

Aşağıdaki kodlar bazı fonksiyonların yapılma aşamalarını gösteriyor. Gösterilen fonksiyonlar: rastgele parça fonksiyonunu, Y kordinatını bulma fonksiyonu , parçayı ekrana yazdırma fonksiyonu , oyun tahtasını ekrana yazdırma fonksiyonu.|

```

7
8     int i,j;
9
10    int piece1[3][3]={1,0,0},
11                      {1,0,0},
12                      {1,1,0};//L parçamızı tanımladık
13
14    int piece2[3][3]={1,1,1},
15                      {0,0,0},
16                      {0,0,0};// - parçamızı tanımladık
17
18    int piece3[3][3]={1,1,0},
19                      {1,1,0},
20                      {0,0,0};// [] parçamızı tanımladık
21
22    int piece4[3][3]={1,1,0},
23                      {0,1,1},
24                      {0,0,0};// Z parçamızı tanımladık
25
26    int piece5[3][3]={1,0,0},
27                      {1,0,0},
28                      {0,0,0};// | parçamızı tanımladık
29
30    int piece6[3][3]={1,0,0},
31                      {0,0,0},
32                      {0,0,0};// parçamızı tanımladık
33
34    if(num=1){
35        for(i=0;i<3;i++){
36            for(j=0;j<3;j++){
37                matrix[i][j]=piece1[i][j];//piece matriximize 1.parçanın değerini atadık.
38            }
39        }
40    else if(num=2){
41        for(i=0;i<3;i++){
42            for(j=0;j<3;j++){
43                matrix[i][j]=piece2[i][j];//piece matriximize 2.parçanın değerini atadık.

```

```
34 if(num=1){
35     for(i=0;i<3;i++){
36         for(j=0;j<3;j++){
37             matrix[i][j]=piece1[i][j]; //piece matriximize 1.parçanın değerini atadık.
38         }
39     }
40     else if(num=2){
41         for(i=0;i<3;i++){
42             for(j=0;j<3;j++){
43                 matrix[i][j]=piece2[i][j]; //piece matriximize 2.parçanın değerini atadık.
44             }
45         }
46     else if(num=3){
47         for(i=0;i<3;i++){
48             for(j=0;j<3;j++){
49                 matrix[i][j]=piece3[i][j]; //piece matriximize 3.parçanın değerini atadık.
50             }
51         }
52     else if(num=4){
53         for(i=0;i<3;i++){
54             for(j=0;j<3;j++){
55                 matrix[i][j]=piece4[i][j]; //piece matriximize 4.parçanın değerini atadık.
56             }
57         }
58     else if(num=5){
59         for(i=0;i<3;i++){
60             for(j=0;j<3;j++){
61                 matrix[i][j]=piece5[i][j]; //piece matriximize 5.parçanın değerini atadık.
62             }
63         }
64     else if(num=6){
65         for(i=0;i<3;i++){
66             for(j=0;j<3;j++){
67                 matrix[i][j]=piece6[i][j]; //piece matriximize 6.parçanın değerini atadık.
68             }
69         }
70     }
```

```

49     matrix[i][j]=piece3[i][j];//piece matriximize 3.parçanın değerini atadık.
50 }
51 }
52 else if(num=4){
53     for(i=0;i<3;i++){
54         for(j=0;j<3;j++){
55             matrix[i][j]=piece4[i][j];//piece matriximize 4.parçanın değerini atadık.
56         }
57     }
58     else if(num=5){
59         for(i=0;i<3;i++){
60             for(j=0;j<3;j++){
61                 matrix[i][j]=piece5[i][j];//piece matriximize 5.parçanın değerini atadık.
62             }
63         }
64         else if(num=6){
65             for(i=0;i<3;i++){
66                 for(j=0;j<3;j++){
67                     matrix[i][j]=piece6[i][j];//piece matriximize 6.parçanın değerini atadık.
68                 }
69             }
70         }
71     }
72 }
73 int main(){
74     char X='a',Y='a';//karakter variablelarımızı tanımladık
75
76     int x=0,y,z,N,M,Highscore=0,RandNum,Score=0;//integer variablelarımızı tanımladık
77
78     int piece[3][3]={0};//bu bizim rastgele blockumuz
79
80     while (X!='n') {//While döngüsünde X ile koşul kurduk.Burada X, kullanıcı oyun bitiminde oyunu tekrar oynamak istemezse while'ı kıracak.
81
82         printf("Welcome to my Tetris game\n");//menümüz
83
84         printf("Current Highscore:%d\n",Highscore);//Highscore gösterdik
85

