National University
of computer and emerging sciences

COAL LAB SEMESTER PROJECT

Course code: EL-229

BS (CS)-A

Batch: Fall 2018

Submitted by:

Hassan Shahzad 18I-0441

Ayesha Marriyam 18I-0457

Shoaib Murtaza 18I-0496

Submitted to:

Sir Zia Ur Rehman

Date of Submission:

01-06-2020

Project Report:

1. Graphics (Done by Ayesha, Shoaib & Hassan):

This part was done by all 3 group members and the work load was distributed equally. We started off by designing the interface of the game for which we had to do a lot of brainstorming. Then we started off by designing the interface of the game in which we changed the background of the gameplay along with dividing the screen into two parts (3:1). The right most part was left for the health, score etc. then we started drawing the robot (pixel by pixel). After that canons were drawn and placed in different locations on the very top of the screen whereas the robot was at the very bottom of it. To draw graphics, we used many procedures like Draw Box, Draw Square, Draw Background etc.

2. Robot Movement (Done by Hassan):

The next step was to move the robot. In this step we had to manually detect the keypress and then move the robot accordingly. So, for this purpose, we made an array containing all the pixels of robot. Then we made two variables which contained the x-coordinates and y coordinates namely robox, roboy respectively. We made a procedure in which first we checked the condition whether a key is pressed or no and if yes, then which key is pressed. if right key is pressed, then we increment the x-axis of the robot which allows it to move in right direction and similarly if a user presses left key, then we decrement x-axis of the robot which allows it to move in left direction. The speed of robot movement was adjusted by incrementing with a number other than 1 (i.e. if you want the robot to move faster, you'll increment/decrement it by 2 or 3 or 4 depending on how fast you want it to move.

3. Menu Display (Done by Hassan):

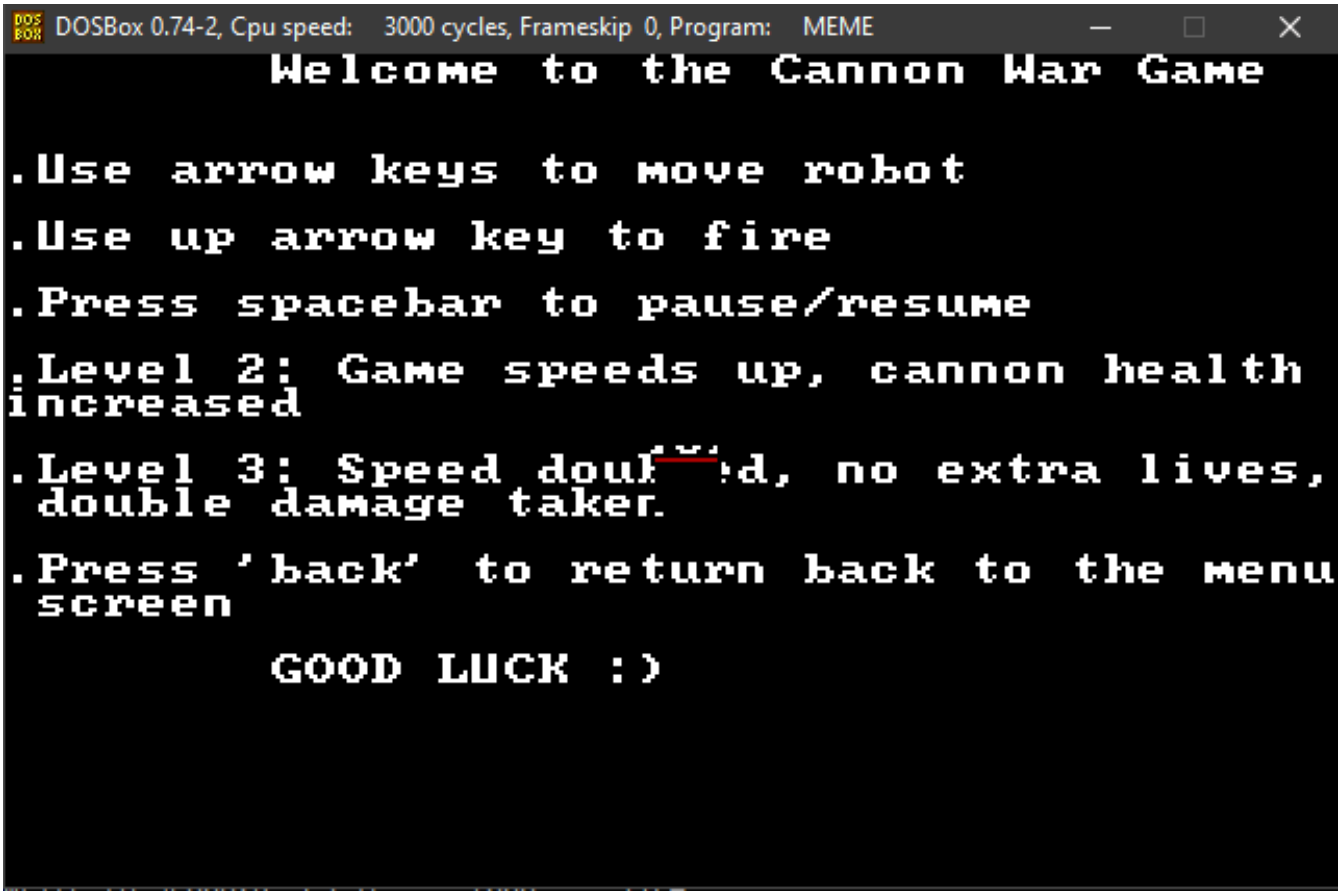
The next part was to display the menu. For this purpose, graphics were used to display the different options followed by proper coding to execute those options.
The options that we gave were

a. Play
b. Resume
c. Instructions
d. Score
e. Exit

When the player clicks “play”, he will be redirected to the gameplay. If he clicks “Resume”, he’ll be redirected to the game where he left. When a user will click on “Instructions”, he’ll be showed a screen containing all the rules and instructions regarding how to play the game. When the user will click on “Score”, he’ll be shown the high score along with the time and date of that score. This was again done by file handling as every time a score crosses a high score, his new score will be written to the high score file as the new high score which will be displayed when the user selects this option. Lastly, when the player selects “Exit”, he will be redirected tot the terminal and the game will quit.

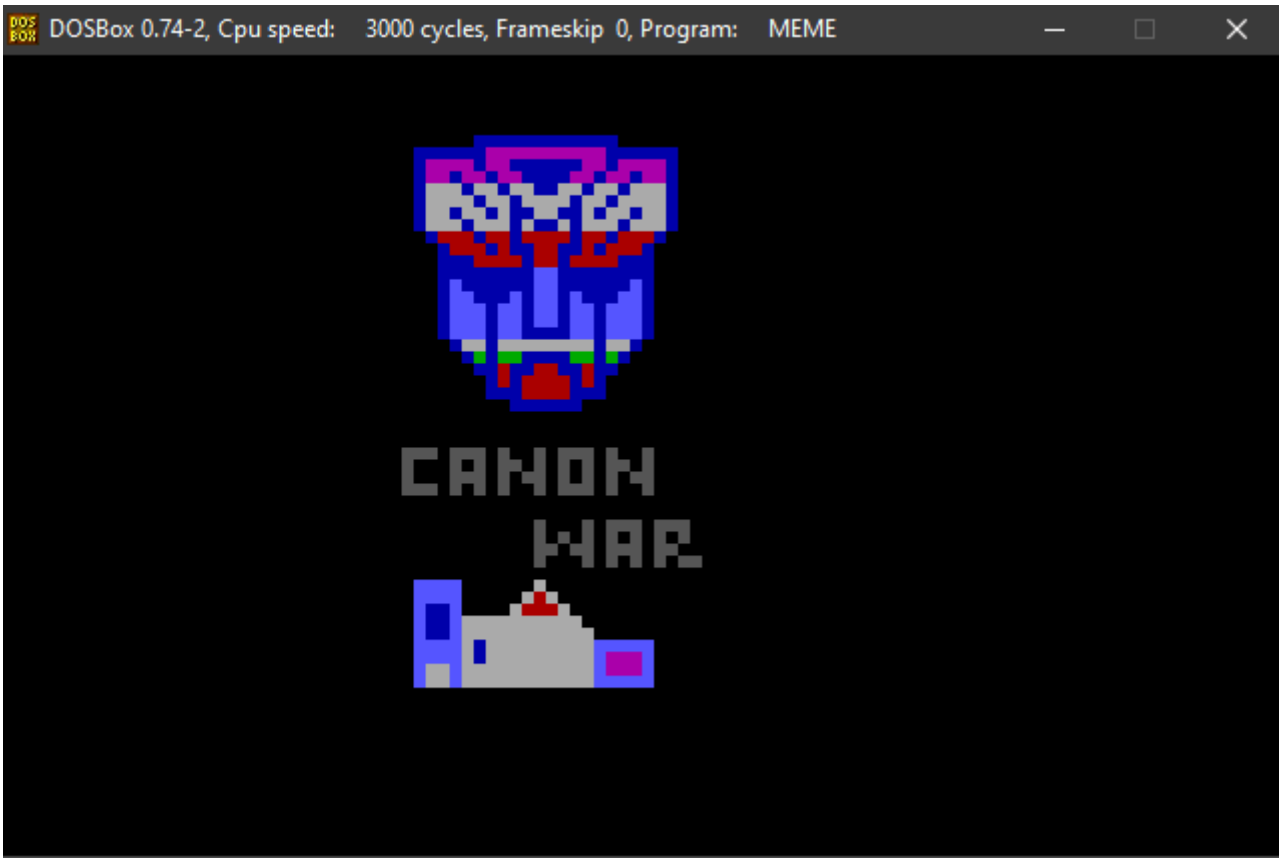
4. File Handling – Instructions (Done by Hassan):

This was done using file handling. For “Instructions”, file handling was done. We made a ***txt*** file in which we wrote the list of instructions. Then we simply read the file and displayed its contents on the console. All the instructions from file were read and stored into a string which was later printed/ displayed on the screen. So basically, when a user will click on “Instructions”, he’ll be showed a screen containing all the rules and instructions regarding how to play the game.



