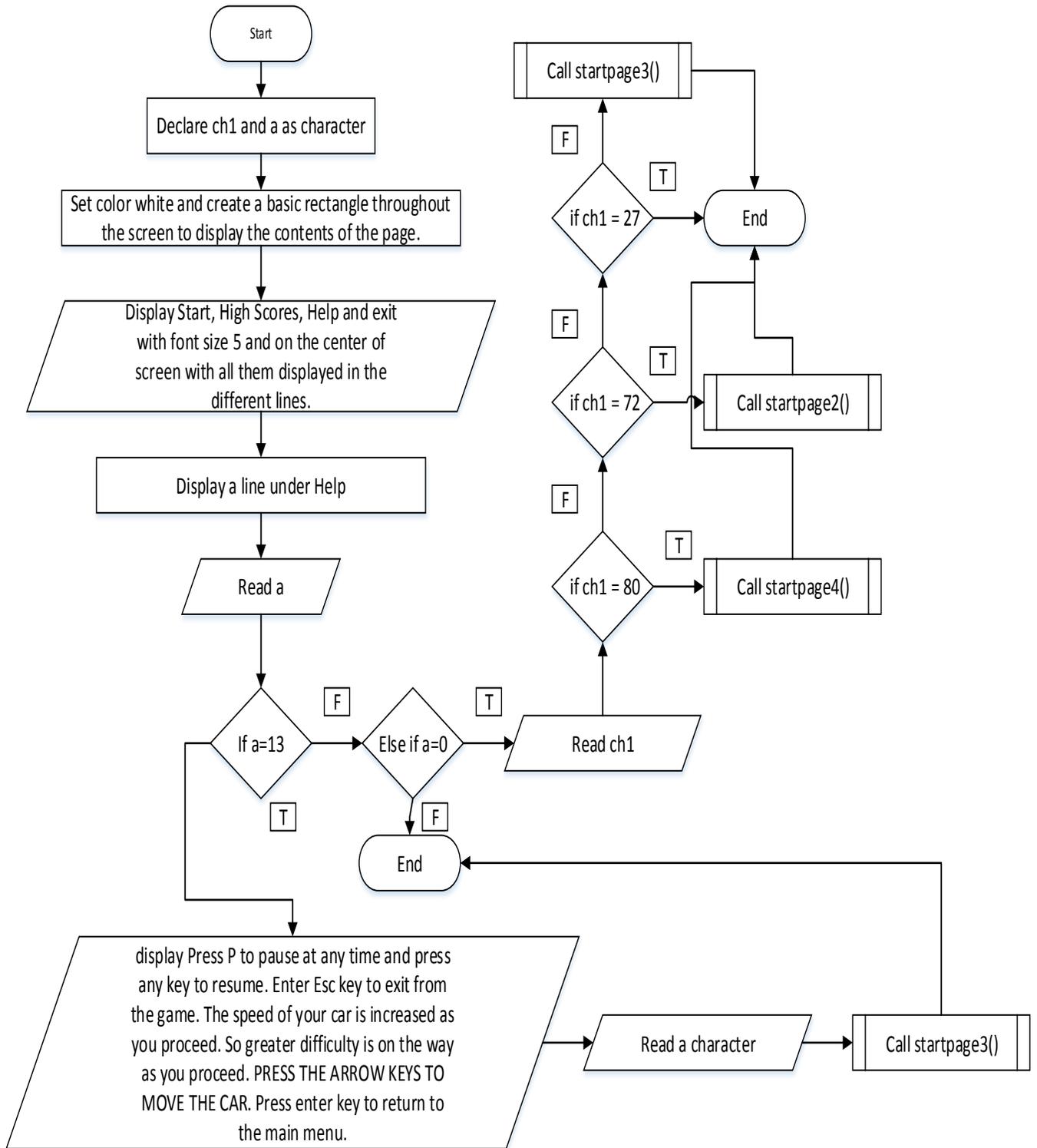
Flowchart for function startpage1()



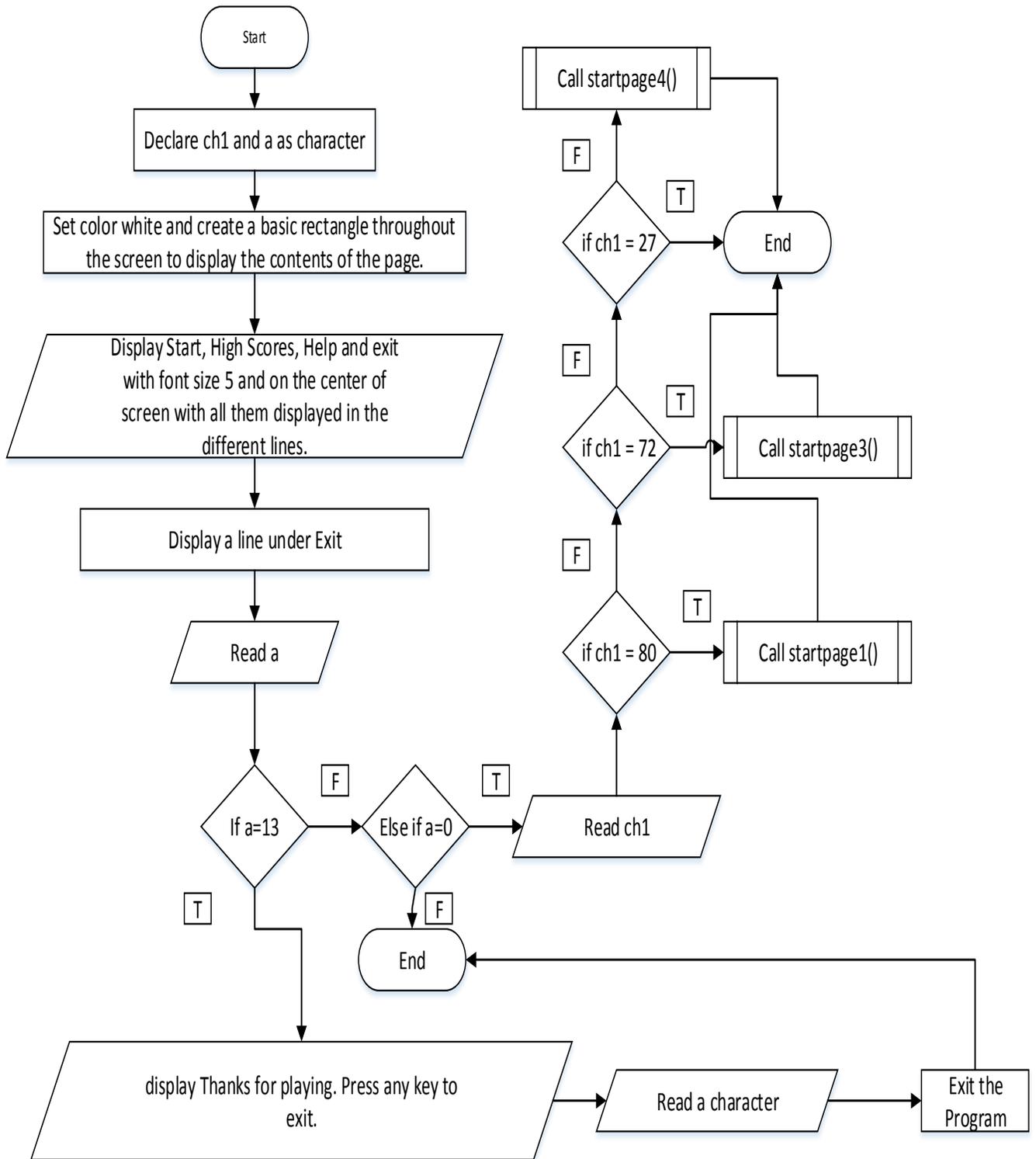
Flowchart for function startpage2()



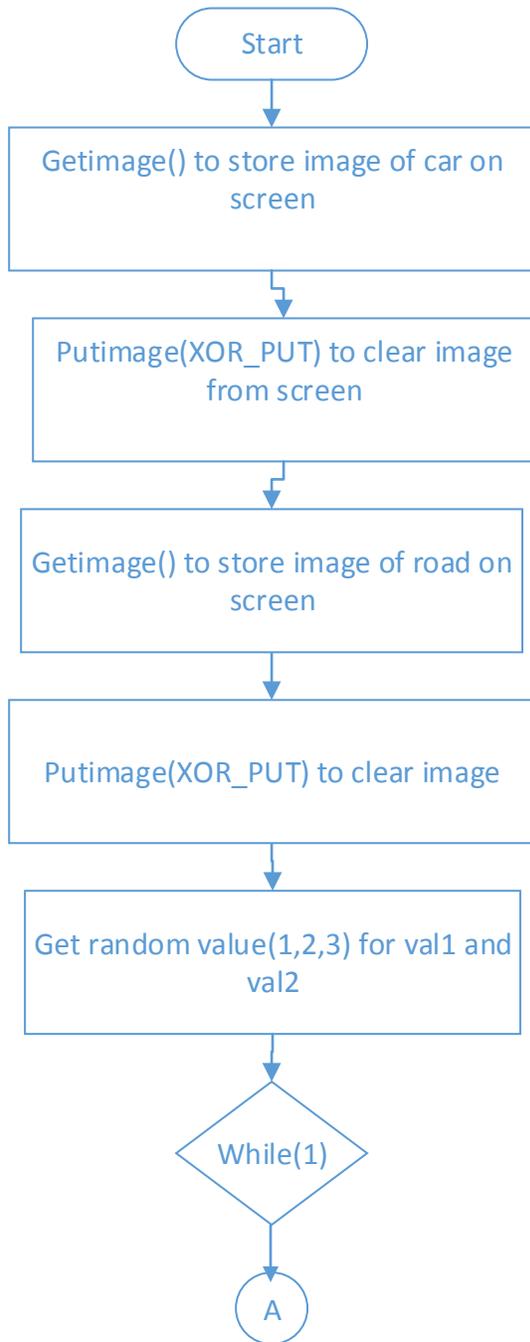
Flowchart for function startpage3()

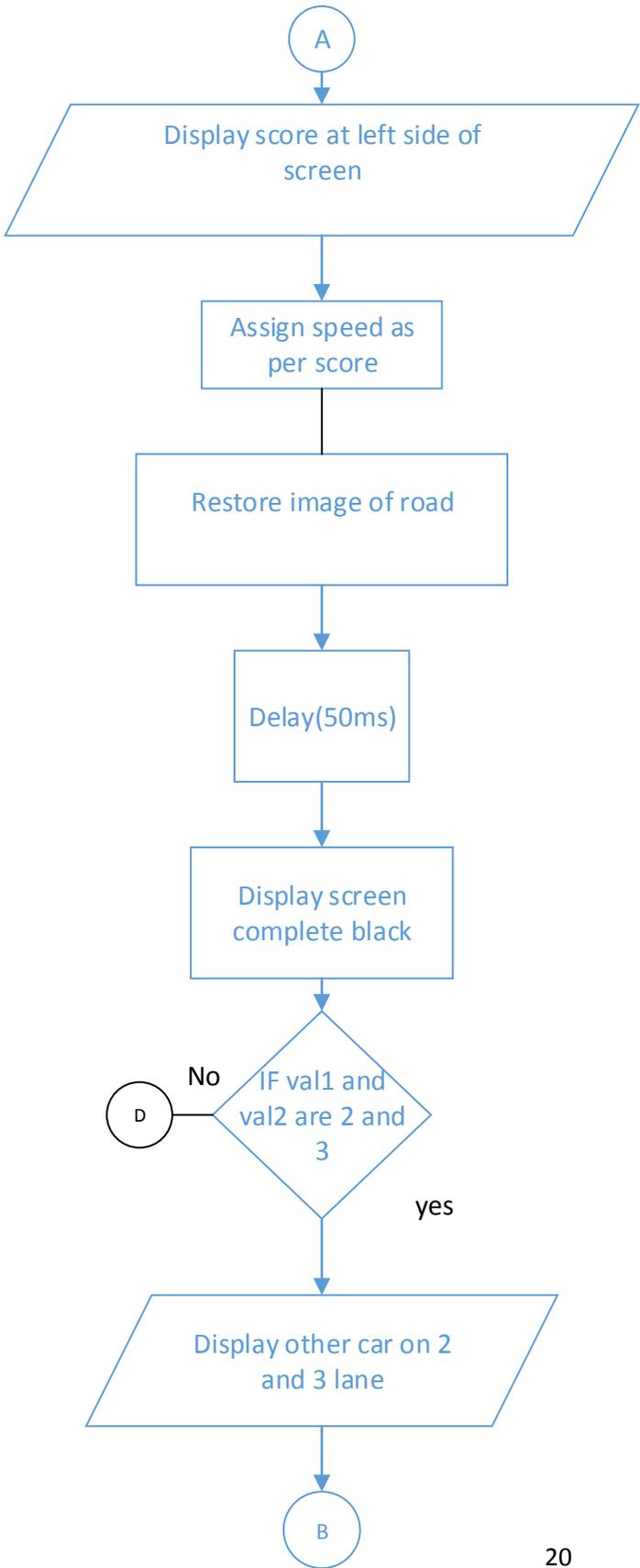


Flowchart for function startpage4()