```

[*] Proje ödevi.c

```
43     matrix[i][j] = pieces[num-1][i][j]; //piece matriximize num sayısına denk gelen parçayı atadık
44 }
45 }
46 }
47 }
48
49 void printpiece(int matrix[3][3]){//parçayı ekrana yazdıracak fonksiyon tanımladık
50     int i,j;
51     printf("\nCurrent Piece: ")
52     for (i=0;i<3;i++) {
53         for (j=0;j<3;j++) {
54             if (matrix[i][j]==0)//matrixin elemanı 0 ise boşluk, 1 ise 1 yazdıracak
55                 printf(" ");
56             else
57                 printf("1");
58         }
59         printf("\n");
60     }
61 }
62
63 void printboard(int board[][],int N,int M){//boardı ekrana yazdıracak fonksiyon tanımladık
64     int i,j;
65     for (i=0;i<N;i++) {
66         for (j=0;j<M;j++) {
67             if (board[i][j] == 0)
68                 printf("0 ");
69             else
70                 printf("1 ");
71         }
72         printf("\n");
73     }
74 }
75
76
77 int main(){
78
79     srand(time(0)); //aynı sayıyı almamak için zamanı sıfırladık
80 }
```


Compile Log Debug Find Results

Sel: 0 Lines: 118 Length: 3247 Insert Done parsing in 0,234 seconds

```

98
99 int Ycordinate(int N,int M,int board[N][M],int piece[3][3],int Xc){//Y kordinatımızı bulacak fonksiyon tanımladık
100     int i=1,j=0,Yc=N,x=2,k=1,Yc1=N,Yc2=N;
101     while (i < N && j < 3) {while koşulumuzu belli ettik
102
103         while (j < 3 && board[i][Xc + j] != 1) {//Üsten başlayarak boardda 1 değeri aradık
104             j++;
105         }
106         if (j == 3) {// bu koşul sağlanıyorsa 1 değeri bulunmamıştır
107             j = 0;//j değerimizi sıfırladık
108             i++;//altaki satırdan devam ederiz
109             continue;//altaki koşulları atladık
110         }
111
112         if (piece[x][j] == 1 && board[i][Xc + j] == 1) {//piecein en altı satırı ile board kesişiyormu diye bakarız
113             Yc=i-1;//Yc değerimiz
114             break;//while döngüsünden çıkıyoruz
115         }
116         else if (piece[x - 1][j] == 1 && board[i][Xc + j] == 1) {//piecein orta satırı ile board kesişiyormu diye bakarız
117             Yc1=i;//Yc1 değerimiz
118         }
119         else if (piece[x - 2][j] == 1 && board[i][Xc + j] == 1) {//piecein en üst satırı ile board kesişiyormu diye bakarız
120             Yc2=i+1;//Yc2 değerimiz
121         }
122         j=0;
123         i++;
124     }
125     if(i == N){//boarda 1 yoksa en alt kısım bizim Yc kordinatımızdır
126         Yc=N-1;
127     }
128     if (Yc <= Yc1 && Yc <= Yc2) {
129         return Yc;
130     }
131 }

```

 Compile Log
  Debug
  Find Results

Sel: 0 Lines: 287 Length: 9714 Insert Done parsing in 0,016 seconds


```

109         continue;//altaki koşulları atladık
110     }
111
112     if (piece[x][j] == 1 && board[i][Xc + j] == 1) {//piecein en altı satırı ile board kesişiyormu diye bakarız
113         Yc=i-1;//Yc değerimiz
114         break;//while döngüsünden çıkıyoruz
115     }
116     else if (piece[x - 1][j] == 1 && board[i][Xc + j] == 1) {//piecein orta satırı ile board kesişiyormu diye bakarız
117         Yc1=i;//Yc1 değerimiz
118     }
119     else if (piece[x - 2][j] == 1 && board[i][Xc + j] == 1) {//piecein en üst satırı ile board kesişiyormu diye bakarız
120         Yc2=i+1;//Yc2 değerimiz
121     }
122     j=0;
123     i++;
124 }
125 if(i == N){//boarda 1 yoksa en alt kısım bizim Yc kordinatımızdır
126     Yc=N-1;
127 }
128 if (Yc <= Yc1 && Yc <= Yc2) {
129     return Yc;
130 }
131 else {
132     if (Yc1 < Yc && Yc1 < Yc2) {
133         return Yc1;
134     }
135     else {
136         return Yc2;
137     }
138 }
139 }
140

```

<https://youtu.be/ryyKRkbG7CU>

Liste dışı video olduğu için videoyu görmek için bu linke ihtiyacınız var.