5. Splash Welcome Screen (Done by Hassan):

We did a bonus task of creating a welcome screen. Once the game starts, a welcome screen is displayed. On this welcome screen, there is a robot logo along with writing and some buildings. All this was done using graphics and a procedure called “Time Delay”. Time delay is basically used to create a delay. In this case, this procedure determines the amount of time we want this screen to be displayed for. In our case we are displaying it for 3-5 seconds only.

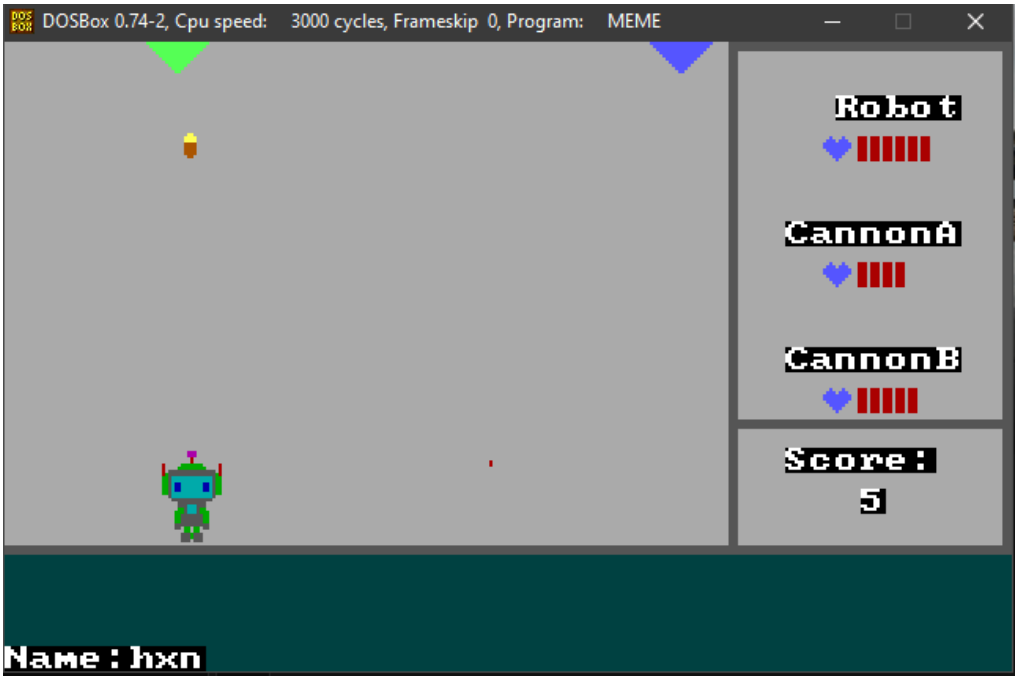


6. Damage Detection (Done by Hassan):

Damage detection was also done. In this we used multiple conditions, such as checking bullets collision with the robot, bullets collision with canon 1 and bullets collision with canon 2 and also the collision of both canons too. For this we simply compared the coordinates of the two things and if they are equal or overlapping then this shows that collision was detected, resulting in the condition being true and damage being occurred. If collision was detected, then we decreased the health of robot, canon 1, canon 2 accordingly. For example, if a bullet on canon hits the robot, then the health of the robot will decrease gradually. Similarly, if the bullet of the robot hits canon1 or canon2, then the health of canons will also decrease respectively. Once the health of a robot is finished, the game will be over. Similarly, if a canon's health is finished, the canon will be destroyed and it will stop firing. Moreover, score is also linked with damage detection. If the robot gets hit by the bullet from a canon, the score will decrease and if a canon gets hit by the bullet of a robot, the score will increase.
In the attached screenshot, you can clearly see that the health of Coal Man is decreasing and his bars are decreasing too.

7. Canon Movement (Done by Ayesha):

For the movement of canons, we opted random movement. One canon will move towards left and the other one will move towards right. One important thing to note is that there is a bounding condition for these canons that when they reach the left end or right end of the screen, then it will automatically detect the end and will change its direction 180 degrees. Meaning if it was already moving towards left, then after reaching the left end of the screen, it will change its direction and start moving towards right side and vice versa.



From the figure, you can clearly see the changed locations of both of the canons.

8. Bullets Firing from Canons (Done by Ayesha):

We also implemented the bullets firing such that both of the canons will fire bullets at different intervals. And furthermore, the bullets fired will also reach the player with different speeds. These bullets were drawn using pixels and they are moved by moving their y coordinates. The bullets when touching the robot (i.e. player) at any point will decrease their health.

The white and black bullets seen in the picture above are the bullets fired from the canons.

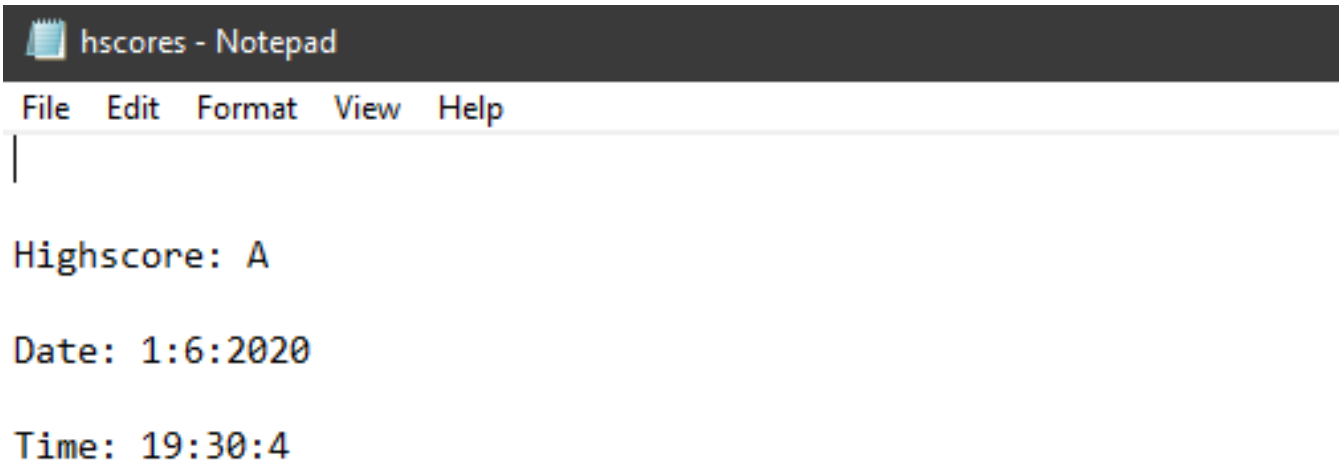
9. Bullets Firing from Robot (Done by Ayesha):

Similarly, we implemented bullets from robot. The player will also be able to fire bullets from the robots using the above arrow key ⬆ and the bullet will be fired. This bullet when hits the canon will reduce its health and increase the score of the user. User will be able to move the player along with controlling the bullets.

The black and white bullet seen in the picture above are the bullets fired from the robot's end.

10. File Handling – High Score (Done by Ayesha):

We also implemented high scores using file handling. Every time a user breaks the previous high score, his new score is entered and written on the file and the file is saved. So, the next time when a player accesses the “Score Menu”, the saved file is read again and the high score along with the time and date of the score is displayed on the screen.



11. Sound (Done by Ayesha):

We also added sound in our game. We used a procedure that does the beeping sound and we used it at different places and changed the frequency to create a different sound. The sounds we added can be heard when a bullet is fired, when a bullet hits the canon, when the bullet hits the robot, when level is changed and similarly when the game is won / lost.

```
beep proc
    mov al, 182
    out 43h, al
    mov ax, sound

    out 42h, al
    mov al, ah
    out 42h, al
    in al, 61h

    or al, 3
    out 61h, al
    ;mov cx, 3h
     mov dx, 4240h
     mov ah, 86h
     int 15h
    in al, 61h

    and al, 11111100b
    out 61h, al
ret
beep endp
```

12. Health Log (Done by Shoaib):

This part was done by using graphics for all canons and robots and it works simply to explain the user the process of game as simple when the bullet is hit to either canons or robot It causes the health bar of certain object to decrease by implying certain checks and displaying it to the user by doing it so.

You can see the health bar of the robot decreasing.

13. Pause / Resume (Done by Shoaib):

This part was done by using keyboard interrupt while playing the game user can hit spacebar to pause which will cause a message to display indicating that game is paused simple click on spacebar to resume the game from where it was paused.

14. Username (Done by Shoaib):

This part allows the user to enter the username which will take him to the menu screen the name is stored in array and in the end the array will display the name along with other attributes. The name will take the user to menu screen and further will take him to the game.

15. Win/Lose (Done by Shoaib):

This part allows the game to display the end results of the player of certain name if you destroy all the canons the user will result in winning the game else if our robot dies then the user with certain name will lose. The health bar will reach to zero resulting in destroying of the canons or robots.