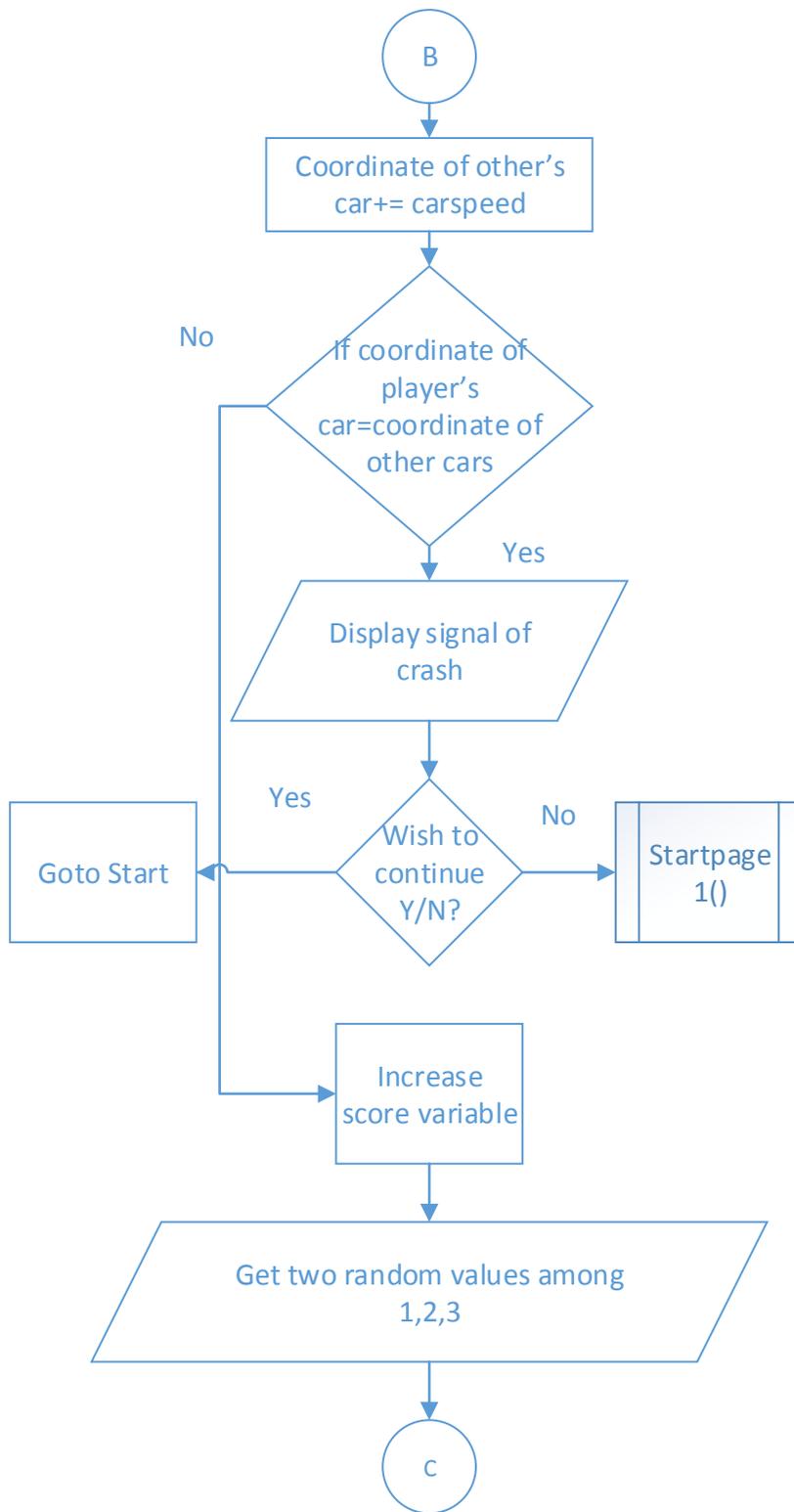


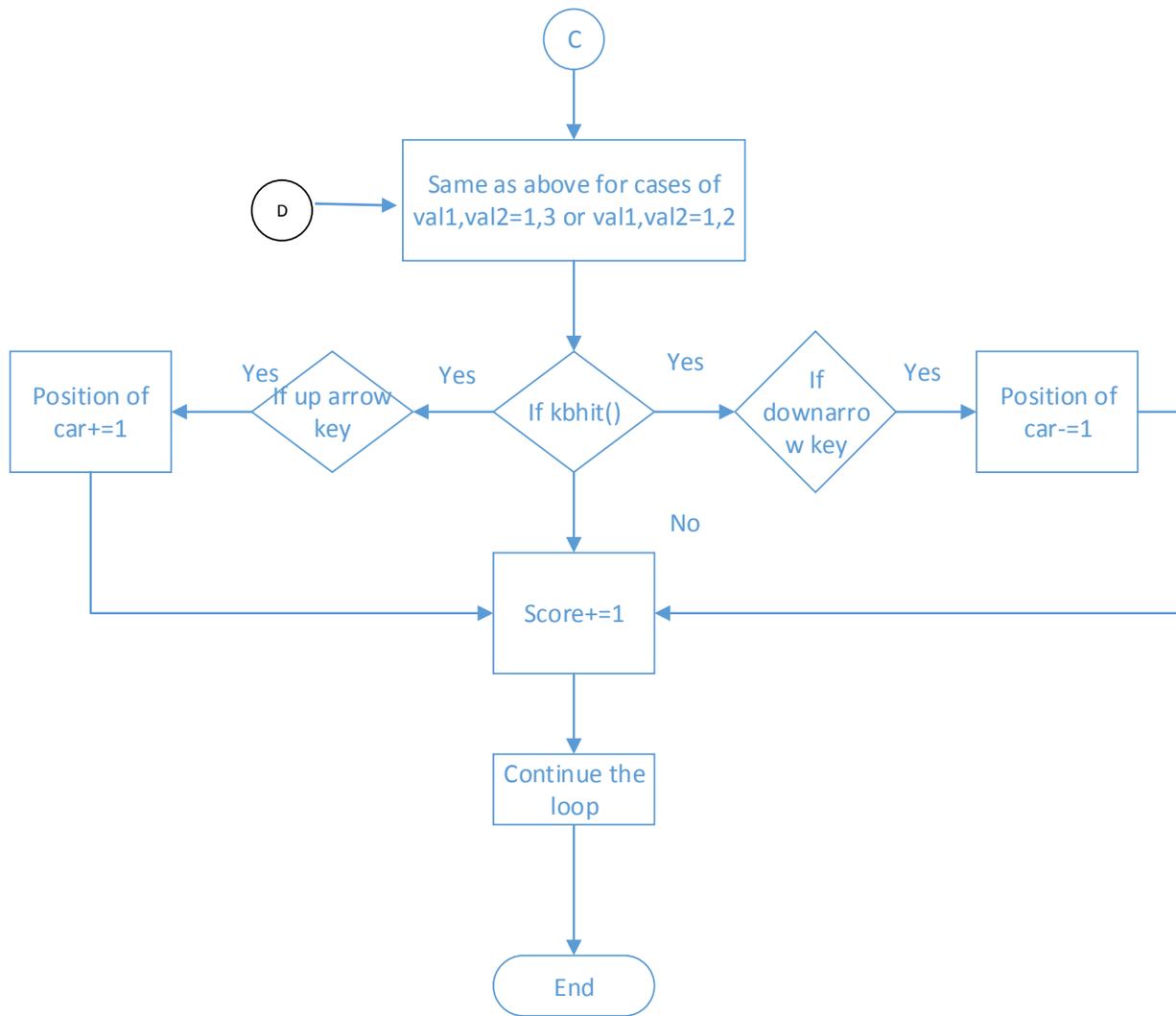
Flowchart for function animation()