Aşağıdaki kodlar prototip kodumu gösteriyor. Aslında kodum prototype.c olarak zip dosyasının içine koymak istemişim. ama sonra hocalar yanlış anlamasın diye pdf dosyasına eklemeye karar verdim. Aşağıdaki kod orjinal 23011914.c kodumun beta sürümü. kodda bazı hataların farkına vardım özellikle Y kordinatını bulma fonksiyonu ile ilgili hatalar.Sonrasında bu hataları düzelterip yeni ve gelişmiş 23011914.c kodumu yaptım.

```
1  #include <stdio.h>
2  #include <stdlib.h> //rand() fonksiyonunu kullanmak için bu kütüphaneyi ekledim
3  #include <time.h> //program zamanı kullanarak bize rastgele bir sayı vereceği için bu kütüphaneyi ekledim
4
5  int score = 0; //global variable tanımladım
6  int highscore = 0; //global variable tanımladım
7
8  void randomblock(int matrix[3][3], int num) { //kolaylık olsun diye rastgele block fonksiyonu tanımladık
9      int i, j;
10     int pieces[7][3][3] = { //parçalarımızdan oluşan bir küp tanımladık if else zincirinden kurtulmak için
11         {{1,0,0},
12          {1,0,0},
13          {1,1,0}}, // L parçası
14
15         {{0,0,0},
16          {0,0,0},
17          {1,1,1}}, // - parçası
18
19         {{0,0,0},
20          {1,1,0},
21          {1,1,0}}, // [] parçası
22
23         {{0,0,0},
24          {1,1,0},
25          {0,1,1}}, // Z parçası
26
27         {{1,0,0},
28          {1,0,0},
29          {1,0,0}}, // | parçası
30
31         {{0,0,0},
32          {0,0,0},
33          {1,0,0}}, // . parçası
34
35         {{0,0,0},
36          {1,1,1},
37          {0,1,0}} // T parçası
38     }
```

```
37     {0,1,0}}// T parçası
38 };
39
40 for (i=0;i<3;i++) {
41     for (j=0;j<3;j++) {
42         matrix[i][j] = pieces[num-1][i][j]; //piece matriximize num sayısına denk gelen parçayı atadık
43     }
44 }
45 }
46
47 void printpiece(int matrix[3][3]){ //parçayı ekrana yazdıracak fonksiyon tanımladık
48     int i,j;
49     printf("\nCurrent Piece: \n");
50     for (i=0;i<3;i++) {
51         for (j=0;j<3;j++) {
52             if (matrix[i][j]==0) //matrixin elemanı 0 ise boşluk, 1 ise 1 yazdıracak
53                 printf("0 ");
54             else
55                 printf("1 ");
56         }
57         printf("\n");
58     }
59     printf("^-----\n");
60 }
61
62 void printboard(int N,int M,int board[N][M]){ //boardı ekrana yazdıracak fonksiyon tanımladık
63     int i,j;
64     printf("\nCurrent score: %d\n",score); //şuanki scoreumuzu gösterdik
65     for (i=0;i<N;i++) {
66         for (j=0;j<M;j++) {
67             if (board[i][j] == 0) //eğer board 0 sa 0, 1 se 1 yazdırdık
68                 printf("0 ");
69             else
70                 printf("1 ");
71         }
72         printf("\n");
73     }
74 }
```

```

73     }
74     for(j=0;j<M;j++)
75     printf("--");
76
77     printf("\n");
78
79     for(j=1;j<=M;j++)
80     printf("%d ",j);
81
82     printf("\n");
83 }
84
85 void rotatepiece(int matrix[3][3]){//parçamızı döndürecek fonksiyon tanımladık
86     int matrix2[3][3];//geçici olarak parçamızın değerlerini saklayacak matrix tanımladık
87     int i,j;
88     for (i=0;i<3;i++) {
89         for (j=0;j<3;j++)
90             matrix2[i][j]=matrix[i][j];//parçamızı geçici matrixe kopyaladık
91     }
92     for (i=0;i<3;i++) {
93         for (j=0;j<3;j++)
94             matrix[i][j]=matrix2[2-j][i];//matrixi saat yönünde döndürmek için bu işlemi kullandık
95     }
96 }
97
98 int Ycoordinate(int N,int M,int board[N][M],int piece[3][3],int Xc){//Y kordinatımızı bulacak fonksiyon tanımladık
99     int i=1,j=0,Yc=N,x=2,k=1,Yc1=N,Yc2=N;
100     while (i < N && j < 3) { //while koşulumuzu belli ettik
101
102         while (j < 3 && board[i][Xc + j] != 1) { //Üsten başlayarak boardda 1 değeri aradık
103             j++;
104         }
105         if (j == 3) { // bu koşul sağlanıyorsa 1 değeri bulunmamıştır
106             j = 0; //j değerimizi sıfırladık
107             i++; //altaki satırdan devam ederiz
108             continue; //altaki koşulları atladık
109         }