16. Levels (Done by Shoaib):

This part was done by Shoaib in which a user can increase the levels of the game to challenge himself. Each increment in level results in faster pace of the game which demands more challenge from the user to play at his very best. There are maximum three levels of the game in which user can select a certain to challenge himself.

Code:

```
;bulets zyada damage canon ka zyada krna damage player ka kam krna hy
.model small
.stack 100h
.386
.data

scor db "Score:$"
cman db "Robot$"
canon1 db "CannonA$"
canon2 db "CannonB$"
play db "PLAY$"
endd db "Game End$"
inst db "INSTRUCTIONS$"
scr db "SCORE$"
exts db "EXIT$"
;ooof db "<- ->: move, Up Arrow: shoot, e: exit$"
win db "YOU WIN!!$"
los db "LOSER HAHA$"
pose db "PAUSE", 10, 13, "PRESS SPACE BAR TO RESUME$"
entname db "Enter player name: $"
ename db "Name: $"
lstring db "Press Enter to go to the next level, any other key to end the game $"
uname db 20 dup('$')
score dw 0
sound dw 0
level dw 0


                    ;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
Splash      DB      0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0 ;01
            DB      0, 1, 1, 1, 1, 1, 1, 5, 5, 5, 5, 5, 5, 5, 5, 5, 1, 1, 1, 1, 1, 1, 0, 0, 0 ;02
            DB      0, 1, 5, 5, 5, 5, 1, 5, 5, 1, 1, 1, 1, 1, 5, 5, 1, 5, 5, 5, 5, 1, 0, 0, 0 ;03
            DB      0, 1, 5, 5, 1, 5, 5, 1, 5, 5, 1, 1, 1, 1, 5, 5, 1, 5, 5, 1, 5, 5, 1, 0, 0, 0 ;04
            DB      0, 1, 7, 7, 7, 1, 7, 7, 1, 7, 7, 1, 1, 7, 7, 1, 7, 7, 1, 7, 7, 7, 1, 0, 0, 0 ;05
            DB      0, 1, 7, 7, 7, 7, 1, 7, 7, 1, 7, 7, 7, 7, 1, 7, 7, 1, 7, 7, 7, 7, 1, 0, 0, 0 ;06
            DB      0, 1, 7, 7, 1, 7, 7, 1, 7, 1, 1, 7, 7, 1, 1, 7, 1, 7, 7, 1, 7, 7, 1, 0, 0, 0 ;07
            DB      0, 1, 7, 7, 7, 1, 7, 7, 7, 1, 7, 1, 1, 7, 1, 7, 7, 1, 7, 7, 7, 1, 0, 0, 0 ;08
            DB      0, 0, 1, 4, 4, 4, 1, 4, 4, 1, 4, 4, 4, 4, 1, 4, 4, 1, 4, 4, 4, 1, 0, 0, 0, 0 ;09
            DB      0, 0, 0, 1, 4, 4, 4, 1, 4, 1, 1, 4, 4, 1, 1, 4, 1, 4, 4, 4, 1, 0, 0, 0, 0, 0 ;10
            DB      0, 0, 0, 1, 1, 1, 4, 4, 4, 4, 1, 4, 4, 1, 4, 4, 4, 4, 1, 1, 1, 0, 0, 0, 0, 0 ;11
            DB      0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 9, 9, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0 ;12
            DB      0, 0, 0, 1, 9, 1, 1, 1, 1, 1, 1, 9, 9, 1, 1, 1, 1, 1, 1, 9, 1, 0, 0, 0, 0, 0 ;13
            DB      0, 0, 0, 1, 9, 9, 1, 1, 1, 9, 1, 9, 9, 1, 9, 1, 1, 1, 9, 9, 1, 0, 0, 0, 0, 0 ;14
            DB      0, 0, 0, 1, 9, 9, 9, 1, 9, 9, 1, 9, 9, 1, 9, 9, 1, 9, 9, 9, 1, 0, 0, 0, 0, 0 ;15
```

```
                DB      0, 0, 0, 1, 9, 9, 9, 1, 9, 9, 1, 9, 9, 1, 9, 9, 1, 9, 9, 9, 1, 0, 0, 0, 0, 0 ;16
                DB      0, 0, 0, 1, 9, 9, 9, 1, 9, 9, 1, 1, 1, 1, 9, 9, 1, 9, 9, 9, 1, 0, 0, 0, 0, 0 ;17
                DB      0, 0, 0, 0, 1, 7, 7, 1, 7, 7, 7, 7, 7, 7, 7, 7, 1, 7, 7, 1, 0, 0, 0, 0, 0, 0 ;18
                DB      0, 0, 0, 0, 0, 1, 2, 1, 2, 2, 1, 1, 1, 1, 2, 2, 1, 2, 1, 0, 0, 0, 0, 0, 0, 0 ;19
                DB      0, 0, 0, 0, 0, 0, 1, 1, 4, 1, 1, 4, 4, 1, 1, 4, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0 ;20
                DB      0, 0, 0, 0, 0, 0, 0, 1, 4, 1, 4, 4, 4, 4, 1, 4, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0 ;21
                DB      0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 4, 4, 4, 4, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0 ;22
                DB      0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ;23
                DB      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ;24
                DB      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ;25
                DB      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ;26
                DB      8, 8, 8, 0, 8, 8, 8, 0, 8, 0, 0, 8, 0, 8, 8, 8, 0, 8, 0, 0, 8, 0, 0, 0, 0, 0 ;27
                DB      8, 0, 0, 0, 8, 0, 8, 0, 8, 8, 0, 8, 0, 8, 0, 8, 0, 8, 8, 0, 8, 0, 0, 0, 0, 0 ;28
                DB      8, 0, 0, 0, 8, 8, 8, 0, 8, 0, 8, 8, 0, 8, 0, 8, 0, 8, 0, 8, 8, 0, 0, 0, 0, 0 ;29
                DB      8, 8, 8, 0, 8, 0, 8, 0, 8, 0, 0, 8, 0, 8, 8, 8, 0, 8, 0, 0, 8, 0, 0, 0, 0, 0 ;30
                DB      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ;31
                DB      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ;32
                DB      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 8, 0, 0, 8, 0, 8, 8, 8, 0, 8, 8, 8, 0, 0, 0, 0 ;33
                DB      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 8, 0, 8, 0, 8, 0, 8, 0, 8, 0, 8, 0, 0, 0, 0, 0 ;34
                DB      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 8, 8, 0, 8, 8, 0, 8, 8, 8, 0, 8, 8, 0, 0, 0, 0 ;35
                DB      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 8, 0, 0, 0, 8, 0, 8, 0, 8, 0, 8, 8, 8, 0, 0, 0 ;36
                DB      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ;37
                DB      0, 9, 9, 9, 9, 0, 0, 0, 0, 0, 0, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ;38
                DB      0, 9, 9, 9, 9, 0, 0, 0, 0, 0, 7, 4, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ;39
                DB      0, 9, 1, 1, 9, 0, 0, 0, 0, 7, 4, 4, 4, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ;40
                DB      0, 9, 1, 1, 9, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ;41
                DB      0, 9, 1, 1, 9, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ;42
                DB      0, 9, 9, 9, 9, 7, 1, 7, 7, 7, 7, 7, 7, 7, 7, 9, 9, 9, 9, 9, 0, 0, 0, 0, 0, 0 ;43
                DB      0, 9, 9, 9, 9, 7, 1, 7, 7, 7, 7, 7, 7, 7, 7, 9, 5, 5, 5, 9, 0, 0, 0, 0, 0, 0 ;44
                DB      0, 9, 7, 7, 9, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 9, 5, 5, 5, 9, 0, 0, 0, 0, 0, 0 ;45
                DB      0, 9, 7, 7, 9, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 9, 9, 9, 9, 9, 0, 0, 0, 0, 0, 0 ;46

splashx dw 100
splashy dw 10


                ;1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18 19 20 21 22
Canoon1     DB      07,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,07;01
                DB      07,07,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,07,07;02
                DB      07,07,07,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,07,07,07;03
                DB      07,07,07,07,10,10,10,10,10,10,10,10,10,10,10,10,10,10,07,07,07,07;04
                DB      07,07,07,07,07,10,10,10,10,10,10,10,10,10,10,10,10,07,07,07,07,07;05
                DB      07,07,07,07,07,07,10,10,10,10,10,10,10,10,10,10,07,07,07,07,07,07;06
                DB      07,07,07,07,07,07,07,10,10,10,10,10,10,10,10,07,07,07,07,07,07,07;07
                DB      07,07,07,07,07,07,07,07,10,10,10,10,10,10,07,07,07,07,07,07,07,07;08
```

```
                DB      07,07,07,07,07,07,07,07,07,10,10,10,10,07,07,07,07,07,07,07,07,07;09
                DB      07,07,07,07,07,07,07,07,07,07,10,10,07,07,07,07,07,07,07,07,07,07;10


                ;1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18 19 20 21 22
Canoon2     DB      07,09,09,09,09,09,09,09,09,09,09,09,09,09,09,09,09,09,09,09,09,07;01
                DB      07,07,09,09,09,09,09,09,09,09,09,09,09,09,09,09,09,09,09,09,07,07;02
                DB      07,07,07,09,09,09,09,09,09,09,09,09,09,09,09,09,09,09,09,07,07,07;03
                DB      07,07,07,07,09,09,09,09,09,09,09,09,09,09,09,09,09,09,07,07,07,07;04
                DB      07,07,07,07,07,09,09,09,09,09,09,09,09,09,09,09,09,07,07,07,07,07;05
                DB      07,07,07,07,07,07,09,09,09,09,09,09,09,09,09,09,07,07,07,07,07,07;06
                DB      07,07,07,07,07,07,07,09,09,09,09,09,09,09,09,07,07,07,07,07,07,07;07
                DB      07,07,07,07,07,07,07,07,09,09,09,09,09,09,07,07,07,07,07,07,07,07;08
                DB      07,07,07,07,07,07,07,07,07,09,09,09,09,07,07,07,07,07,07,07,07,07;09
                DB      07,07,07,07,07,07,07,07,07,07,09,09,07,07,07,07,07,07,07,07,07,07;10

                ;1   2   3   4   5   6
Bullet      DB      07, 07, 07, 07, 07, 07 ;01
                DB      07, 07, 14, 14, 07, 07 ;02
                DB      07, 14, 14, 14, 14, 07 ;03
                DB      07, 14, 14, 14, 14, 07 ;04
                DB      07, 06, 06, 06, 06, 07 ;05
                DB      07, 06, 06, 06, 06, 07 ;06
                DB      07, 06, 06, 06, 06, 07 ;07
                DB      07, 06, 06, 06, 06, 07 ;08
                DB      07, 07, 06, 06, 07, 07 ;09
                DB      07, 07, 07, 07, 07, 07 ;10
                DB      07, 07, 07, 07, 07, 07 ;11
                DB      07, 07, 07, 07, 07, 07 ;12


                ;1   2   3   4   5   6
BulletC     DB      07, 07, 07, 07, 07, 07 ;01
                DB      07, 07, 07, 07, 07, 07 ;10
                DB      07, 07, 07, 07, 07, 07 ;11
                DB      07, 07, 15, 15, 07, 07 ;02
                DB      07, 15, 15, 15, 15, 07 ;03
                DB      07, 15, 15, 15, 15, 07 ;04
                DB      07, 15, 15, 15, 15, 07 ;05
                DB      07, 00, 00, 00, 00, 07 ;06
                DB      07, 00, 00, 00, 00, 07 ;07
                DB      07, 00, 00, 00, 00, 07 ;08
                DB      07, 07, 00, 00, 07, 07 ;09
                DB      07, 07, 07, 07, 07, 07 ;12
```

```
                    ;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
robut       DB      7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 5, 5, 5, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7 ;71