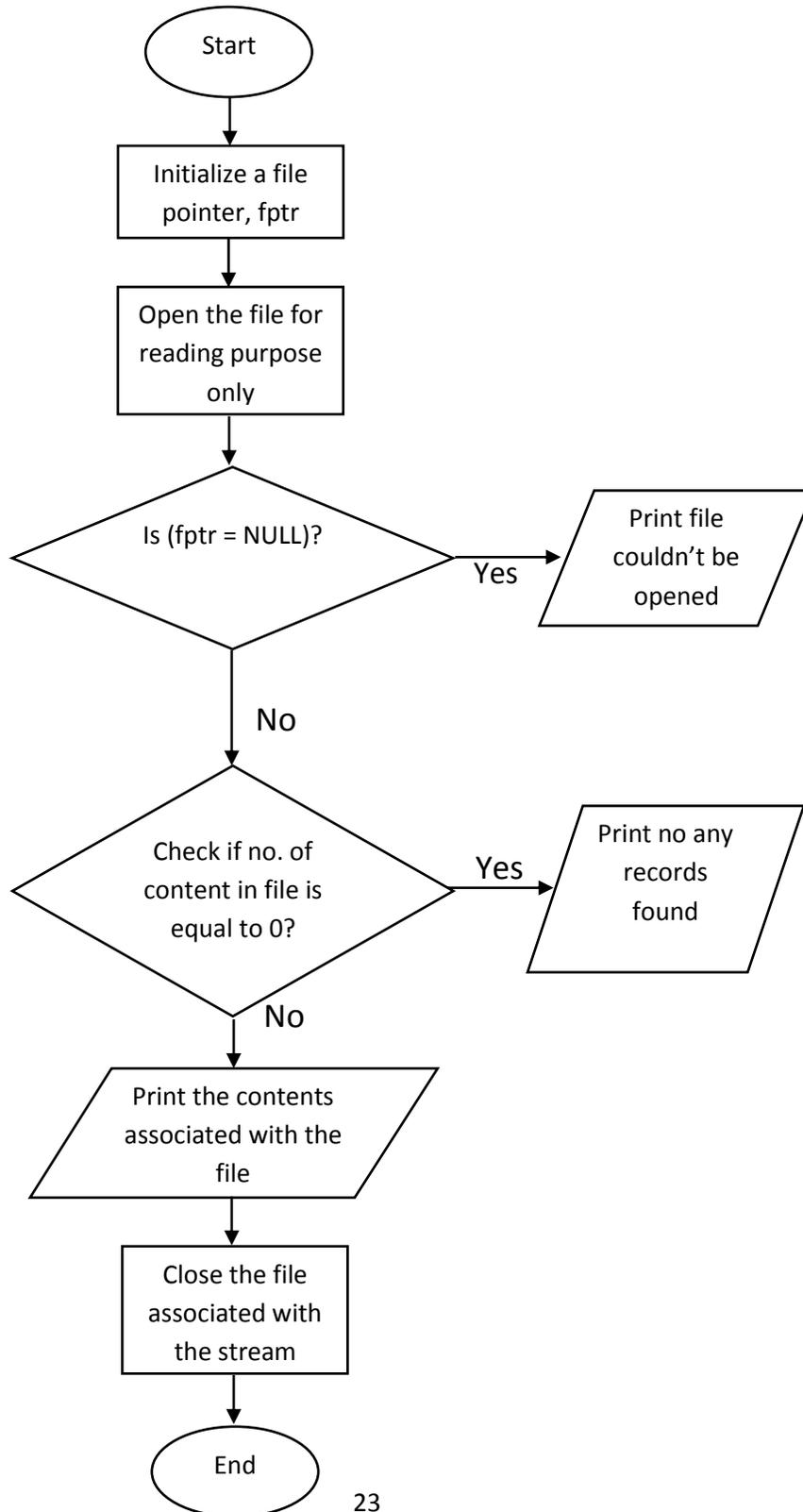


\

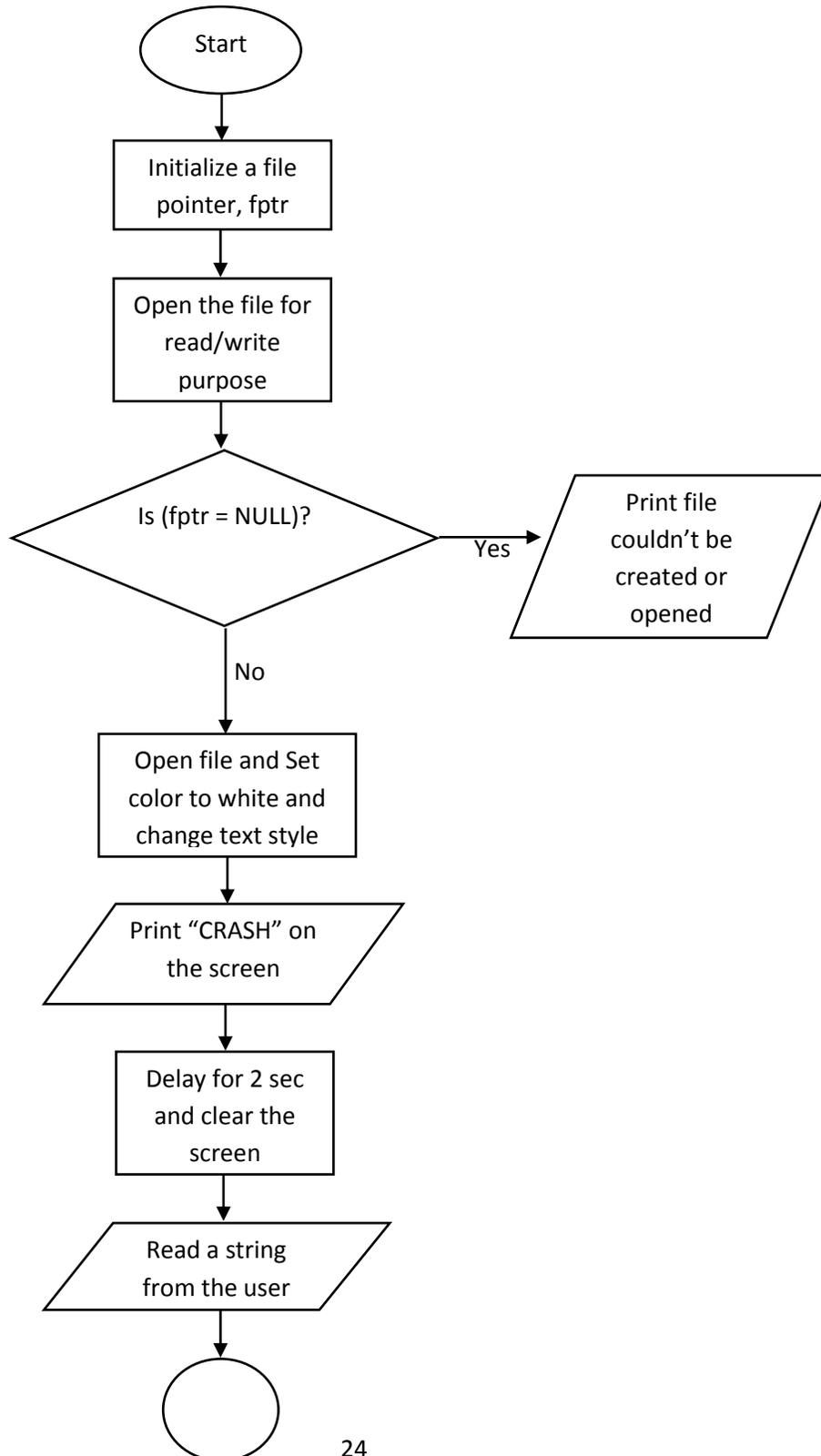


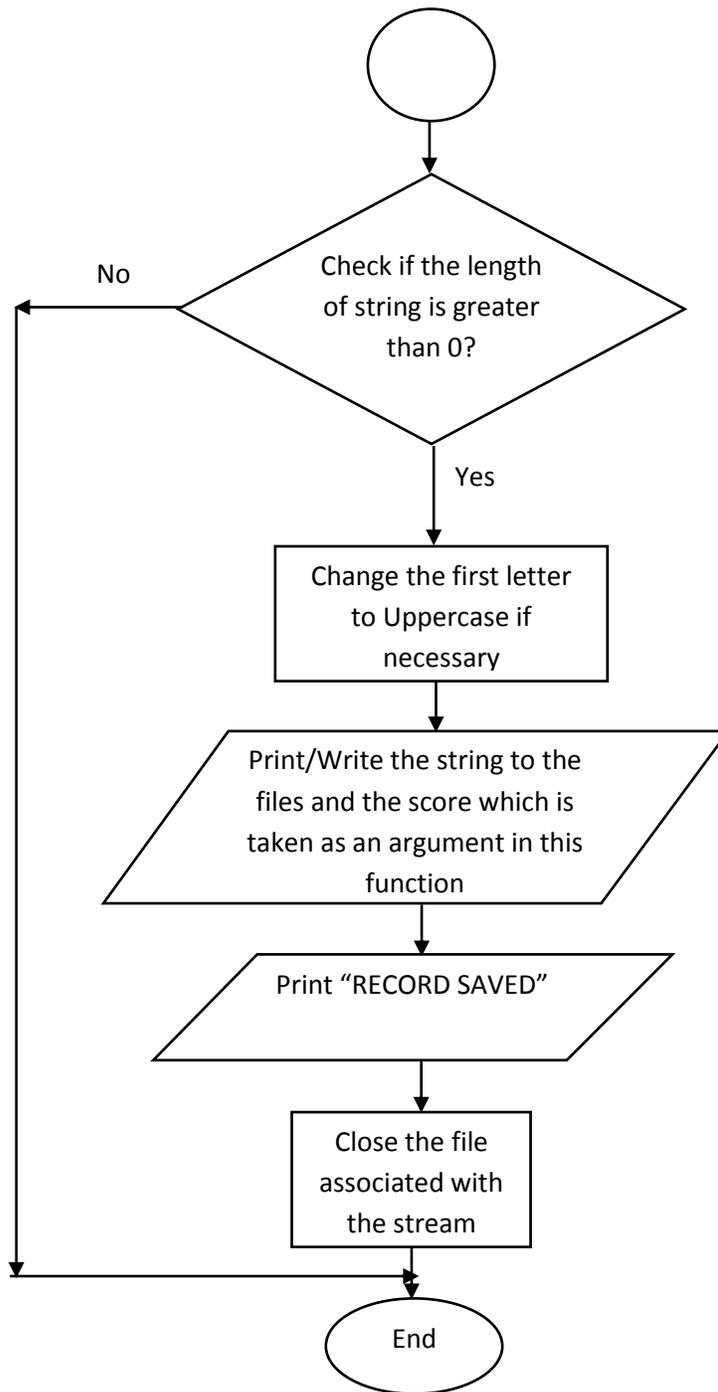


Flowchart for function Drecord()

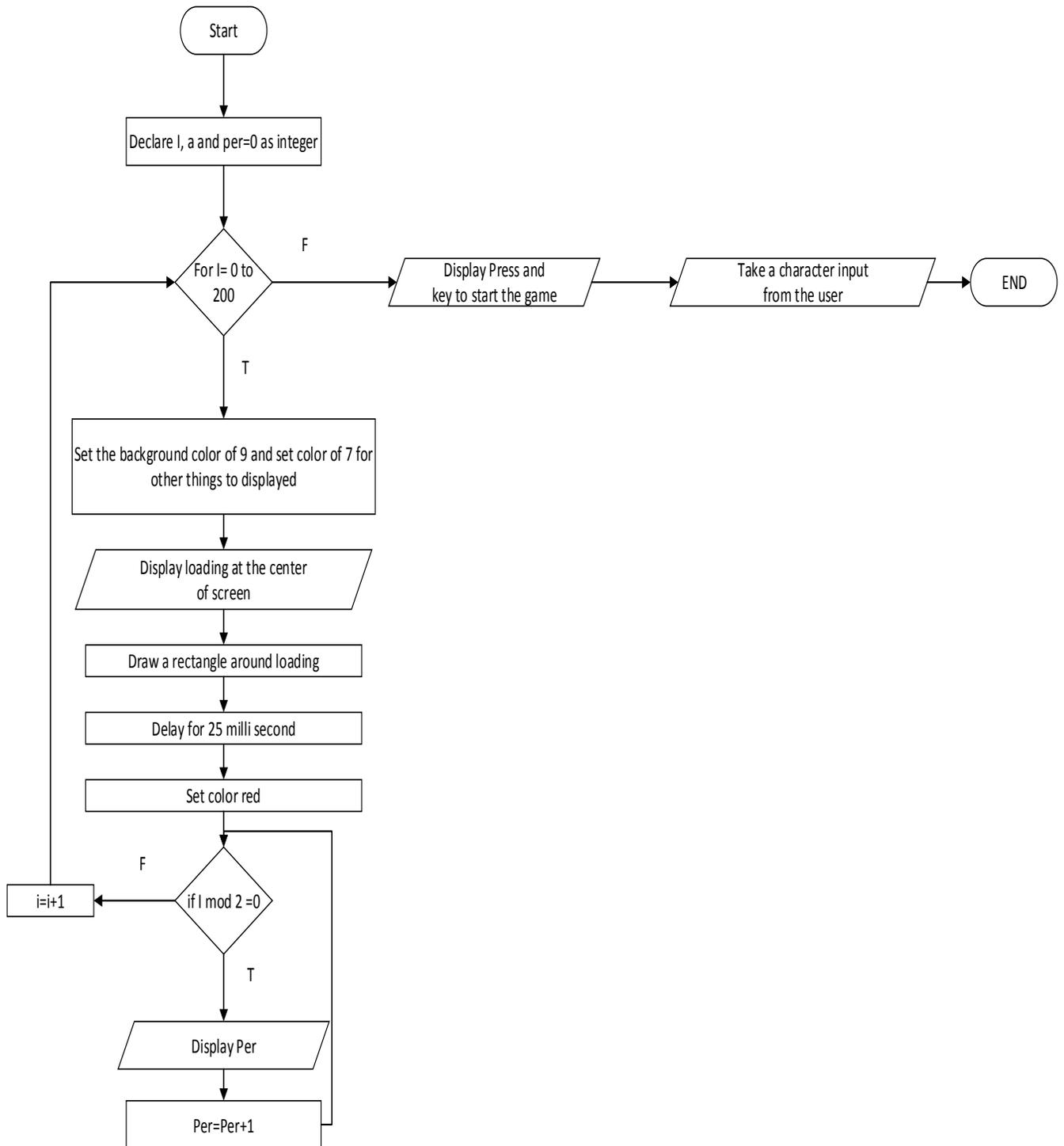


Flowchart for function Record(int)