```

```

109     }
110
111     if (piece[x][j] == 1 && board[i][Xc + j] == 1) { //piecein en altı satırı ile board kesişiyormu diye bakarız
112         Yc=i-1; //Yc değerimiz
113         break; //while döngüsünden çıkıyoruz
114     }
115     else if (piece[x - 1][j] == 1 && board[i][Xc + j] == 1) { //piecein orta satırı ile board kesişiyormu diye bakarız
116         Yc1=i; //Yc1 değerimiz
117     }
118     else if (piece[x - 2][j] == 1 && board[i][Xc + j] == 1) { //piecein en üst satırı ile board kesişiyormu diye bakarız
119         Yc2=i+1; //Yc2 değerimiz
120     }
121     j=0;
122     i++;
123 }
124 if(i == N){ //boarda 1 yoksa en alt kısım bizim Yc kordinatımızdır
125     Yc=N-1;
126 }
127 if (Yc <= Yc1 && Yc <= Yc2) {
128     return Yc;
129 }
130 else {
131     if (Yc1 < Yc && Yc1 < Yc2) {
132         return Yc1;
133     }
134     else {
135         return Yc2;
136     }
137 }
138 }
139
140 void placepiece(int N,int M,int board[N][M],int piece[3][3],int Xc,int Yc){ //parçamızı boarda koyacak bir fonksiyon tanımladık
141     int i,j,x=2,y=-1;
142
143     for (i=0;i<3 && y== -1;i++) { //bastırmaya başlayacağımız satırı arıyoruz
144         for (j=0;j<3;j++) {
145             if (piece[x - i][j] == 1) {

```

```

145     if (piece[x - i][j] == 1) {
146         y = x - i; //başlama satırımızı bulduk
147         j=4; //satırı bulduktan sonra döngülerden çıkıyoruz
148     }
149 }
150 }
151
152 if (y!=-1) { //eğer bir başlama satırı bulduysak o satırdan başlayarak boarda yazıyoruz
153     for (i=0; i<3 && Yc - i>=0; i++) {
154         for (j=0; j<3; j++) {
155             if (piece[y - i][j] == 1) { //sadece 1 değerlerimizi boarda bastırıyoruz
156                 board[Yc - i][Xc + j] = 1; //bu işlemle başlama satırından başlayıp değerlerimizi boarda yazıyoruz
157             }
158         }
159     }
160 }
161 }
162
163 int checklines(int N, int M, int board[N][M]) {
164     int i, j, x=1, y=0, k;
165     for (i=N-1; i>=0; i--) {
166         x=1; //x değerimizi her satır için tekrar 1 e eşitliyoruz
167         for (j=0; j<M; j++) {
168             x *= board[i][j]; //eğer satırımızda her sütün 1 se o zaman x değeri 1 kalacak
169         }
170         if (x == 1) { //eğer satırımız doluyorsa bu koşul sağlanacak
171             for (j=0; j<M; j++) {
172                 board[i][j] = 0; //dolu satırı sildik
173                 score++; //puanı artırdık
174             }
175
176             for (k=i; k>0; k--) { //üste kalan 1 lerimizi aşağı kaydırmak için sildiğimiz satırın üstündeki satırlar 1 aşağı indirilir
177                 for (j=0; j<M; j++) {
178                     board[k][j] = board[k - 1][j]; //üst sıradaki değerleri alt satıra kopyaladık
179                 }
180             }
181             for (j=0; j<M; j++) { //en üste kalan elemanları sildik