                DB      7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 5, 5, 5, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7 ;02
                DB      7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 4, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7 ;03
                DB      7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 4, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7 ;04
                DB      7, 7, 7, 7, 4, 7, 7, 7, 7, 7, 2, 2, 2, 2, 2, 2, 2, 7, 7, 7, 7, 7, 4, 7, 7, 7, 7, 7 ;05
                DB      7, 7, 7, 7, 4, 7, 7, 7, 7, 2, 2, 2, 2, 2, 2, 2, 2, 2, 7, 7, 7, 7, 4, 7, 7, 7, 7, 7 ;06
                DB      7, 7, 7, 7, 4, 7, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 7, 4, 7, 7, 7, 7, 7 ;07
                DB      7, 7, 7, 7, 4, 2, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 2, 4, 7, 7, 7, 7, 7 ;08
                DB      7, 7, 7, 7, 2, 2, 8, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 8, 2, 2, 7, 7, 7, 7, 7 ;09
                DB      7, 7, 7, 7, 2, 2, 8, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 8, 2, 2, 7, 7, 7, 7, 7 ;10
                DB      7, 7, 7, 7, 2, 2, 8, 3, 1, 1, 3, 3, 3, 3, 3, 3, 3, 1, 1, 3, 8, 2, 2, 7, 7, 7, 7, 7 ;11
                DB      7, 7, 7, 7, 2, 2, 8, 3, 1, 1, 3, 3, 3, 3, 3, 3, 3, 1, 1, 3, 8, 2, 2, 7, 7, 7, 7, 7 ;12
                DB      7, 7, 7, 7, 2, 2, 8, 3, 1, 1, 3, 3, 3, 3, 3, 3, 3, 1, 1, 3, 8, 2, 2, 7, 7, 7, 7, 7 ;13
                DB      7, 7, 7, 7, 2, 2, 8, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 8, 2, 2, 7, 7, 7, 7, 7 ;14
                DB      7, 7, 7, 7, 7, 7, 8, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 8, 7, 7, 7, 7, 7, 7, 7 ;15
                DB      7, 7, 7, 7, 7, 7, 7, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 7, 7, 7, 7, 7, 7, 7, 7 ;16
                DB      7, 7, 7, 7, 7, 7, 7, 7, 7, 8, 8, 8, 8, 8, 8, 8, 8, 8, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7 ;17
                DB      7, 7, 7, 7, 7, 7, 7, 7, 7, 8, 8, 8, 3, 3, 3, 8, 8, 8, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7 ;18
                DB      7, 7, 7, 7, 7, 7, 7, 7, 7, 2, 2, 8, 3, 3, 3, 8, 2, 2, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7 ;19
                DB      7, 7, 7, 7, 7, 7, 7, 7, 2, 2, 2, 8, 3, 3, 3, 8, 2, 2, 2, 7, 7, 7, 7, 7, 7, 7, 7, 7 ;20
                DB      7, 7, 7, 7, 7, 7, 7, 7, 2, 2, 8, 8, 8, 8, 8, 8, 8, 2, 2, 7, 7, 7, 7, 7, 7, 7, 7, 7 ;21
                DB      7, 7, 7, 7, 7, 7, 7, 7, 2, 2, 8, 8, 8, 8, 8, 8, 8, 2, 2, 7, 7, 7, 7, 7, 7, 7, 7, 7 ;22
                DB      7, 7, 7, 7, 7, 7, 7, 7, 2, 2, 8, 8, 8, 8, 8, 8, 8, 2, 2, 7, 7, 7, 7, 7, 7, 7, 7, 7 ;23
                DB      7, 7, 7, 7, 7, 7, 7, 7, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 7, 7, 7, 7, 7, 7, 7, 7, 7 ;24
                DB      7, 7, 7, 7, 7, 7, 7, 7, 8, 8, 7, 2, 2, 7, 2, 2, 7, 8, 8, 7, 7, 7, 7, 7, 7, 7, 7, 7 ;25
                DB      7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 2, 2, 7, 2, 2, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7 ;26
                DB      7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 8, 8, 8, 2, 8, 8, 8, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7 ;27
                DB      7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 8, 8, 8, 2, 8, 8, 8, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7 ;28
                DB      7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 8, 8, 8, 7, 8, 8, 8, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7 ;29
                DB      7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7 ;30

; central point of bullets
shotr dw 83
shotc1 dw 61
shotc2 dw 111

time db 0


; x y of bullets initial/final canon 1/2
xbic1 sword 11
xbfc1 sword 22
ybic1 dw 58
ybfc1 dw 64
```

```
xbic2 sword 11
xbfc2 sword 22
ybic2 dw 108
ybfc2 dw 114

; x/y bulets initial/final robot
xbir sword 118
xbfr sword 130
ybir dw 80
ybfr dw 86

; y coordinate of canon initial/final 1/2
yci1 dw 50
ycf1 dw 72
yci2 dw 100
ycf2 dw 122

; y coordinate of robot initial/final 1/2
yxi         dw 70
yxf dw 97

;-- generic initial coordinates
xcordi dw 0             ; x coordinate initial to draw
xcordf dw 0             ; final
ycordi dw 0
ycordf dw 0
colorOf db 0

;---movement variable for canons
movv1 dw 0
movv2 dw 0
movvb dw 0
movvcb1 dw 0
movvcb2 dw 0

Helth       DB 07, 07, 09, 07, 07, 07, 09, 07, 07, 07, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04;36
            DB 07, 09, 09, 09, 07, 09, 09, 09, 07, 07, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04
            DB 09, 09, 09, 09, 09, 09, 09, 09, 09, 07, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04
            DB 09, 09, 09, 09, 09, 09, 09, 09, 09, 07, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04
            DB 07, 09, 09, 09, 09, 09, 09, 09, 07, 07, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04
            DB 07, 07, 09, 09, 09, 09, 09, 07, 07, 07, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04
            DB 07, 07, 07, 09, 09, 09, 07, 07, 07, 07, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04
            DB 07, 07, 07, 07, 09, 07, 07, 07, 07, 07, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04, 07, 04, 04, 04

exHelth db 07,07,04,04,07,04,04,07,07 ;09
            db 07,04,15,04,04,04,04,04,07
```

```
                db 07,04,04,04,04,04,04,04,07
                db 07,07,04,04,04,04,04,07,07
                db 07,07,07,04,04,04,07,07,07
                db 07,07,07,07,04,07,07,07,07 ;06

helthx dw 0
helthy dw 0
heltcontr dw 0
helttimr dw 0

heltx dw 260
helty dw 10

;health final robot (x axis)
hfr dw 298
hifr dw 298
hfc1 dw 298
hfc2 dw 298
hifc1 dw 298
hifc2 dw 298

;counters
hlos dw 7
hc1 dw 7
hc2 dw 7
        hscore dw 17
;---file load k variable
        handle dw ?
        filename db "insturr.txt", 0
        buffer db 500 DUP('$')

;---file handling for highscores
        fname db "hscores.txt",0
        handle1 dw ?
        buffer1 db 500 dup('$')
        count db 10
        scount db 3
        oof dw ":"
        sstring db 10,13, "Highscore: "
         hss dw 0
        dstring db 10,13 ,"Date: "
        date db 10 dup(?)
        tstring db 10,13 ,"Time: "
        ttime db 8 dup (?)
        entr db 10,13
```

```
;*****************************************************************************************************************************
.code
        main proc
mov ax,@data
mov ds,ax
   ; setting video mode
   mov al, 13h
   mov ah, 00h
   int 10H


;=========================================SPLASH=============================================================================
        mov xcordi, 0
        mov xcordf, 320
        mov ycordi, 0
        mov ycordf, 200
        mov colorOf, 0
        call boxKap




        mov si, offset Splash      ;draw splash
        mov xcordi, 20
        mov xcordf, 158
        mov ycordi, 100
        mov ycordf, 178

        mov ah,0ch
        mov dx, xcordi                  ; x coordinate initial( up down)
ywalas:
        mov cx, ycordi                  ;y coordinate initial (left right)
xwalas:
        mov al,[si]
        int 10h
        inc cx
        int 10h
        inc cx
        int 10h
        inc si
        inc cx
        cmp cx, ycordf                          ; y coordinate final( left right)
        jb xwalas
        inc dx
        mov cx, ycordi
        sub si,26
```

```
xwalas2:
        mov al,[si]
        int 10h
        inc cx
        int 10h
        inc cx
        int 10h
        inc si
        inc cx
        cmp cx, ycordf                          ; y coordinate final( left right)
        jb xwalas2
        inc dx
        mov cx, ycordi
        sub si,26
xwalas3:
        mov al,[si]
        int 10h
        inc cx
        int 10h
        inc cx
        int 10h
        inc si
        inc cx
        cmp cx, ycordf                          ; y coordinate final( left right)
        jb xwalas3
        inc dx
        cmp dx, xcordf                          ; x coordinate final( up down)
        jb ywalas




;TIme delay
         mov CX, 13H
         mov DX, 4240H
         mov AH, 86H
         int 15H
;==============================================================ENTER NAME==============================================
        mov xcordi, 0
        mov xcordf, 320
        mov ycordi, 0
        mov ycordf, 200
        mov colorOf, 0
        call boxKap

        mov ch,2                ;write name
        mov cl,2
```

```
                mov si, offset entname
                call stringKap

mov si,offset uname
string1:
   mov ah,1
                int 21h
                cmp al,13
                je meenu

   mov [si],al
   inc si
                jmp string1

;=============================================M3Nu=============================================================

meenu:
;----------M3Nu-------------

                mov xcordi, 0
                mov xcordf, 320
                mov ycordi, 0
                mov ycordf, 200
                mov colorOf, 0
                call boxKap

;-----------strings with dabbe unk
;play box on menu

                mov xcordi, 110
                mov xcordf, 147
                mov ycordi, 72
                mov ycordf, 82
                mov colorOf, 2
                call boxKap

                mov ch,9                ;write play
                mov cl,54
                mov si, offset play
                call stringKap

 ;---------------------------next
                mov xcordi, 110