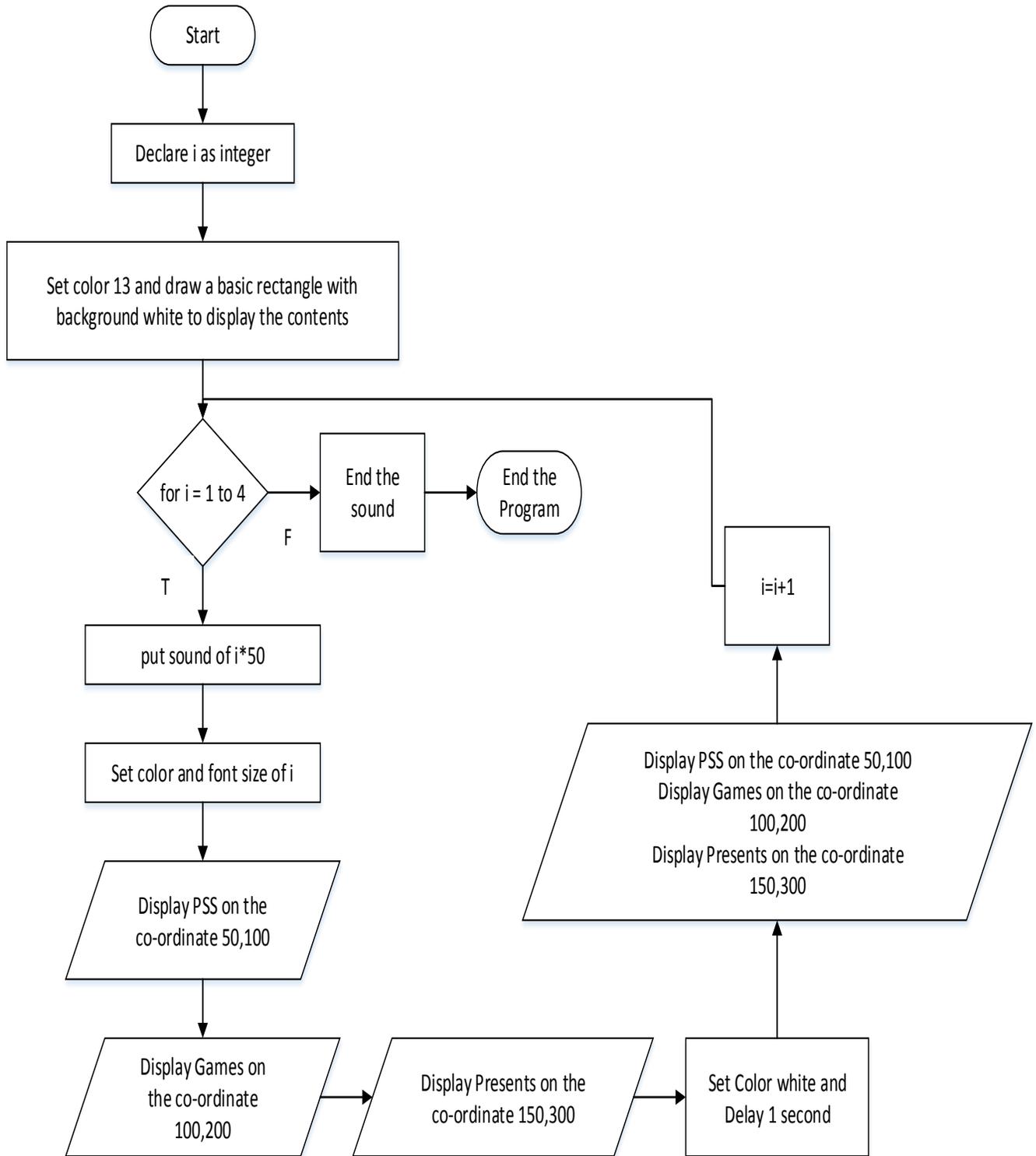




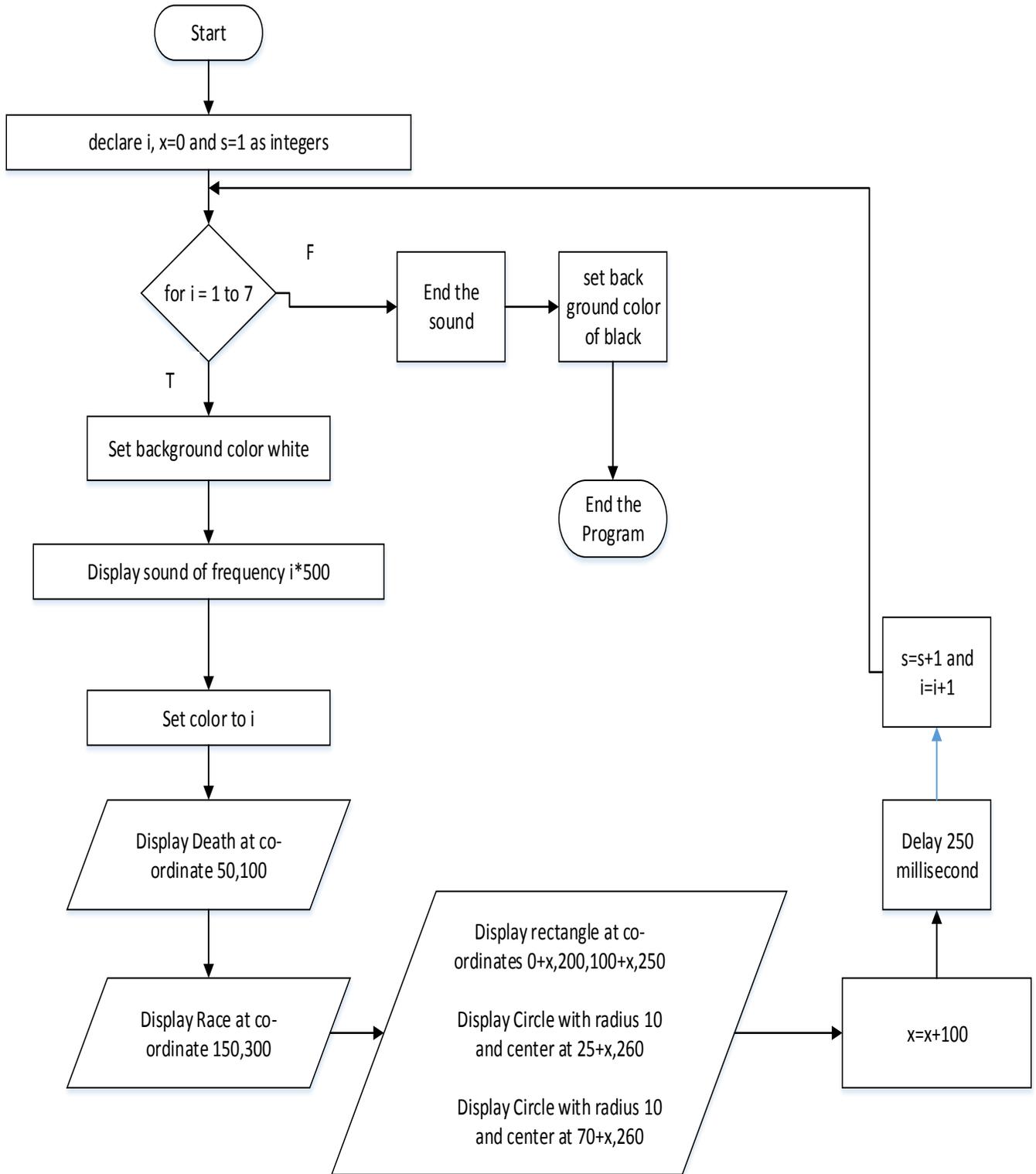
Flowchart for function loading()



Flowchart for function soundd()



Flowchart for function display()



6. Codes

Codes of the file STH.h

```
#ifndef STH_H
#define STH_H
#endif

#include <stdio.h>
#include <conio.h>
#include <dos.h>
#include <alloc.h>
#include <graphics.h>
#include <stdlib.h>
#include <time.h>
#include <ctype.h>
#include <process.h>
#include <string.h>

#define maxx 640
#define maxy 480

void animation(void);
void loading(void);
void startpage1();
void startpage2();
void startpage3();
```

```

void startpage4();

void record(int);

void Drecord();

void soundd (void);

void display (void);

void game();

int car[]={165,375,238,375,238,450,165,450,165,375};

int wheel[]={155,385,165,385,165,400,155,400,155,385};

int rect[]={ 0,0,maxx,0,maxx,maxy,0,maxy,0,0}; //whole screen

int rect2[]={150,0,maxx-150,0,maxx-150,maxy,150,maxy,150,0}; //for road

int rect3[]={150+103,1,150+103+15,1,150+103+15,50,150+103,50,150+103,1}; //for strip on road

int      rect4[]={maxx-150-119,1,maxx-150-119+15,1,maxx-150-119+15,50,maxx-150-119,50,maxx-150-119,1}; //for strip on road

void loading(void)
{
    int i,a, per = 0;

    for(i=0;i<=200;i++)
    {
        setbkcolor(9);

        setcolor(7);

        outtextxy(300,244,"LOADING...");

        setcolor(8);

        line(224+i,238,224+i,256);

        setcolor(8);
    }
}

```

```

    rectangle(224,238,425,256);

    delay(25);

    gotoxy(55,16);

    setcolor(RED);

    if(i % 2 == 0)
    {
printf("%d%",per);
per++;
    }
    }

    settextstyle(4, 0, 2);

    setcolor(WHITE);

    outtextxy(getmaxx()/2 - 140, getmaxy()/2 + 30, "Press any key to start the game!!!");

    getch();

    }

void startpage1(void)

{

char a, ch1;

int b;

setcolor(15);

line(215,300,275,300);

line(215,200,420,200);

setcolor(11);

rectangle(50,50,600,400);

```

```
setfillstyle(SOLID_FILL, GREEN);  
rectangle(40, 40, 610, 410);  
floodfill(45, 45, 11);  
setfillstyle(SOLID_FILL, 15);  
floodfill(55, 55, 11);  
setcolor(BLUE);  
settextstyle(6, DEFAULT_FONT, 5);  
outtextxy(215, 100, "Start");  
line(215, 150, 300, 150);  
setcolor(RED);  
outtextxy(215, 150, "High Scores");  
outtextxy(215, 200, "Help");  
outtextxy(215, 250, "Exit");  
a = getch();  
if(a == 13)  
    game();  
else if(a == 0)  
{  
    ch1 = getch();  
    if(ch1 == 80)  
        startpage2();  
    else if(ch1 == 72)  
        startpage4();  
    else if(ch1 == 27) exit(1);  
}  
else
```

```
startpage1();  
  
}  
  
void startpage2()  
{  
char a, ch1;  
setcolor(15);  
line(215,150,300,150);  
line(215,250,280,250);  
setcolor(11);  
rectangle(50,50,600,400);  
setfillstyle(SOLID_FILL, GREEN);  
rectangle(40,40,610,410);  
floodfill(45,45,11);  
setfillstyle(SOLID_FILL, 15);  
floodfill(55,55,11);  
setcolor(RED);  
settextstyle(6, DEFAULT_FONT, 5);  
outtextxy(215,100, "Start");  
setcolor(BLUE);  
line(215,200,420,200);  
outtextxy(215,150, "High Scores");  
setcolor(RED);  
outtextxy(215,200, "Help");  
outtextxy(215,250, "Exit");  
a = getch();
```

```
if( a == 13) {  
    cleardevice();  
    setcolor(11);  
    rectangle(50,50,600,400);  
    setfillstyle(SOLID_FILL, GREEN);  
    rectangle(40,40,610,410);  
    floodfill(45,45,11);  
    Drecord();  
    getch();  
    cleardevice();  
    startpage2();  
}  
else if(a == 0)  
{  
    ch1 = getch();  
    if (ch1 == 80)  
        startpage3();  
    else if (ch1 == 72)  
        startpage1();  
}  
else  
    startpage2();  
}  
  
void startpage3()  
{
```

```
char a, ch1;

setcolor(15);

line(215,200,420,200);

line(215,300,275,300);

setcolor(11);

rectangle(50,50,600,400);

setfillstyle(SOLID_FILL, GREEN);

rectangle(40,40,610,410);

floodfill(45,45,11);

setfillstyle(SOLID_FILL, 15);

floodfill(55,55,11);

setcolor(RED);

settextstyle(6, DEFAULT_FONT, 5);

outtextxy(215,100, "Start");

outtextxy(215,150, "High Scores");

setcolor(BLUE);

line(215,250,280,250);

outtextxy(215,200, "Help");

setcolor(RED);

outtextxy(215,250, "Exit");

a = getch();

if(a == 13)

{

cleardevice();

setcolor(11);

rectangle(50,50,600,400);
```

```
setfillstyle(SOLID_FILL, GREEN);  
rectangle(40, 40, 610, 410);  
floodfill(45, 45, 11);  
setfillstyle(SOLID_FILL, 15);  
floodfill(55, 55, 11);  
settextstyle(TRIPLEX_FONT, HORIZ_DIR, 4);  
setcolor(RED);  
outtextxy(55, 80, "GAME TIPS");  
settextstyle(6, HORIZ_DIR, 1);  
setcolor(BLUE);  
gotoxy(5, 13); outtextxy(55, 150, "Press P to pause at any time and press any key to resume. You can");  
outtextxy(55, 170, "Enter Esc key to exit from the game.");  
gotoxy(5, 15); outtextxy(55, 190, "The speed of your car is increased as you proceed. So greater");  
outtextxy(55, 210, "difficulty is on the way as you proceed");  
outtextxy(55, 250, "PRESS THE ARROW KEYS TO MOVE THE CAR");  
settextstyle(TRIPLEX_FONT, HORIZ_DIR, 2);  
setcolor(0);  
outtextxy(55, 300, "Press enter key to return to the main menu");  
getch();  
cleardevice();  
startpage3();  
}  
else if(a == 0)  
{  
ch1 = getch();  
if (ch1 == 80)
```

```
startpage4();

else if (ch1 == 72)

startpage2();

}

else

startpage3();

}

void startpage4()

{

char a, ch1;

setcolor(15);

line(215,250,280,250);

line(215,150,300,150);

setcolor(11);

rectangle(50,50,600,400);

setfillstyle(SOLID_FILL, GREEN);

rectangle(40,40,610,410);

floodfill(45,45,11);

setfillstyle(SOLID_FILL, 15);

floodfill(55,55,11);

setcolor(RED);

settextstyle(6, DEFAULT_FONT, 5);

outtextxy(215,100, "Start");

outtextxy(215,150, "High Scores");
```

```
outtextxy(215,200,"Help");

setcolor(BLUE);

line(215,300,275,300);

outtextxy(215,250,"Exit");

a = getch();

if( a == 13)

{

cleardevice();

setcolor(11);

rectangle(50,50,600,400);

setfillstyle(SOLID_FILL,GREEN);

rectangle(40,40,610,410);

floodfill(45,45,11);

setcolor(11);

setfillstyle(SOLID_FILL,15);

floodfill(55,55,11);

setcolor(BLUE);

settextstyle(3,0,4);

outtextxy(155,150,"THANKS FOR PLAYING");

settextstyle(6,0,5);

outtextxy(120,200,"PRESS ANY KEY TO EXIT");

getch();

exit(0);

}

if(a == 0)

{
```

```
ch1 = getch();  
if (ch1 == 80)  
startpage1();  
else if (ch1 == 72)  
startpage3();  
}  
else  
startpage4();  
}
```

```
void game(void)  
{  
int driver=DETECT,mode;  
initgraph(&driver,&mode,"C:\\tc\\bgi");  
loading();  
cleardevice();  
setbkcolor(BLACK);  
setcolor(WHITE);  
back:  
animation();  
getch();  
nosound();  
}
```

```

void animation(void)
{

unsigned char ch,ch1;

int carspeed=5; //speed of car coming from the top of the screen

int posn= 1; //player's car stored in first lane

int coord=1;

int x,i;

int val1,val2;

int score=0; //for programming

int displayscore=0;

void *buff,*buff2,*buff3;

void *buff4,*buff5,*buff6,*buff7; //for storing image to memory

setfillstyle(SOLID_FILL,BLUE);

fillpoly(5,car); //draw a sample of other car on road which is blue

setfillstyle(CLOSE_DOT_FILL,BROWN);

fillpoly(5,wheel); //draw wheel for sample of other car on road

buff4=malloc(imagesize(155,385,165,400)); //for storing the wheel car on road to the memory

getimage(155,385,165,400,buff4); //wheel of car stored in memory

putimage(238,385,buff4,COPY_PUT); //wheel of car printed in remaining three sides of blue car

putimage(155,425,buff4,COPY_PUT); //as above

putimage(238,425,buff4,COPY_PUT);//as above

buff5=malloc(imagesize(155,375,248,450)); //to store image of the blue car on memory

getimage(155,375,248,450,buff5);//image of blue car stored in memory

putimage(155,375,buff5,XOR_PUT); //the blue car deleted from screen