```



```

181     for (j=0;j<M;j++) { //en üste kalan elemanları sildik
182         board[1][j] = 0;
183     }
184 }
185 }
186 y = 0;
187 for (j = 0; j < M; j++) { //en üst satırımızda 1 var mı diye kontrol ederiz
188     y += board[0][j];
189 }
190 if( y == 0 ){
191     return 0; //bir yoksa 0
192 }
193 else if( y > 0 ){
194     return 1; // bir varsa 1 gönderdik
195 }
196 }
197
198 int canplace(int M,int piece[3][3],int Xc){ //eğer bize verilen x kordinatı boardın dışındamı değil diye bakan fonksiyon
199     int i,j,y,x;
200     for (i=2;i>=0;i--) {
201         for (j=0;j<3;j++) { //bastırmaya başlayacağımız sütunu arıyoruz
202             if(piece[j][i] == 1){
203                 y=i+1; //bastırmaya başlayacağımız sütunu bulduk
204                 i=-1; //for koşullarını break kullanmadan kırmak için bunu kullanıyoruz
205                 j=4; //for koşullarını break kullanmadan kırmak için bunu kullanıyoruz
206             }
207         }
208     }
209     for (i=0;i<3;i++) {
210         for (j=0;j<3;j++) { //bastırmaya başlayacağımız sütunu arıyoruz
211             if(piece[j][i] == 1){
212                 x=i+1; //bastırmaya başlayacağımız sütunu bulduk
213                 i=4; //for koşullarını break kullanmadan kırmak için bunu kullanıyoruz
214                 j=4; //for koşullarını break kullanmadan kırmak için bunu kullanıyoruz
215             }
216         }
217     }
218     if( Xc < 0 ) {

```

```
217 }
218 if( Xc < 0 ) {
219     if ( Xc + x > 0 ){
220         return 1;
221     }
222     else
223         return 0;
224 }
225 if( Xc + y > M ){
226     return 0;
227 }
228 else if( Xc + y <= M ){
229     return 1;
230 }
231 }
232
233 int main(){
234
235     srand(time(0)); //aynı sayıyı almamak için zamanı sıfırladık
236
237     char X='a',Y='a',x='a'; //karakter variablelarımızı tanımladık
238
239     int xcordinate,ycordinate,N,M,RandNum,i,j,Gameover,can; //integer variablelarımızı tanımladık
240
241     int piece[3][3]={0}; //bu bizim rastgele blockumuz
242
243     while (X!='n') { //While döngüsünde X ile koşul kurduk. Burada X, kullanıcı oyun bitiminde oyunu tekrar oynamak istemezse while'ı kırarak.
244
245         Y='a'; //tekrar oynanmak isterse Y'yi resetledik
246
247         printf("\nWelcome to my Tetris game\n"); //menümüz
248
249         printf("\nCurrent Highscore:%d\n",highscore); //Highscore gösterdik
250
251         printf("\nPlease enter the boards Row Number min(5) max(20):");
252         scanf(" %d",&N);
253
254         while (N<5 || N>20) { //eğer tekrar dağama oynadıkta değilse tekrar dağama oynamak istemezse
```

```

253
254 while (N<5 || N>20){//eğer tanımladığımız aralıkta değilse tekrar değer alıyoruz
255     printf("wrong");
256     printf("\nPlease enter the boards Row Number min(5) max(20):");
257     scanf(" %d",&N);
258 }
259
260 printf("Please enter the boards Column Number min(5) max(10):");
261 scanf(" %d",&M);
262
263 while (M<5 || M>10){//eğer tanımladığımız aralıkta değilse tekrar değer alıyoruz
264     printf("wrong");
265     printf("\nPlease enter the boards Column Number min(5) max(10):");
266     scanf(" %d",&N);
267 }
268
269 int board[N][M];//oyun tahtasını kullanıcıdan aldığımız bilgilerle tanımladık
270
271 for (i=0;i<N;i++) {
272     for (j=0;j<M;j++)
273         board[i][j]=0;//oyun tahtasını sıfırladık
274 }
275 while(Y!='n'){//burada Y oyuncu oyunu kaybederse veya oyundayken çıkmak isterse whileı kırıp ana menüye geri dönecek.
276
277     RandNum=rand() % 7 + 1;// 7 parçamız olduğu için 7den 1e rastgele sayı istiyoruz
278
279     randomblock(piece,RandNum);//randomblock fonksiyonuyla rastgele parçamızı elde ettik
280
281     printboard(N,M,board);//boardı ve score u kullanıcıya gösterdik
282
283     printpiece(piece);//parçamızı kullanıcıya gösterdik
284
285     printf("\nDo you want to rotate the piece to the right? (y,n)\n");//kullanıcı parçayı sağa döndürmek istiyormu diye sorduk
286     scanf(" %c",&x);
287
288     while (x!='n'){//eğer kullanıcı soruya no yani n ile cevap verirse parça döndürülmeyecek
289         rotatepiece(piece);//parçayı sağa döndürdük
290     }