                mov xcordf, 210
                mov ycordi, 88
                mov ycordf, 98
                mov colorOf, 4
```

```
        call boxKap

        mov ch,11           ;write Instructions
        mov cl,54
        mov si, offset inst
        call stringKap


;------------------------------next
        mov xcordi, 110
        mov xcordf, 154
        mov ycordi, 104
        mov ycordf, 114
        mov colorOf, 0
        call boxKap

        mov ch,13           ;write score
        mov cl,54
        mov si, offset scr
        call stringKap


;----------------------------------next
        mov xcordi, 110
        mov xcordf, 147
        mov ycordi, 120
        mov ycordf, 130
        mov colorOf, 6
        call boxKap

        mov ch,15           ;write Exit
        mov cl,54
        mov si, offset exts
        call stringKap


;=================================================================================================================

;-------Mouse----------------
        mov ax,01
        int 33h

mouse:
```

```
                    mov ax, 03
                    int 33h
                    cmp bx,1
                    je  Cecks
                    ;je dispppp
                    jmp mouse
Cecks:

;---Play screen----
                    .if(cx>=210 && cx<=277 && dx>=72 && dx<=82 )
                    jmp dispppp
                    .endif
                    .if(cx>=210 && cx<=420 && dx>=88 && dx<=98)
                    jmp ifiifiifi
                    .endif
                    .if(cx>=210 && cx<=287 && dx>=104 && dx<=114)
                    jmp hsc
                    .endif
                    .if(cx>=210 && cx<=287 && dx>=120 && dx<=130)
                    jmp extt
                    .endif
                    jmp mouse


;--- PRESS SPACE BAR TO START

 inputt:
            mov ah,0h
            int 16h
;           mov kee,al
;---display---
            ; cmp al,020H
             ; je dispppp
;-----I/i----
                    ; cmp al,049H
                    ; je ifiifiifi
                    ; cmp al,069H
                    ; je ifiifiifi
;---back to menu from backspace
                    cmp al,008H
                    je meenu
                    jmp inputt

;---
hsc:
```

```
        mov xcordi, 0
        mov xcordf, 320
        mov ycordi, 0
        mov ycordf, 200
        mov colorOf, 0
        call boxKap

;---reset  background---
        mov xcordi, 0
        mov xcordf, 320
        mov ycordi, 0
        mov ycordf, 200
        mov colorOf, 0
        call boxKap

  mov ah,02h            ;reset cursor
  mov bx,0
  mov dh,0
  mov dl,0
  int 10h

        ;LOAD FILE HANDLE
                lea dx, fname           ; Load address of String "file"
                mov al, 0               ; Open file (read)
                mov ah, 3Dh             ; Load File Handler and store in ax
                int 21h
                mov handle1, ax

        ;READ FROM FILE
                mov bx, handle1         ; Move file Handle to bx
                lea dx, buffer1
                mov ah, 3Fh             ; Function to read from file
                int 21h

        ;PRINT BUFFER
                lea dx, buffer1
                mov ah, 09h
                int 21h

        ;CLOSE FILE HANDLE
                mov ah, 3Eh
                mov bx, handle1
                int 21h

jmp inputt
```

```
;--- INSTRUCTIONS TO LOAD
ififfifi:

;---reset  background---



        mov xcordi, 0
        mov xcordf, 320
        mov ycordi, 0
        mov ycordf, 200
        mov colorOf, 0
        call boxKap

  mov ah,02h          ;reset cursor
  mov bx,0
  mov dh,0
  mov dl,0
  int 10h

        ;LOAD FILE HANDLE
                lea dx, filename        ; Load address of String "file"
                mov al, 0               ; Open file (read)
                mov ah, 3Dh             ; Load File Handler and store in ax
                int 21h
                mov handle, ax

        ;READ FROM FILE
                mov bx, handle          ; Move file Handle to bx
                lea dx, buffer
                mov ah, 3Fh             ; Function to read from file
                int 21h

        ;PRINT BUFFER
                lea dx, buffer
                mov ah, 09h
                int 21h

        ;CLOSE FILE HANDLE
                mov ah, 3Eh
                mov bx, handle
                int 21h

jmp inputt
```

```
;=========================================================DISPLAY==========================================================
dispppp:

                mov ax, 02                                      ; hides the mouse
                int 33h
;------asmaan ka ganda rang----
;top box
                mov xcordi, 0
                mov xcordf, 320
                mov ycordi, 0
                mov ycordf, 160
                mov colorOf, 7
                call boxKap
;division line
                mov xcordi, 0
                mov xcordf, 320
                mov ycordi, 160
                mov ycordf, 163
                mov colorOf, 8
                call boxKap
;ground box
                mov xcordi, 0
                mov xcordf, 320
                mov ycordi, 163
                mov ycordf, 200
                mov colorOf, 196
                call boxKap

;---canons--
                mov si, offset Canoon1    ;draw canon1
                mov xcordi, 0
                mov xcordf, 10
                mov ax, yci1
                mov ycordi, ax
                mov ax, ycf1
                mov ycordf, ax
                call drawKap

                mov si, offset Canoon2    ;draw canon2
                mov xcordi, 0
                mov xcordf, 10
                mov ax, yci2
                mov ycordi, ax
                mov ax, ycf2
                mov ycordf, ax
```

```
        call drawKap



;---robot--
        mov si, offset robut        ;draw robot
        mov xcordi, 130
        mov xcordf, 160
        mov ax, yxi
        mov ycordi, ax
        mov ax, yxf
        mov ycordf, ax
        call drawKap

        ; mov ax, ycordi
        ; add ax,13
        ; mov shotr, ax
;----health---

;---1
        mov si, offset Helth        ;draw health of robot
        mov xcordi, 30
        mov xcordf, 38
        mov ycordi, 260
        mov ycordf, 298
        call drawKap

        mov ax,xcordf
        mov ycordf,ax
        mov ax,xcordi
        mov ycordi,ax

        .if(hlos > 7)
        ; mov xcordi, 299
        ; ;mov xcordf, 301
        ; mov cx, hlos
        ; sub cx, 7
        ; l3helth:
                ; mov ycordi, 30
                ; mov ycordf, 38
                ; mov ax, xcordi
                ; mov xcordf, ax
                ; add xcordf, 2
                ; add ax, 3
                ; push ax
                ; push cx
```

```
                ; mov colorOf, 04
                ; call boxKap
                ; pop cx
                ; pop ax
                ; mov xcordi, ax
        ; loop l3helth
        .endif

        mov ycordi, 30
        mov ycordf, 38
        mov ax, hifr
        mov xcordi, ax
        mov ax, hfr
        mov xcordf, ax
        mov colorOf, 07
        call boxKap


;---2
        mov si, offset Helth        ;draw health of canon1
        mov xcordi, 70
        mov xcordf, 78
        mov ycordi, 260
        mov ycordf, 298
        call drawKap

        mov ax,xcordf
        mov ycordf,ax
        mov ax,xcordi
        mov ycordi,ax

        mov ax, hifc1
        mov xcordi, ax
        mov ax, hfc1
        mov xcordf, ax
        mov colorOf, 07
        call boxKap

;--3
        mov si, offset Helth        ;draw health of canon2
        mov xcordi, 110
        mov xcordf, 118
        mov ycordi, 260
        mov ycordf, 298
        call drawKap
```

```
        mov ax,xcordf
        mov ycordf,ax
        mov ax,xcordi
        mov ycordi,ax

        mov ax, hifc2
        mov xcordi, ax
        mov ax, hfc2
        mov xcordf, ax
        mov colorOf, 07
        call boxKap

;----------side health box/ status bar----
;1
        mov xcordi, 230
        mov xcordf, 320
        mov ycordi, 0
        mov ycordf, 3
        mov colorOf, 8
        call boxKap


;2
        mov xcordi, 230
        mov xcordf, 233
        mov ycordi, 0
        mov ycordf, 160
        mov colorOf, 8
        call boxKap


;3
        mov xcordi, 317
        mov xcordf, 320
        mov ycordi, 0
        mov ycordf, 160
        mov colorOf, 8
        call boxKap


;4
        mov xcordi, 230
        mov xcordf, 320
        mov ycordi, 160
        mov ycordf, 163
        mov colorOf, 8
```

```
        call boxKap


;5
        mov xcordi, 230
        mov xcordf, 320
        mov ycordi, 120
        mov ycordf, 123
        mov colorOf, 8
        call boxKap


;------------------strings of heath box-------


        mov ch,2            ;write robot
        mov cl,73
        mov si, offset cman
        call stringKap


        mov ch,7            ;write canonA
        mov cl,71
        mov si, offset canon1
        call stringKap


        mov ch,12           ;write canonB
        mov cl,71
        mov si, offset canon2
        call stringKap

        mov ch,24           ;write neechy wali line
        mov cl,0
        mov si, offset ename
        call stringKap

        mov ch,24           ;write neechy wali line
        mov cl,5
        mov si, offset uname
        call stringKap

;---bullet--
        ; mov si, offset BulletC   ;draw bullet
```

```
        ; mov xcordi, 110
        ; mov xcordf, 121
        ; mov ycordi, 70
        ; mov ycordf, 76