```

```

setfillstyle(SOLID_FILL,RED);

fillpoly(5,car);    //players car

setfillstyle(CLOSE_DOT_FILL,BROWN);

fillpoly(5,wheel);

putimage(238,385,buff4,COPY_PUT);

putimage(155,425,buff4,COPY_PUT);

putimage(238,425,buff4,COPY_PUT);

buff3=malloc(imagesize(155,375,248,450)); //image of red car stored on memory

getimage(155,375,248,450,buff3);

putimage(155,375,buff3,XOR_PUT); //delete red car on screen

srand(time(NULL));

x=1;

setfillstyle(SOLID_FILL,WHITE);

fillpoly(5,rect3);//draw strips on road

fillpoly(5,rect4);//as above

buff=malloc(imagesize(150+103,1,150+103+15,50)); //to store left strip on memory

getimage(150+103,1,150+103+15,50,buff);//stored left strip on memory

putimage(150+103,1,buff,XOR_PUT);//strips on road deleted from the memory

start:

setfillstyle(SOLID_FILL,GREEN);

fillpoly(5,rect); //draw green background

setfillstyle(SOLID_FILL,BLACK);

fillpoly(5,rect2);//draw black roads

```

```

    carspeed=5; //speed of car coming from the top of the screen

posn= 1; //player's car stored in first lane

coord=1;

score=0; //for programming

displayscore=0; //score that is displayed for user
again:
val1=rand()%3+1; //get random value among 1,2,and 3
val2=rand()%3+1; // another random value among 1,2,3
if (val1==val2) goto again; //values of val1 and val2 must be different, va11 and val2 are for positions of
other car on road

while(1)
{
{
settextstyle(TRIPLEX_FONT,HORIZ_DIR,4);
outtextxy(25,25,"SCORE"); //SCORE on left side of screen
}
gotoxy(6,5);printf("%d",displayscore); //score continously displayed

if (score<50) carspeed=30;
if (score>=50 && score<75) carspeed=35;
if (score>=75 && score<100) carspeed=40; //score and speedin the program code
if (score>=100 && score<125) carspeed=45;
if (score>=125) carspeed=48;

```

```

sound(100);

//below for printing moving strips on road
putimage(150+103,(x-1)*25,buff,COPY_PUT);
putimage(150+103+15+103,(x-1)*25,buff,COPY_PUT);
putimage(150+103,100+(x-1)*25,buff,COPY_PUT);
putimage(150+103+15+103,100+(x-1)*25,buff,COPY_PUT);
putimage(150+103,200+(x-1)*25,buff,COPY_PUT);
putimage(150+103+15+103,200+(x-1)*25,buff,COPY_PUT);
putimage(150+103,300+(x-1)*25,buff,COPY_PUT);
putimage(150+103+15+103,300+(x-1)*25,buff,COPY_PUT);
if (x!=5)
{putimage(150+103,400+(x-1)*25,buff,COPY_PUT);
  putimage(150+103+15+103,400+(x-1)*25,buff,COPY_PUT); }
if (x==5) {delay(50);
  putimage(150+103,1,buff,COPY_PUT);
  putimage(150+103+15+103,1,buff,COPY_PUT); }

delay(50); //for speed of moving road

//above for strips on road

setfillstyle(SOLID_FILL,BLACK);
fillpoly(5,rect2); //for removing strips and printing completely black

if (val1!=1 && val2!=1) //if random values are 2 and 3 that means other car in lane 2 and 3
{

```

```

coord=coord+carspeed;

if (coord>=300 && (posn==2 || posn==3)) //if players car coincides with other car
{
nosound();

sound(40);

delay(100);

nosound();

record(displayscore);

cleardevice();

setcolor(11);

rectangle(50,50,600,400);

setfillstyle(SOLID_FILL, GREEN);

rectangle(40,40,610,410);

floodfill(45,45,11);

setfillstyle(SOLID_FILL,15);

floodfill(55,55,11);

setcolor(9);

outtextxy(100,200,"Do you wish to play again: [Y/N]");

rectangle(400,200,440,225);

setfillstyle(SOLID_FILL, BLACK);

floodfill(405,205,9);

ch = getch();

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

goto start;

else if((ch == 'n') || (ch == 'N')) {

```

```

cleardevice();
startpage1();
}
}

putimage(273,coord,buff5,COPY_PUT); //display others car on second and third lane
putimage(391,coord,buff5,COPY_PUT);

if (coord>=450 )
{
score=score+5;
coord=1;
srand(time(NULL));
go1:
val1=rand()%3+1;
val2=rand()%3+1;
if (val1==val2) goto go1;
}
}

if (val1!=2 && val2!=2) //same as above, now for random value 1 and 3
{
coord=coord+carspeed;
if (coord>=300 && (posn==1 || posn==3))
{
nosound();
sound(40);
}
}

```

```
delay(100);  
nosound();  
record(displayscore);  
cleardevice();  
setcolor(11);  
rectangle(50,50,600,400);  
setfillstyle(SOLID_FILL, GREEN);  
rectangle(40,40,610,410);  
floodfill(45,45,11);  
setfillstyle(SOLID_FILL, 15);  
floodfill(55,55,11);  
setcolor(9);  
outtextxy(100,200, "Do you wish to play again: [Y/N]");  
rectangle(400,200,440,225);  
  
setfillstyle(SOLID_FILL, BLACK);  
floodfill(405,205,9);  
ch=getch();  
if ((ch=='y') || (ch=='Y'))  
goto start;  
else if((ch=='n') || (ch=='N')) {  
cleardevice();  
startpage1();  
}  
  
}  
  
putimage(155,coord,buff5,COPY_PUT);
```

```

putimage(391,coord,buff5,COPY_PUT); //display other's car on first and third lane

if (coord>=450 )
{
score=score+5;

coord=1;

srand(time(NULL));

go2:

val1=rand()%3+1;

val2=rand()%3+1;

if (val1==val2) goto go2;

}

}

if (val1!=3 && val2!=3) //same as above, now for random value 1 and 2

{

coord=coord+carspeed;

if (coord>=300 && (posn==1 || posn==2))

{

nosound();

sound(40);

delay(100);

nosound();

record(displayscore);

cleardevice();

setcolor(11);

rectangle(50,50,600,400);

setfillstyle(SOLID_FILL,GREEN);

```

```

rectangle(40,40,610,410);

floodfill(45,45,11);

setfillstyle(SOLID_FILL,15);

floodfill(55,55,11);

setcolor(9);

outtextxy(100,200,"Do you wish to play again: [Y/N]");

rectangle(400,200,440,225);

setfillstyle(SOLID_FILL,BLACK);

floodfill(405,205,9);

ch=getch();

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

goto start;

else if((ch == 'n' ) || (ch == 'N')) {

cleardevice();

startpage1();

}

}

putimage(155,coord,buff5,COPY_PUT); //display others car on first and second lane

putimage(273,coord,buff5,COPY_PUT);

if (coord>=450 )

{

score=score+5;

coord=1;

srand(time(NULL));

go3:

val1=rand()%3+1;

```

```

val2=rand()%3+1;

if (val1==val2)

goto go3;

}

}

if (posn==1) //for players'car posn = 1 means first lane

{

    putimage(155,375,buff3,COPY_PUT); //put image in first lane

}

if (posn==2) putimage(278,375,buff3,COPY_PUT); //like above

if (posn==3) putimage(391,375,buff3,COPY_PUT);

x++;

if (x==6) x=2; //this is related with displaying moving strips

if (kbhit())

{

ch=getch();

if (ch==27) //Esc key

{

nosound();

gotoxy(30,12); printf(" Sure to exit the game?" ) ;

gotoxy(30,13);printf(" If yes, press Y");

switch(getch())

{

```

```
case 'y':
case 'Y':
exit(0);
}
}
if (ch==112) //P to pause
{
nosound();
gotoxy(30,12);
printf("Press any key to continue");
getch();
sound(100);

}
if (ch==0)
{
ch1=getch();
switch(ch1)
{
case 75:
posn--; //changing positions according to left and right arrow
break;
case 77:
posn++;
break;
}
}
```

```

}

}

if (posn<1)

posn=1;

if (posn>3)

posn=3;

displayscore++; //Score is continously increased by value 1

nosound();

}

}

```

Codes of the file OPEN.c

```

#include "STH.H"

void Drecord()

{

char fname[50];

int score, numRecords = 0, repeat = 1, x_co = 10, y_co = 10;

FILE *fptr;

if ((fptr = fopen("record.txt", "r")) == NULL) {

printf("Error: Loading the file\n");

exit(1);

}

settextstyle(10,0,3);

outtextxy(70,50,"MOST WANTED RACERS");

setcolor(5);

```

```

line(51,100,520,100);

setcolor(7);

rectangle(50,110,600,120);

setfillstyle(5,7);

floodfill(51,111,7);

setlinestyle(0,1,3);

setcolor(RED);

line(51,180,200,180);

line(51,230,200,230);

line(51,280,200,280);

line(51,330,200,330);

line(52,380,200,380);

setlinestyle(0,1,1);

while(1) {

if(fscanf(fp, "%s %d", &fname, &score) == EOF)

break;

numRecords++;

}

if(numRecords == 0) {

setcolor(5);

settextstyle(3,0,4);

outtextxy(100,200,"Sorry, No records Available!!!");

}

rewind(fp);