```

```
289 rotatepiece(piece); //parçayı sağa döndürdük
290
291 printpiece(piece); //parçayı tekrar kullanıcıya gösterdik
292
293 printf("\nDo you want to rotate the piece to the right again? (y,n)"); //kullanıcıya tekrar parçayı sağa döndürmek istiyormu diye sorduk
294 scanf(" %c",&x);
295 } //kullanıcıya parçayı sonsuz defa döndürme hakkı verdik
296
297 printboard(N,M,board); //boardı ve score u kullanıcıya gösterdik
298
299 printpiece(piece); //parçamızı kullanıcıya gösterdik
300
301 printf("\nPlease enter the x cordinate of the pieces lower left side:"); //kullanıcıdan sol alt köşenin x kordinatını istedik
302 scanf(" %d",&xcordinate);
303
304 xcordinate--; // Bize 1 verilirse 0. sütundan bahsediliyordur
305
306 can=canplace(M,piece,xcordinate); //verilen x kordinatını boarda yerleştirebiliyormuyuz diye kontrol ediyoruz
307
308 while (can == 0){ //eğer bu koşul sağlanırsa bize verilen x kordinatı boarda yerleştirilmez
309     printf("\nThe given x cordinate isn't valid\n");
310
311     printpiece(piece); //parçayı tekrar kullanıcıya gösterdik
312
313     printf("\nPlease enter the x cordinate of the pieces lower left side:"); //kullanıcıdan tekrar sol alt köşenin x kordinatını istedik
314     scanf(" %d",&xcordinate);
315
316     can=canplace(M,piece,xcordinate); //verilen x kordinatını boarda yerleştirebiliyormuyuz diye kontrol ediyoruz
317 }
318
319 ycoordinate=Ycoordinate(N,M,board,piece,xcordinate); //Y kordinatımızı bulduk
320
321 placepiece(N,M,board,piece,xcordinate,ycoordinate); //parçamızı boarda yerleştirdik
322
323 Gameover=checklines(N,M,board); //burada satırları kontrol ediyoruz,dolu satırları yok edip puana ekliyoruz, ayrıca game over mı diye kontrol ediyoruz
324
325 printboard(N,M,board); //boardı ve score u kullanıcıya gösterdik
326
```



```
313     printf("\nPlease enter the x cordinate of the pieces lower left side:");//kullanıcıdan tekrar sol alt köşenin x kordinatını istedik
314     scanf(" %d",&xcordinate);
315
316     can=canplace(M,piece,xcordinate);//verilen x kordinatını boarda yerleştirebiliyormuyuz diye kontrol ediyoruz
317 }
318
319 ycoordinate=Ycoordinate(N,M,board,piece,xcordinate);//Y kordinatımızı bulduk
320
321 placepiece(N,M,board,piece,xcordinate,ycoordinate);//parçamızı boarda yerleştirdik
322
323 Gameover=checklines(N,M,board);//burada satırları kontrol ediyoruz,dolu satırları yok edip puana ekliyoruz,ayrıca game over mı diye kontrol ediyoruz
324
325 printboard(N,M,board);//boardı ve score u kullanıcıya gösterdik
326
327 if(Gameover == 1){//eğer bu şart sağlanırsa kullanıcı oyunu kaybetmiş demektir
328     printf("Game over!!!\n");
329     printf("Your final score:%d\n",score);//kullanıcıya son scoreunu gösterdik
330     Y='n';
331     printf("Do you want to play again? (y,n):");//kullanıcıya tekrar oynamak istiyor mu diye sorduk
332     scanf(" %c",&X);
333     if (score > highscore){//eğer puanımız Highscore dan büyükse highscore olur
334         highscore=score;
335         printf("CONGRATS YOU HAVE BEATEN THE HIGHSCORE");//kullanıcıya highscoreu geçtiğini söyledik
336     }
337 }
338 else if(Gameover == 0) {//eğer bu şart sağlanırsa kullanıcı oyuna devam edebilir
339     printf("\nDo you want to continue this game? (y,n):");//kullanıcıya oyuna devam etmek istiyormu diye sorduk
340     scanf(" %c",&Y);
341     if(Y == 'n') {
342         printf("\nDo you want to play again? (y,n):");//kullanıcıya tekrar oynamak istiyor mu diye sorduk
343         scanf(" %c",&X);
344     }
345 }
346 }
347 }
348 return 0;
349 }
```