        ; call drawKap


cnonn:
        ;-----------------------------------------------------------------------------------------------------------------move canons--

        ;mov ah, 02h
         ;              mov bh,00h
          ;             mov dh,10    ;row k liye
           ;            mov dl,3   ;column k liye
                        ;               int 10h


                        ;mov  dx, score
                        ;mov  ah, 02
                        ;int  21h
        ;----------------- score display--------------------

                        mov ch,16                   ;write score
                        mov cl,71
                        mov si, offset scor
                        call stringKap

        mov ah,2        ;setting cursor to show score
        mov bx,0
        mov dh,69
        mov dl,18
        int 10h

        mov ax,score
        call scoreKaP

        ; ;/////////////////////////////
        ; mov ah,2        ;setting cursor to show score
        ; mov bx,0
        ; mov dh,70
        ; mov dl,20
        ; int 10h


        ; mov ax,hscore
        ; call scoreKaP
```

```
; ;//////////////////////////////

        mov ah,0ch
        mov al, 0h
        int 21h

; Adding Time Delay

        mov cx, 01H
        .if(level > 0)
        mov cx, 0
        .endif
        mov dx, 2120H
.if(level > 0)
        mov dx, 9090H
        .if(level > 1)
                mov dx, 6060H
        .endif
.endif
        mov ah, 86H
        int 15H

;--------shooting---

        mov ah,2Ch
        int 21h
        mov time, dh
        mov ah,0
        mov al,dh
        mov dl,5
        div dl
        .if(ah == 4 && movvcb1 == 0)
        mov movvcb1,1
        mov sound, 9121
        mov cx, 0
        call beep
        .endif

        mov ah,2Ch
        int 21h
        mov time, dh
        mov ah,0
        mov al,dh
        mov dl,5
        mov bx, level
        sub dl, bl
```

```
div dl
.if(ah == 0 && movvcb2 == 0)
mov movvcb2,1
mov sound, 9121
mov cx, 0
call beep
.endif

;----------------------------extra life gand------------------------
.if(level < 2)
        .if(helttimr == 0)
                .if(heltcontr == 0)
                        mov ah,2Ch
                        int 21h
                        mov time, dl
                        mov ah,0
                        mov al,dh
                        mov dl,20
                        div dl
                .endif
                .if(ah == 9 && heltcontr == 0)
                        mov ah,2Ch
                        int 21h
                        ;mov time, dl
                        mov ah,0
                        mov al,dh
                        mov dl, 3
                        mul dl

                        mov ycordi,ax
                        mov ycordf, ax
                        add ycordf, 9
                        mov helthx, ax

                        mov si, offset exHelth      ;draw health of robot
                        mov xcordi, 132
                        mov xcordf, 138
                        ;mov ycordi, 160
                        ;mov ycordf, 169
                        call drawKap
                        mov heltcontr,1
                .endif


                .if(heltcontr == 60)
                        mov ax, helthx
```

```
                mov xcordi, ax
                mov xcordf, ax
                add xcordf, 9

        ;       mov xcordi, 160
        ;       mov xcordf, 169
                mov ycordi, 132
                mov ycordf, 138
                mov colorOf, 07
                call boxKap
                mov heltcontr, 0
        .elseif(heltcontr > 0)
                inc heltcontr
                mov si, yxi
                mov di, yxf
                .if(helthx <= di && helthx >= si)
                        .if(hlos <= 10)
                                ;add hfr, 4
                                add hifr, 4

                                ;mov heltcontr, 0
                                mov helttimr, 60
                                mov helthx, 0
                                inc hlos
                                mov ax, hifr
                                mov xcordi, ax
                                mov xcordf, ax
                                sub xcordi, 3
                                mov ycordi, 30
                                mov ycordf, 38
                                mov colorOf, 04
                                call boxKap
                                .if(hlos > 7)
                                add hfr, 4
                                .endif
                        .else
                                .if(helttimr == 0)
                                        inc score
                                        mov helttimr, 60
                                        mov helthx, 0
                                .endif
                        .endif
                .endif
        .endif
.else
        dec helttimr
```

```
                .endif
        .endif
;-------canons
.if(hc1 > 0)
        .if(yci1<=1)
        mov movv1, 0
        .endif
        .if(yci1 >= 110)
        mov movv1, 1
        .endif
.endif

.if(hc2 > 0)
        .if (yci2>=207)
        mov movv2,1
        .endif
        .if(yci2 <= 133)
        mov movv2, 0
        .endif
.endif
.if(hc1 > 0)
        .if (movv1==0)
                jmp forwrd1
        .elseif (movv1==1)
                jmp bckwrd1
        .endif
        forwrd1:
                inc ycf1
                inc yci1
                .if(movvcb1 == 0)
                        inc ybic1
                        inc ybfc1
                .endif
        ;       jmp input1
                jmp contnuee1
        bckwrd1:
                dec ycf1
                dec yci1
                .if(movvcb1 == 0)
                        dec ybic1
                        dec ybfc1
                .endif
        ;       jmp input1
                jmp contnuee1
        contnuee1:
                mov si, offset Canoon1    ;draw canon1
```

```
                        mov xcordi, 0
                        mov xcordf, 10
                        mov ax, yci1
                        mov ycordi, ax
                        mov ax, ycf1
                        mov ycordf, ax
                        call drawKap
        .endif
        .if(hc2 > 0)
                .if(movv2 == 0)
                        jmp forwrd2
                .elseif(movv2 == 1)
                        jmp bckwrd2
                .endif
                forwrd2:
                        inc ycf2
                        inc yci2
                        .if(movvcb2 == 0)
                                inc ybic2
                                inc ybfc2
                        .endif
                        jmp contnuee2
                bckwrd2:
                        dec ycf2
                        dec yci2
                        .if(movvcb2 == 0)
                                dec ybic2
                                dec ybfc2
                        .endif
                        jmp contnuee2

                contnuee2:
                        mov si, offset Canoon2    ;draw canon2
                        mov xcordi, 0
                        mov xcordf, 10
                        mov ax, yci2
                        mov ycordi, ax
                        mov ax, ycf2
                        mov ycordf, ax
                        call drawKap
        .endif
        .if(movvb==0)
                jmp cnonnfire
        .endif

contnuee3:
```

```
mov si, offset Bullet        ;draw bullet again robot

mov ax, xbir
mov xcordi, ax
mov ax, xbfr
mov xcordf, ax

mov ax, ybir
mov ycordi, ax
mov ax, ybfr
mov ycordf, ax

call drawKap

sub xbfr, 3
sub xbir, 3

.if(xbir<=0)
        mov movvb,0
        mov ycordi,0
        mov ycordf,12

        mov ax, ybir
        mov xcordi, ax
        mov ax, ybfr
        mov xcordf, ax

        mov colorOf, 07
        call boxKap
.endif
.if(hc1 > 0)
        mov si, hifc1
        mov di, 70
        mov ax, ycf1
        mov bx, yci1
        mov cx, ybir
        mov dx, xbfr
        call DamageCanon
.endif
.if(hifc1 > si)
        mov hifc1,si
        dec hc1
        .if(hc1 == 0)
                mov ax, yci1
                mov bx, ycf1
```

```
                        mov xcordi, ax
                        mov xcordf, bx
                        mov ycordi, 0
                        mov ycordf, 11
                        mov colorOf, 07
                        call boxKap
                        mov movvcb1, 0
                .endif
        .endif
        .if(hc2 > 0)
                mov si, hifc2
                mov di, 110
                mov ax, ycf2
                mov bx, yci2
                mov cx, ybir
                mov dx, xbfr
                call DamageCanon
        .endif
        .if(hifc2 > si)
                mov hifc2,si
                dec hc2
                .if(hc2 == 0)
                        mov ax, yci2
                        mov bx, ycf2
                        mov xcordi, ax
                        mov xcordf, bx
                        mov ycordi, 0
                        mov ycordf, 11
                        mov colorOf, 07
                        call boxKap
                        mov movvcb2, 0
                .endif
        .endif

        .if (hc1 == 0 && hc2 == 0)
                call winKaP
                .if(ax == 1)
                        jmp extt
                .endif
                inc level
                call nayaLevelKap
                jmp dispppp