```

```

while ((repeat <= numRecords) && (repeat <= 5)) {
fscanf(fp, "%s %d", &fname, &score);
gotoxy(x_co, y_co);
printf("Name:\t %s\n",fname);
y_co++;
gotoxy(x_co, y_co);
printf("Bounty: %d\n\n",score);
y_co += 2;
repeat++;
}
}

```

```

void record(int displayscore)
{
int loop;
char name[50], ch;
FILE *fp;
if ((fp = fopen("record.txt", "a+")) == NULL) {
printf("Error: Opening the file\n");
exit(1);
}
setcolor(WHITE);

settextstyle(3,0,10);

```

```
outtextxy(100,150,"CRASH!!!");  
  
delay(2000);  
  
cleardevice();  
  
setcolor(11);  
  
rectangle(50,50,600,400);  
  
setfillstyle(SOLID_FILL, GREEN);  
  
rectangle(40,40,610,410);  
  
floodfill(45,45,11);  
  
setfillstyle(SOLID_FILL, 15);  
  
floodfill(55,55,11);  
  
setcolor(5);  
  
settextstyle(3,0,1);  
  
outtextxy(220,160,"Enter Name:");  
  
setcolor(11);  
  
rectangle(220,188,395,213);  
  
rectangle(215,183,400,218);  
  
setfillstyle(SOLID_FILL, GREEN);  
  
floodfill(216,184,11);  
  
setfillstyle(SOLID_FILL, BLACK);  
  
floodfill(225,190,11);  
  
gotoxy(29,13);  
  
gets(name);  
  
name[0] = toupper(name[0]);  
  
for(loop = 0; loop < strlen(name); loop++) {  
  
if((name[loop]) == ' ' )  
  
break;
```

```
}  
name[loop + 1] = toupper(name[loop + 1]);  
if(strlen(name) > 0) {  
    fputs(name,fptr);  
    fputs("\n",fptr);  
    fprintf(fptr, "%d", displayscore);  
    fputs("\n",fptr);  
}  
fclose(fptr);  
  
cleardevice();  
setcolor(11);  
rectangle(50,50,600,400);  
setfillstyle(SOLID_FILL,GREEN);  
rectangle(40,40,610,410);  
floodfill(45,45,11);  
setfillstyle(SOLID_FILL,15);  
floodfill(55,55,11);  
setcolor(11);  
rectangle(215,200,400,230);  
setfillstyle(SOLID_FILL,BLACK);  
floodfill(225,215,11);  
gotoxy(29,14);  
printf("RECORD SAVED!!!");  
delay(1000);  
}
```

```

void soundd(void)
{
int i;
setcolor(13);
rectangle(0,0,getmaxx(),getmaxy());
setcolor(11);
rectangle(10,10,getmaxx() - 10, getmaxy() - 10);
setfillstyle(SOLID_FILL,15);
floodfill(getmaxx()/2, getmaxy()/2, 11);
for(i=1;i<5;i++)
{
sound (i*50);
setcolor(i);
settextstyle(DEFAULT_FONT, HORIZ_DIR,i);
outtextxy(50,100,"PSS");
outtextxy(100,200,"GAMES");
outtextxy(150,300,"PRESENTS");
setcolor(15);
delay(1000);
outtextxy(50,100,"PSS");
outtextxy(100,200,"GAMES");
outtextxy(150,300,"PRESENTS");
}
nosound();

```

```

}

void display(void)
{
int i,x=0,s=1;
for(i=1;i<8;i++)
{
setbkcolor(15);
setcolor(15);
cleardevice();
sound(i*500);
setcolor(i);
settextstyle(DEFAULT_FONT, HORIZ_DIR,i);
outtextxy(50,100,"DEATH");
outtextxy(150,300,"RACE");
rectangle(0+x,200,100+x,250);
circle(25+x,260,10);
circle(70+x,260,10);
x=x+100;
delay(250);
s++;
}
nosound();
cleardevice();
setbkcolor(0);
}

```

```
void main (void)
{
int driver=DETECT,mode;

initgraph(&driver,&mode,"C:\\tc\\bgi");

soundd();

delay(100);

display();

startpage1();

closegraph();

}
```

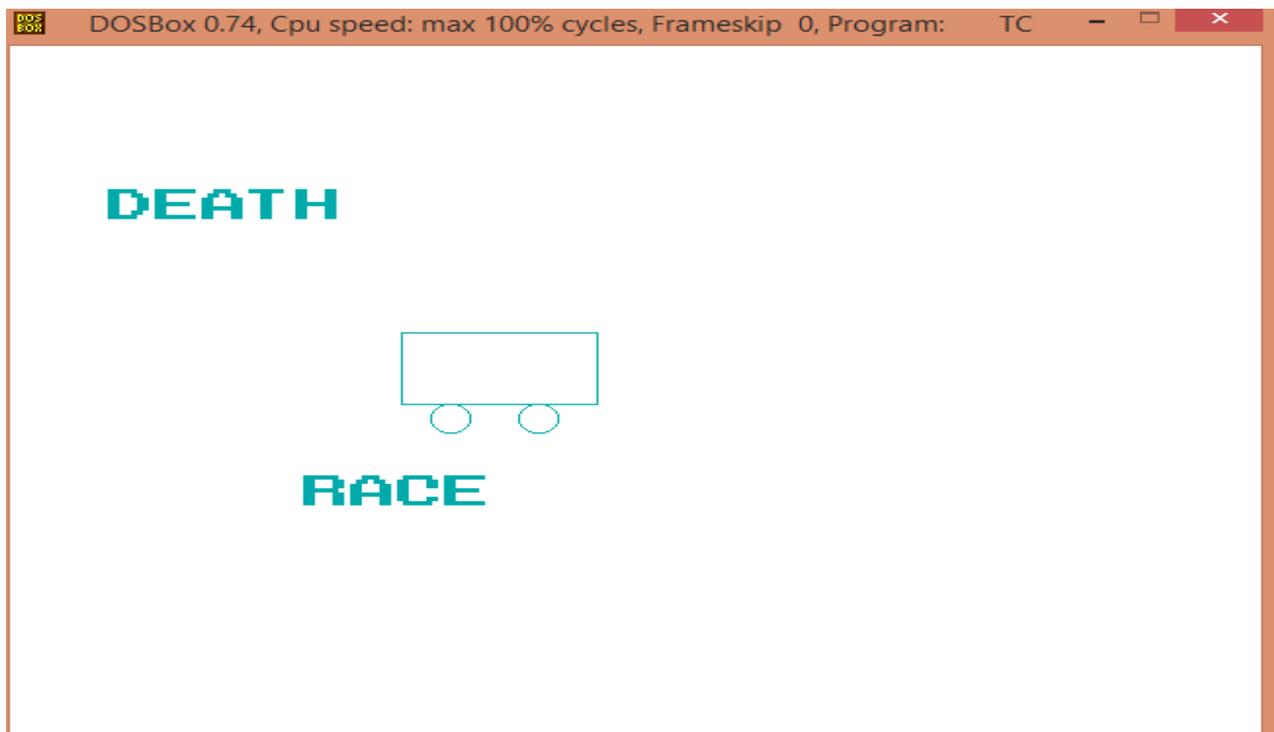
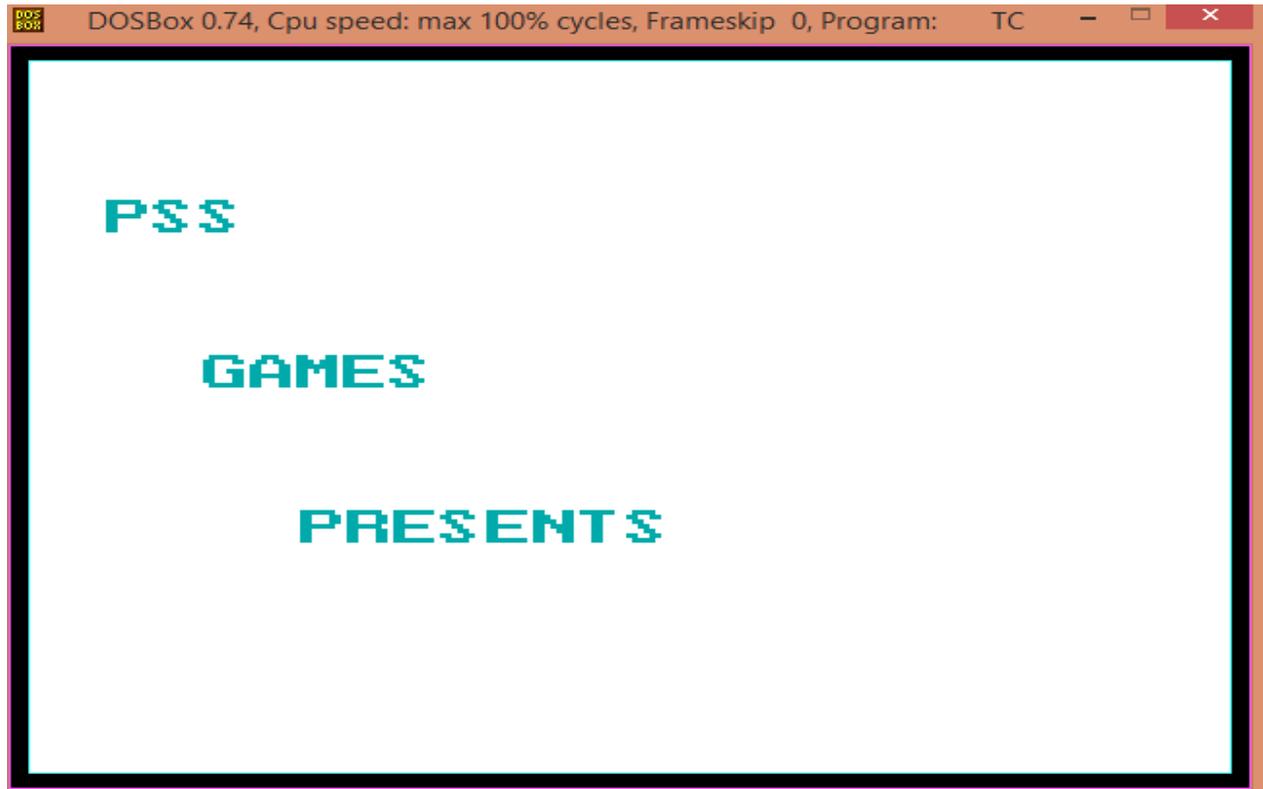
7. Compilation, Testing and Debugging

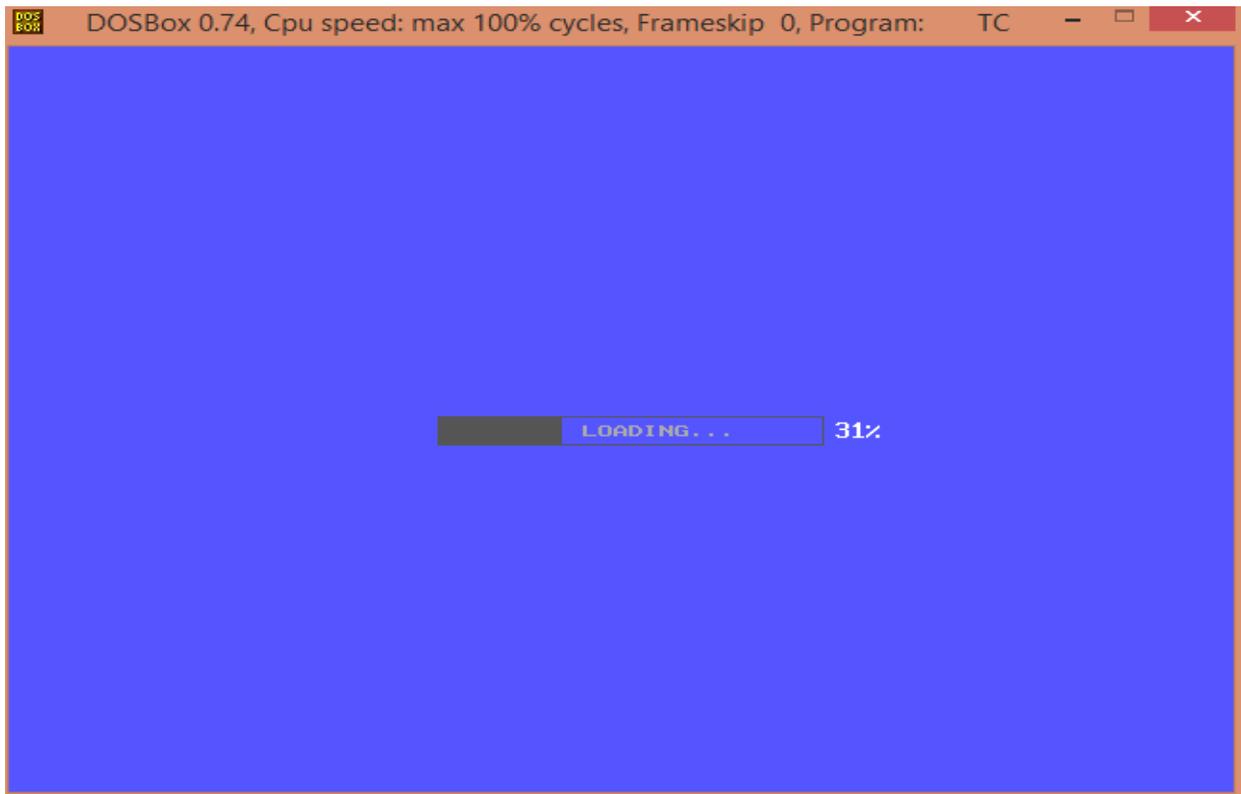
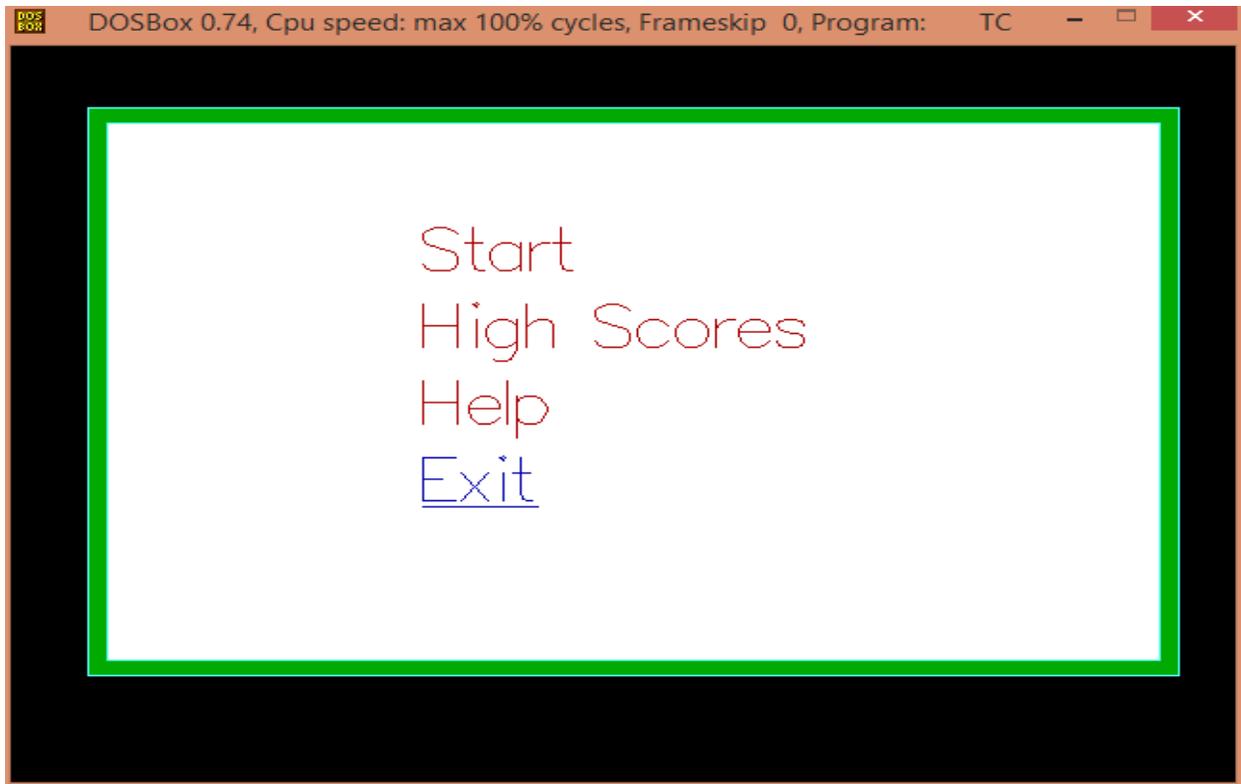
The program was compiled regularly after each function was created so that it would be easy to debug it if there were any bugs. Any errors or bugs that came along the process were debugged by the help of teachers and friends.

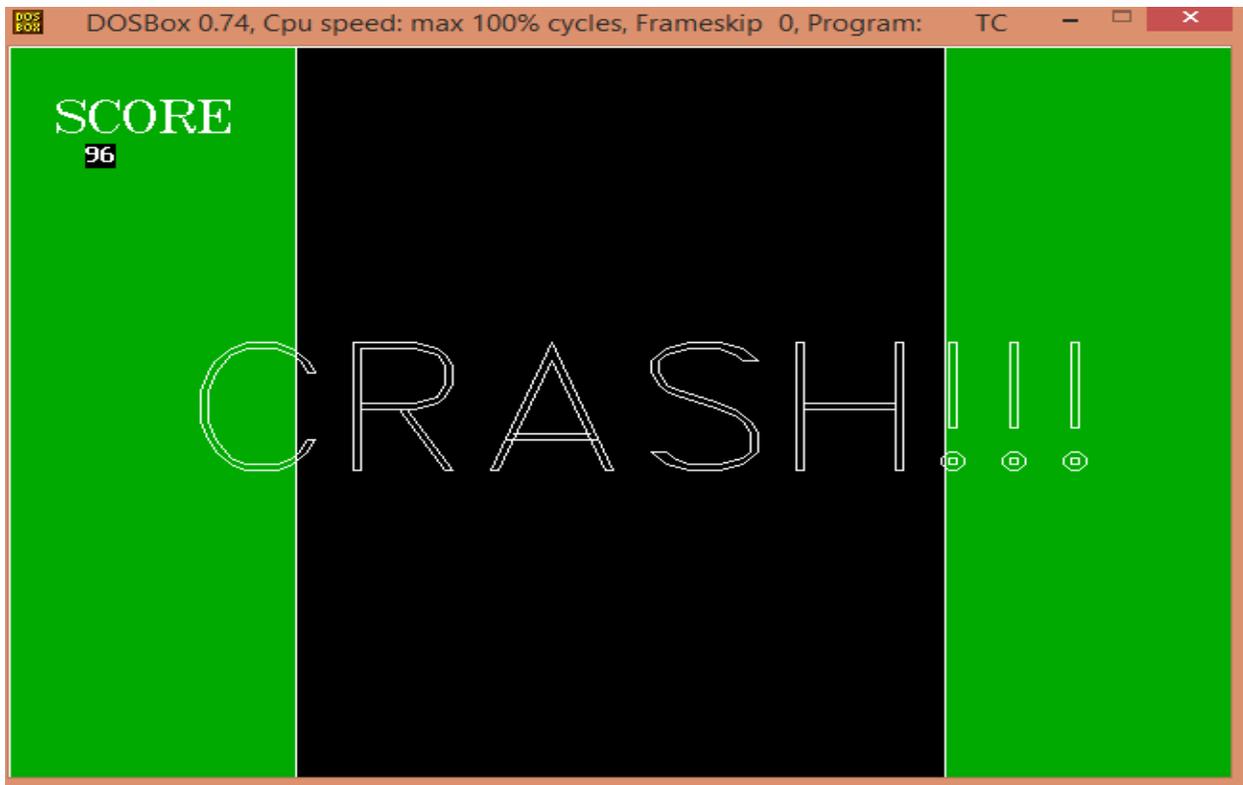
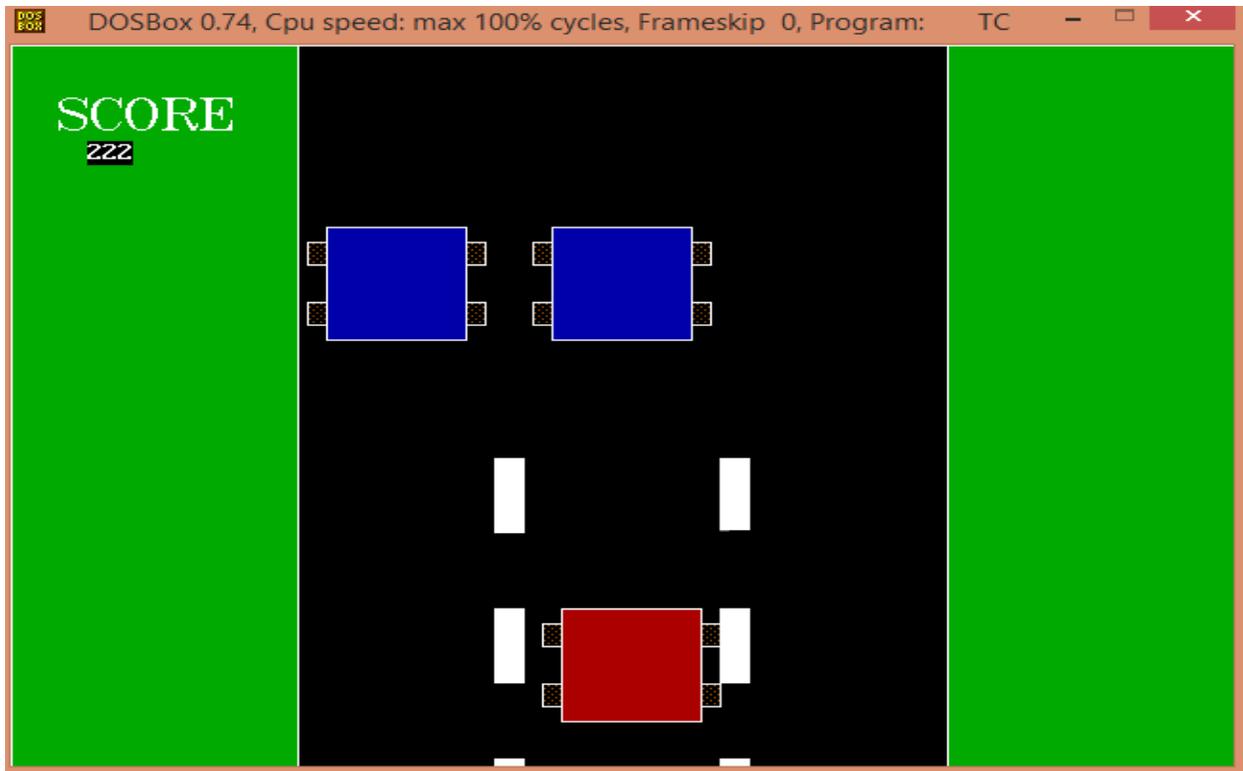
For the testing of the project, we requested our friend to try our game on their computers and according to their suggestions and comments we made certain changes to make it more user friendly and user continent.