        .endif

        ;jmp cnonn
cnonnfire:
```

```
;----------cannon bullet drawing----
;1:
.if(movvcb1 == 1)
        mov si, offset BulletC      ;draw bullet again

        mov ax,  xbic1
        mov xcordi, ax
        mov ax, xbfc1
        mov xcordf, ax

        mov ax, ybic1
        mov ycordi, ax
        mov ax, ybfc1
        mov ycordf, ax

        call drawKap

        add xbfc1, 3
        add xbic1, 3

        .if(xbfc1>=157)
                mov movvcb1,0
                mov ycordi,145
                mov ycordf,157
                mov xbic1,11
                mov xbfc1,22
                mov ax, ybic1
                mov xcordi, ax
                mov ax, ybfc1
                mov xcordf, ax

                mov ax, yci1
                mov ybic1, ax
                add ybic1,10

                mov ybfc1,ax
                add ybfc1, 16

                mov colorOf, 07
                call boxKap
                .if(hc1 == 0)
                        mov movvcb1, 2
                .endif
        .endif
.endif
```

```
        mov cx, ybic1
        mov dx, xbfc1
        call DamageRobo1

        .if (hlos == 0)
                call losKaP
                jmp extt1
        .endif

;2
.if(movvcb2 == 1)
        mov si, offset BulletC      ;draw bullet again

        mov ax,  xbic2
        mov xcordi, ax
        mov ax, xbfc2
        mov xcordf, ax

        mov ax, ybic2
        mov ycordi, ax
        mov ax, ybfc2
        mov ycordf, ax

        call drawKap

        add xbfc2, 3
        add xbic2, 3

        .if(xbfc2>=157)
                mov movvcb2,0
                mov ycordi,145
                mov ycordf,157
                mov xbic2,11
                mov xbfc2,22
                mov ax, ybic2
                mov xcordi, ax
                mov ax, ybfc2
                mov xcordf, ax

                mov ax, yci2
                mov ybic2, ax
                add ybic2,10

                mov ybfc2,ax
                add ybfc2, 16
```

```
                    mov colorOf, 07
                    call boxKap
                    .if(hc2 == 0)
                              mov movvcb2, 2
                    .endif
          .endif
.endif

          mov cx, ybic2
          mov dx, xbfc2
          call DamageRobo2

          .if (hlos == 0)
                    call losKaP
                    jmp extt1
          .endif





input1:
          mov ah,01h
          int 16h
;-----------------------------------------------------------------robot left right q(4Bh) w(4Dh) for now---
           cmp ah,4Bh
           je goLeft
                     cmp ah, 4Dh
                     je goRight
                     cmp ah, 48h                         ;up arrow key
                     je shoot
                     cmp al, 'e'                 ;escape key
                     je extt
                     .if( al == 20h)                     ;spacebar
                              mov ah, 0ch
                              int 21h
                              jmp pauseL
                     .endif

                     jmp cnonn

shoot:
          mov si, offset Bullet        ;draw bullet
          .if(movvb == 0)
                    mov xcordi, 118
                    mov xcordf, 130
                    mov xbir, 118
```

```
                mov xbfr, 130

                mov ax, shotr
                sub ax, 3
                mov ycordi, ax
                mov ybir ,ax
                mov ax, shotr
                add ax, 3
                mov ycordf, ax
                mov ybfr, ax

                call drawKap

                mov movvb,1
                mov sound, 2280
                mov cx, 1
                call beep
;               jmp contnuee3
        .endif
        jmp cnonn

goLeft:

        mov si, offset robut        ;draw robot again
        mov xcordi, 130
        mov xcordf, 160

        .if(yxi>3)
                sub yxi,3
                sub yxf,3
                sub shotr,3
                jmp contnuee
        .endif

        jmp cnonn

goRight:

        mov si, offset robut        ;draw robot
        mov xcordi, 130
        mov xcordf, 160

        .if (yxf<227)
                add yxi,3
                add yxf,3
                add shotr,3
```

```
                jmp contnuee
        .endif

contnuee:
        mov ax, yxi
        mov ycordi, ax
        mov ax, yxf
        mov ycordf, ax
        call drawKap

        jmp cnonn

;=================================================================PAUSE=============================================
pauseL:
        mov xcordi, 0
        mov xcordf, 320
        mov ycordi, 0
        mov ycordf, 200
        mov colorOf, 0
        call boxKap

        mov ch,9                ;write you pause
        mov cl,54
        mov si, offset pose
        call stringKap

        mov movv1,2
        mov movv2,2
        mov movvb,0
        mov movvcb1,0
        mov movvcb2,0


input:
        mov ah,01h
        int 16h

        cmp al,20h   ;spacebar
        je resume
        mov ah, 0ch
        int 21h
        jmp pauseL

resume:

        ; mov movv1,1
```

```
        ; mov movv2,1
        ; mov movvb,1
        ; mov movvcb1,1
        ; mov movvcb2,1

        jmp dispppp


;*****************************************************************************************************************
extt:

        mov xcordi, 0
        mov xcordf, 320
        mov ycordi, 0
        mov ycordf, 200
        mov colorOf, 0
        call boxKap

        mov ch,5
        mov cl,50
        mov si, offset endd
        call stringKap
extt1:
; Adding Time Delay
                mov cx, 13H
                mov dx, 4240H
                mov ah, 86H
                int 15H

        mov ax,score
        .if(hscore<ax)
        mov hscore,ax
        .endif

call hsKaP
call readKaP
mov si, offset buffer
mov ax,[si+14]
add ax,30h
mov hss,ax
mov ax,hscore
;.if(hss<ax)
mov hss,ax
call hsKaP
;.endif
```

```
mov ah,04ch
int 21h
main endp

;==========================================================================================PROCEDURES==============================
;-----------------------------------------------------------------damage detection---------------------
DamageRobo1 proc




        .if (cx<=yxf && cx>=yxi && dx>=130)
        mov ycordi, 30
        mov ycordf, 38

        mov ax, hfr
        mov xcordf, ax
        mov ax, hifr
        mov xcordi, ax
        sub xcordi, 4
        sub hifr, 4

        mov colorOf, 07 ;draw box over lives
        call boxKap
        .if(level == 2)
                mov ax, hfr
                mov xcordf, ax
                mov ax, hifr
                mov xcordi, ax
                sub xcordi, 4
                sub hifr, 4
                mov colorOf, 07 ;draw box over lives
                call boxKap
                dec hlos
                .if(hlos == 0)
                ret
                .endif
        .endif

        mov ax, xbic1
        mov ycordi, ax
        mov ax,xbfc1
        mov ycordf,ax
        mov xbic1,11
        mov xbfc1,22
        mov ax, ybic1
```

```
        mov xcordi, ax
        mov ax, ybfc1
        mov xcordf, ax

        mov ax, yci1
        mov ybic1, ax
        add ybic1,10

        mov ybfc1,ax
        add ybfc1, 16

        mov colorOf, 07 ;hide bullet
        call boxKap
        ;dec score
        dec hlos
        mov sound, 8609
        mov cx, 2
        call beep
        .endif



ret
DamageRobo1 endp

DamageRobo2 proc



        .if (cx<=yxf && cx>=yxi && dx>=130)
        mov ycordi, 30
        mov ycordf, 38

        mov ax, hfr
        mov xcordf, ax
        mov ax, hifr
        mov xcordi, ax
        sub xcordi, 4
        sub hifr, 4

        mov colorOf, 07 ;draw box over lives
        call boxKap

        .if(level == 2)
                mov ax, hfr
                mov xcordf, ax
```

```
                mov ax, hifr
                mov xcordi, ax
                sub xcordi, 4
                sub hifr, 4
                mov colorOf, 07 ;draw box over lives
                call boxKap
                dec hlos
                .if(hlos == 0)
                ret
                .endif
        .endif

        mov ax, xbic2
        mov ycordi, ax
        mov ax,xbfc2
        mov ycordf,ax
        mov xbic2,11
        mov xbfc2,22
        mov ax, ybic2
        mov xcordi, ax
        mov ax, ybfc2
        mov xcordf, ax

        mov ax, yci2
        mov ybic2, ax
        add ybic2,10

        mov ybfc2,ax
        add ybfc2, 16

        mov colorOf, 07 ;hide bullet
        call boxKap
        ;dec score
        dec hlos
        mov sound, 8609
        mov cx, 2
        call beep
        .endif


ret
DamageRobo2 endp


DamageCanon proc
```

```
        .if (cx<=ax && cx>=bx && dx<=10)
        mov ycordi, di
        mov ycordf, di
        add ycordf, 8

        mov xcordf, 298
        mov cx, level
        l2health:
                add xcordf, 4
        loop l2health
        sub si, 4
        mov ax, si
        mov xcordi, ax

        mov colorOf, 07 ;draw box over lives
        call boxKap

        mov ax, xbir
        mov ycordi, ax
        mov ax,xbfr
        mov ycordf,ax
        mov xbir,118
        mov xbfr,130
        mov ax, ybir
        mov xcordi, ax
        mov ax, ybfr
        mov xcordf, ax

        mov ax, shotr
        mov ybir, ax
        sub ybir,3

        mov ybfr,ax
        add ybfr, 3

        mov colorOf, 07 ;hide bullet
        call boxKap

;       .if(

        inc score
        mov sound, 1140
        mov cx, 3
```

```
                call beep
                .endif




ret
DamageCanon endp

;-----------------------------------------------------------------------------------Write string------------
stringKap proc
                mov ah,02h
   mov bh,0                                     ;color
   mov dh,ch                                                    ;y coordinate (left right)
   mov dl,cl                                    ;x coordinate (up down)
   int 10h

   lea dx , [si]                        ;string variable
   mov ah , 9
   int 21h

ret
stringKap endp

;-----------------------------------------------------------------------------------Draw strings (characters)--------
drawKap proc
                mov ah,0ch
                mov dx, xcordi                  ; x coordinate initial( up down)
ywala333:
                mov cx, ycordi                  ;y coordinate initial (left right)
xwala333:
                mov al,[si]
                int 10h
                inc si
                inc cx
                cmp cx, ycordf                                  ; y coordinate final( left right)
                jb xwala333
                inc dx
                cmp dx, xcordf                                  ; x coordinate final( up down)
                jb ywala333
ret
drawKap endp

;-----------------------------------------------------------------------------------Draw box--------------
boxKap proc
                mov ah, 0ch
                mov dx, ycordi                                  ;(up down)
```

```
dradbba1:
        mov cx,xcordi                   ;(left right)
        linebg1:
                mov al, colorOf         ;color Of box
                int 10h
                inc cx
                cmp cx,xcordf
                jl  linebg1
        inc dx
        cmp dx,ycordf
        jl  dradbba1

ret
boxKap endp

;=================================================LOSER SCREEN==============================================
losKaP proc
        mov xcordi, 0
        mov xcordf, 320
        mov ycordi, 0
        mov ycordf, 200
        mov colorOf, 0
        call boxKap

        mov ch,9                ;write you win
        mov cl,54
        mov si, offset los
        call stringKap

        mov sound, 3224
        mov cx, 2
        call beep
        mov sound, 1436
        mov cx, 5
        call beep
        mov sound, 1140
        mov cx, 12
        call beep

; Adding Time Delay
                mov cx, 03H
                mov dx, 4240H
                mov ah, 86H
                int 15H
```

```
ret
losKaP endp
;-------------------------------------------WIN KA P-----------------------------
winKaP proc
        mov xcordi, 0
        mov xcordf, 320
        mov ycordi, 0
        mov ycordf, 200
        mov colorOf, 0
        call boxKap

        mov ch,9                ;write you win
        mov cl,54
        mov si, offset win
        call stringKap



        mov sound, 7239
        mov cx, 2
        call beep
        mov sound, 8126
        mov cx, 4
        call beep
        mov sound, 9121
        mov cx, 7
        call beep
; Adding Time Delay
                mov cx, 03H
                mov dx, 4240H
                mov ah, 86H
                int 15H

        mov ah, 0ch
   int 21h

.if(level < 2)
        mov ch,15               ;write you win
        mov cl,2
        mov si, offset lstring
        call stringKap

        mov ah, 07
        int 21h

        .if(al != 13)
                mov ax, 1
```

```
                .endif

ret
.endif
mov ax, 1
ret
winKaP endp
;----------------------------SCORE---------------------------------
scoreKaP proc
  mov bx, 10
  mov dx, 0000h
  mov cx, 0000h


x1:
  mov dx, 0000h
  div bx
  push dx
  inc cx
  cmp ax, 0
  jne x1

x2:
  pop dx
  add dx, 30h
  mov ah, 02h
  int 21h
  loop x2
  ret
scoreKaP  endp
;-------------------------------------------------highscore--------------------------------------------
hsKaP proc
;//////////////////////////////////////////////////DATE//////////////////////////////////////////

                mov si,offset date

                mov ah,2Ah ;get date: date in dl, month in dh, year in cx
                int 21h
                mov ah,0
                mov al,dl     ;hours

                mov bx, 10      ;initializes divisor
                mov dx, 0000h   ;clears dx
                mov cx, 0000h   ;clears cx

l1:
```

```
   mov dx, 0000h    ;clears dx during jump
   div bx           ;divides ax by bx
   push dx          ;pushes dx(remainder) to stack
   inc cx           ;increments counter to track the number of digits
   cmp ax, 0        ;checks if there is still something in ax to divide
   jne l1           ;jumps if ax is not zero

l2:
   pop dx           ;pops from stack to dx
   add dx, 30h      ;converts to it's ascii equivalent
   mov [si],dx
   inc si
   loop l2

mov bx,oof  ;print :
mov [si],bx
inc si

                    mov ah,2Ah
                    int 21h

                    add dh,48
                    mov [si],dh                    ;month
                    inc si
                    mov bx,oof  ;print :
                    mov [si],bx
                    inc si

                    mov ax,cx

    mov bx, 10
    mov dx, 0000h
    mov cx, 0000h

l3:
   mov dx, 0000h
   div bx
   push dx
   inc cx
   cmp ax, 0
   jne l3

l4:
   pop dx
   add dx, 30h
   mov [si],dx
```

```
  inc si
  loop l4


;////////////////////////////////////////////TIME//////////////////////////////////////////

                mov si,offset ttime

                mov ah,2Ch ;hour: ch, minute cl, seconds dh
                int 21h

                mov ah,0
                mov al,ch     ;hours

   mov bx, 10
   mov dx, 0000h
   mov cx, 0000h
l5:
  mov dx, 0000h
  div bx
  push dx
  inc cx
  cmp ax, 0
  jne l5

l6:
  pop dx
  add dx, 30h
  mov [si],dx
  inc si
  loop l6

mov bx,oof  ;print :
mov [si],bx
inc si

                mov ah,2Ch
                int 21h

                mov ah,0
                mov al,cl      ;minutes

   mov bx, 10
   mov dx, 0000h
   mov cx, 0000h
l7:
```

```
   mov dx, 0000h
   div bx
   push dx
   inc cx
   cmp ax, 0
   jne l7

l8:
   pop dx
   add dx, 30h
   mov [si],dx
   inc si
   loop l8

mov bx,oof  ;print :
mov [si],bx
inc si

                mov ah,2Ch
                int 21h

                mov ah,0
                mov al,dh     ;seconds

    mov bx, 10
    mov dx, 0000h
    mov cx, 0000h
l9:
   mov dx, 0000h
   div bx
   push dx
   inc cx
   cmp ax, 0
   jne l9

l10:
   pop dx
   add dx, 30h
   mov [si],dx
   inc si
   loop l10



                        ;opening  an existing file
                        mov ah,3dh
```

```
                mov dx,offset [fname]
                mov al,1
                int 21h
                mov handle1,ax
                ;File Pointer end of file
                mov cx,0
                mov ah, 42h  ; Move file pointer
                mov al, 02h   ; End of File
                int 21h

                mov ah,40h
                mov bx,handle1
                mov cx,50
                add hss,48
                mov dx,offset[sstring]
                int 21h
                mov ah,3eh
                mov dx,handle1
                int 21h

                mov dx,offset[hss]
                int 21h
                mov ah,3eh
                mov dx,handle1
                int 21h
ret
hsKaP endp

readKaP proc
        ;LOAD FILE HANDLE
                lea dx, fname           ; Load address of String “file”
                mov al, 0               ; Open file (read)
                mov ah, 3Dh             ; Load File Handler and store in ax
                int 21h
                mov handle, ax

        ;READ FROM FILE
                mov bx, handle          ; Move file Handle to bx
                lea dx, buffer
                mov ah, 3Fh             ; Function to read from file
                int 21h

        ;CLOSE FILE HANDLE
                mov ah, 3Eh
                mov bx, handle
                int 21h
```

```
ret
readKaP endp
;=================================================================Sound==============================
beep proc
        mov al, 182
        out 43h, al
        mov ax, sound

        out 42h, al
        mov al, ah
        out 42h, al
        in al, 61h

        or al, 3
        out 61h, al
        ;mov cx, 3h
         mov dx, 4240h
         mov ah, 86h
         int 15h
        in al, 61h

        and al, 11111100b
        out 61h, al
ret
beep endp

nayaLevelKap proc

        mov hlos, 7
        mov hc1, 7
        mov hc2, 7
        mov ax, level
        mov cx, level
        add hc1, ax
        add hc2, ax
        mov hifr, 298
        mov hfr, 298
        mov hfc1, 298
        mov hfc2, 298
        mov movv1, 0
        mov movv2, 1
        mov movvb, 0
        mov movvcb1, 0
        mov movvcb2, 0
        mov hifc1, 298
```

```
        mov hifc2, 298

        lhealth:
                add hifc1, 4
                add hifc2, 4
                add hfc1, 4
                add hfc2, 4
        loop lhealth

        ; x y of bullets initial/final canon 1/2
        mov xbic1,11
        mov xbfc1,22
        mov ax, yci1
        add ax, 10
        mov ybic1, ax
        add ax, 6
        mov ybfc1, ax
        mov xbic2, 11
        mov xbfc2, 22
        mov ax, yci2
        add ax, 10
        mov ybic2, ax
        add ax, 6
        mov ybfc2, 6

        ; x/y bulets initial/final robot
        mov xbir, 118
        mov xbfr, 130
        mov ybir, 80
        mov ybfr, 86

        ret
nayaLevelKap endp
end main
```