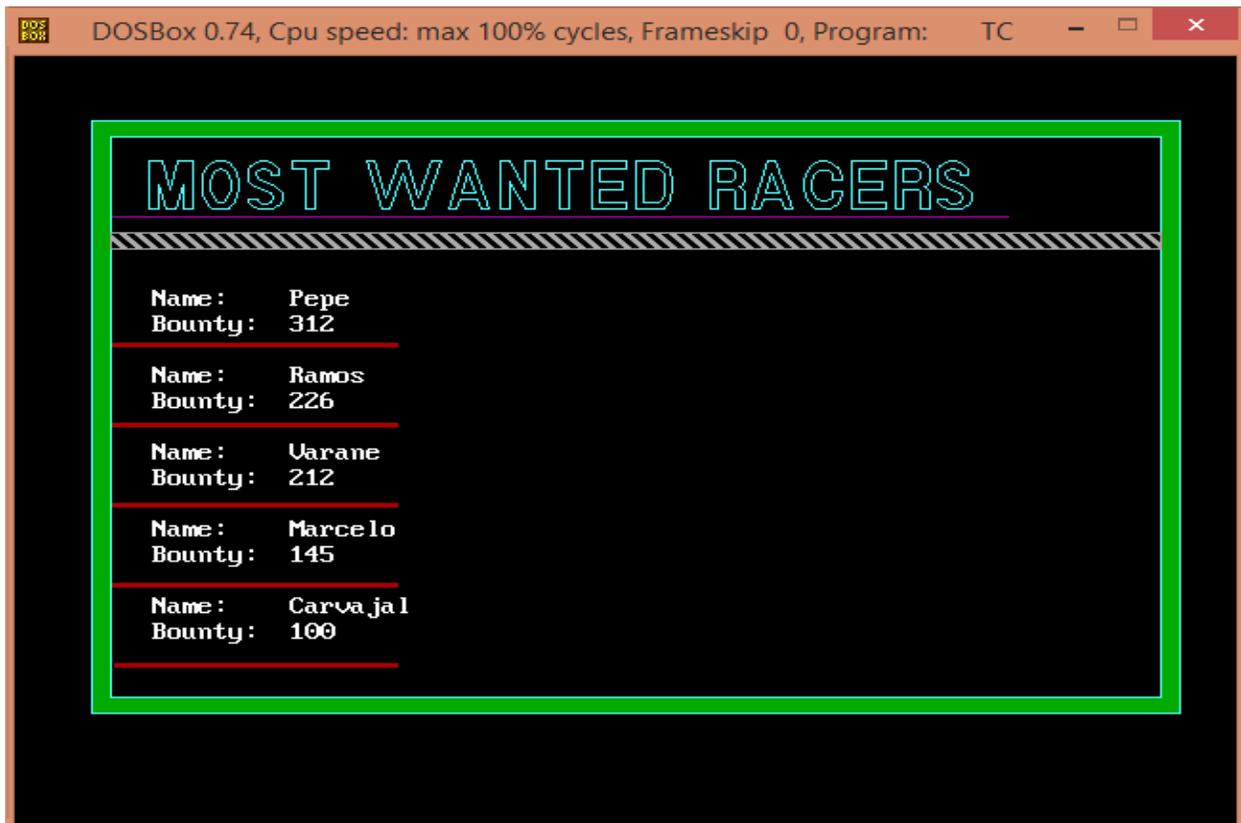
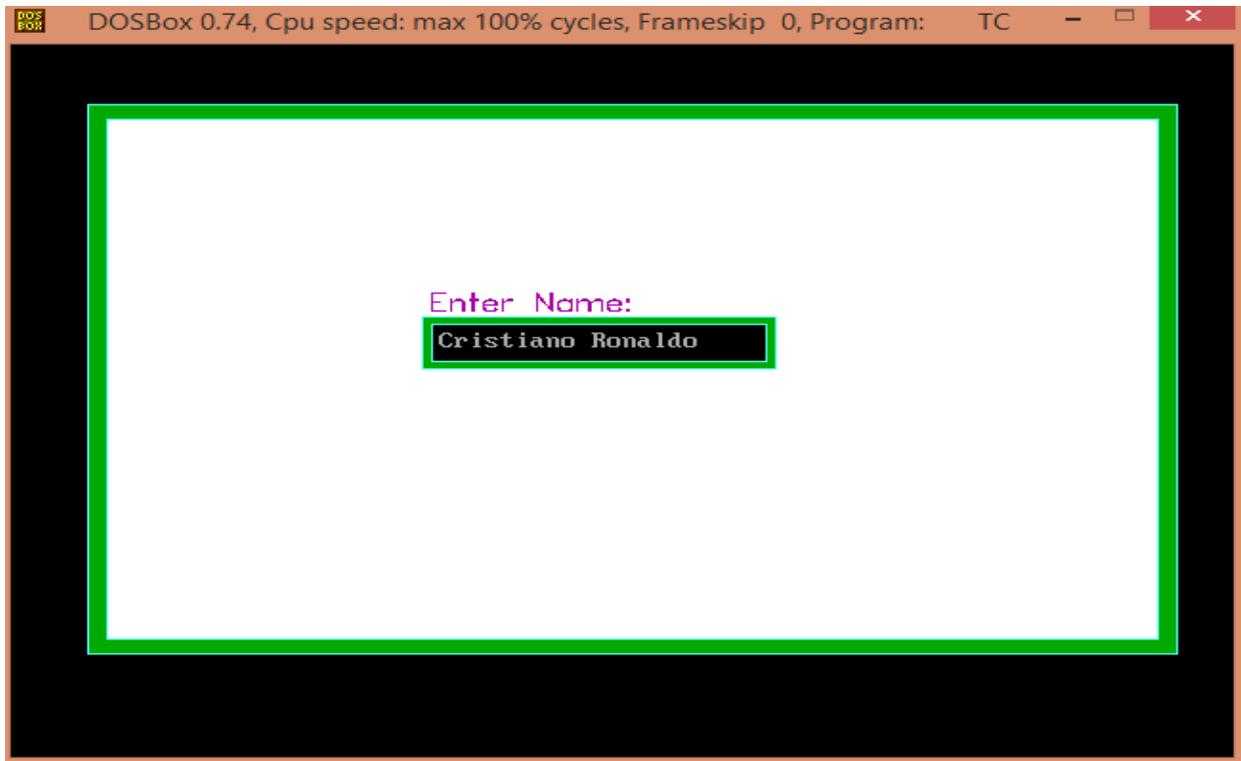
In this way, the project was compiled, tested and debugged to make it error free and user friendly.

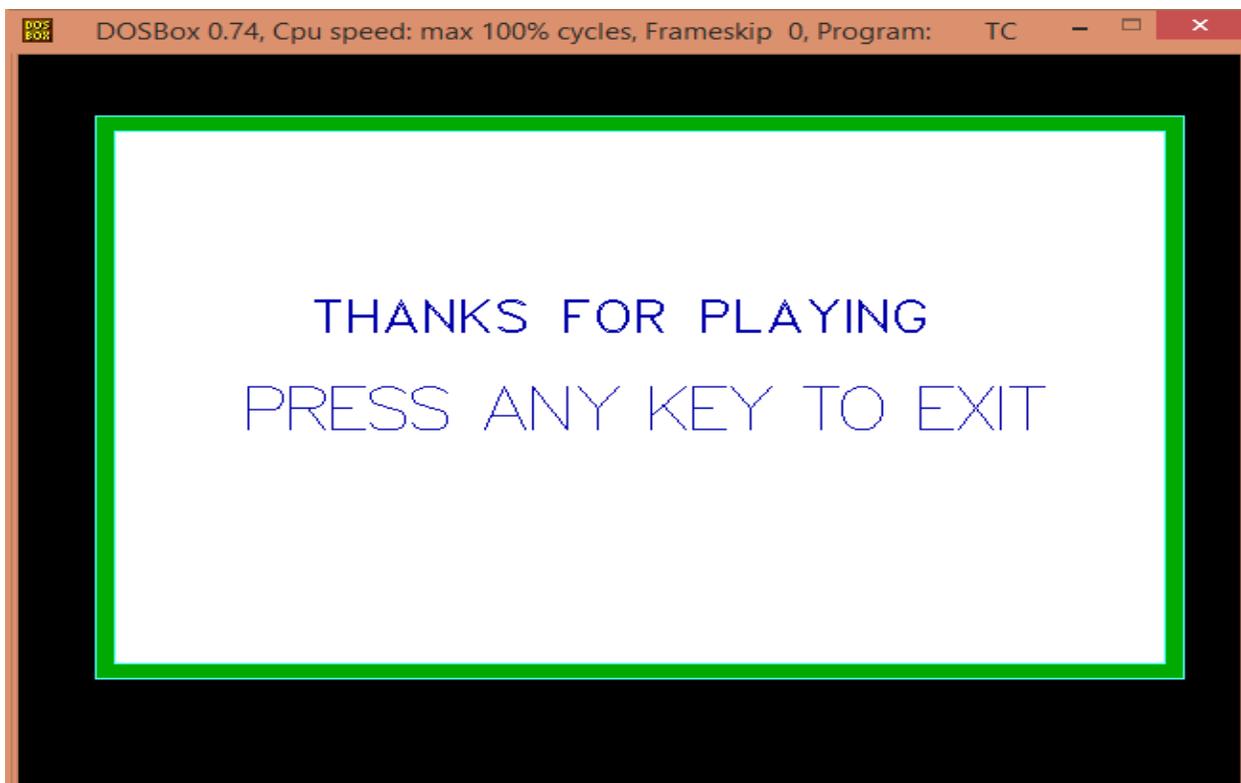
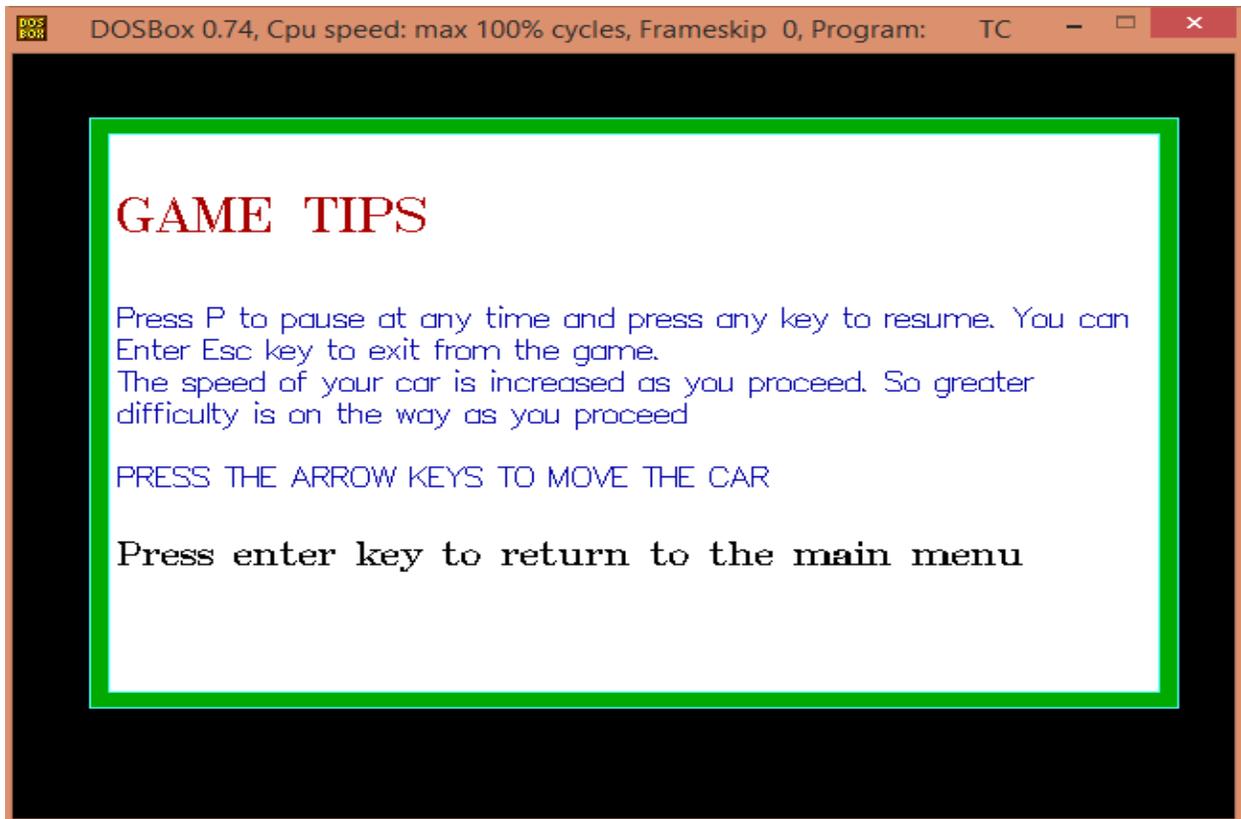
8. Output Snapshots











9. Discussion and Conclusions

This mini project of computer entitled "DEATH RACE" is a simple concept of racing game. The player has to avoid obstacles on the way and as the game proceeds the speed of the car is increased continuously. The game is simple, yet is fun to play. This follows the concept of modular programming, all three of our team members have created separate functions, and when all combined together has given rise to this form. Hence, this is truly a team work. We have used concept of file handling effectively to store the name and score of best players. We have tried to use all our skills in C programming we learned in this semester, as per our teacher's instructions. Yet we acknowledge that a lot can be done to improve this. The future developers can improve this game with creating the motion of car and speed with up arrow keys and other, and setting timers, to create better sensation of car racing game.

This project helped to enhance our skills in C and made us realize that learning to program a computer is real fun. We are again thankful to our teacher who assigned this project and helped us to build confidence in programming. Lastly, we emphasize that this is open source program and future developers can use this for reference and for developing better programs.