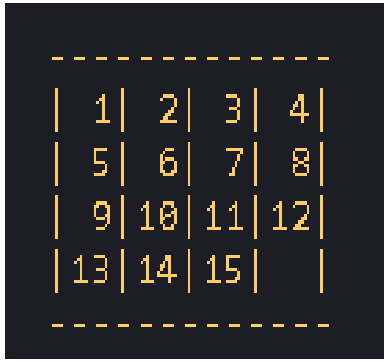
PROJECT

Matrix Puzzle

INTRODUCTION-

- In this project we are required to solve the matrix and arrange the elements of the matrix in sequential order.



Like this

- The elements of the matrix would be placed randomly. In order to move the elements, we must use arrow keys.
- Up arrow key ’ ↑’ makes the blank space to move in upward direction.
- Down arrow key ’ ↓’ makes the blank space to move in downward direction.
- Left arrow key ’ ←’ makes the blank space to move in leftward direction.
- Right arrow key ’ →’ makes the blank space to move in rightward direction.

Coding

```c
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
int arr[4][4], j = 0;
void check(int a[], int num) // checking whether the element is present in the array
{
    for (int i = 0; i < j; i++)
    {
        if (num == a[i])
        {
            return;
        }
    }
    a[j++] = num;
    return;
}
void createarray() // creating matrix
{
    int a[16];
    int lower = 1, upper = 15;
    int i, count = 15;
    while (j < 15)
    {
        int num = (rand() % (upper - lower + 1)) + lower;
        check(a, num);
    }
    a[15] = 0;
    int k = 0;
    for (i = 0; i < 4; i++)
    {
        for (j = 0; j < 4; j++)
            arr[i][j] = a[k++];
```

```
    }
}

int stop = 1;
void display() // Display screen
{
    system("cls");
    printf("\033[0;32m");
    printf("\t\t\t\t\t\tMatrix Puzzle");
    printf("\033[0;31m");
    printf("\n\n\t\t\tRULES OF THIS GAME: ");
    printf("\n\n\t\t1.You can only move 1 step at a time using arrow keys: ");
    printf("\033[0;30m");
    printf("\n\t\t\t\t-To move up- Use upward arrow key.");
    printf("\n\t\t\t\t-To move down- Use downward arrow key.");
    printf("\n\t\t\t\t-To move left- Use leftward arrow key.");
    printf("\n\t\t\t\t-To move right- Use rightward arrow key.");
    printf("\033[0;31m");
    printf("\n\n\t\t2.You can move numbers at empty position only.");
    printf("\n\n\n\t\t3.Winning situation: Number in a 4*4 matrix should be in order from 1 to 15.");
    printf("\033[0;30m");
    printf("\n\t\t\tWINNING SITUATION:");
    printf("\n\033[0;33m");
    printf("\n\t-------------");
    printf("\n\t| 1| 2| 3| 4|");
    printf("\n\t| 5| 6| 7| 8|");
    printf("\n\t| 9|10|11|12|");
    printf("\n\t|13|14|15|  |");
    printf("\n\t-------------");
    printf("\n\033[0;30m");
    printf("\t\t5. You can exit the game anytime by pressing 'E' or 'e' key.");
    printf("\n\n\tPress enter key to start....: ");
```

```
    printf("\n\n\tPress enter key to start....: ");
    getchar();
    system("cls");
}
void print()
{
    printf("\033[0;36m");
    printf("\n-------------\n");
    for (int i = 0; i < 4; i++)
    {
        for (int j = 0; j < 4; j++)
        {
            if (arr[i][j] == 0)
                printf("|  ");
            else if (arr[i][j] > 9)
                printf("|%d", arr[i][j]);
            else
                printf("| %d", arr[i][j]);
            if (j == 3 || j == 6 || j == 9 || j == 12)
                printf("|");
        }
        printf("\n");
    }
    printf("-------------");
}
int readEnteredKey() // it checks which arrow key did user press
{
    char c;
    c = _getch();
    if (c == -32)
        c = _getch();

    return c;
```

```
96      return c;
97  }
98  void swap(int *val1, int *val2) // for moving the elements
99  {
100     int temp;
101     temp = *val1;
102     *val1 = *val2;
103     *val2 = temp;
104 }
105 void arrowup() // for up arrow key
106 {
107     int i, j;
108     for (i = 0; i < 4 && stop; i++)
109     {
110         for (j = 0; j < 4; j++)
111         {
112             if (arr[i][j] == 0)
113             {
114                 stop = 0;
115                 break;
116             }
117         }
118     }
119     i--;
120     if (i == 0)
121     {
122         system("cls");
123         print();
124         printf("\033[1;31m");
125         printf("\nInvalid move");
126         printf("\npress enter key...");
127         getchar();
```

```
127         getchar();
128     }
129     else
130     {
131         swap(&arr[i][j], &arr[i - 1][j]);
132     }
133     system("cls");
134 }
135 void arrowdown() // for down arrow key
136 {
137     int i, j;
138     for (i = 0; i < 4 && stop; i++)
139     {
140         for (j = 0; j < 4; j++)
141         {
142             if (arr[i][j] == 0)
143             {
144                 stop = 0;
145                 break;
146             }
147         }
148     }
149     i--;
150     if (i == 3)
151     {
152         system("cls");
153         print();
154         printf("\033[1;31m");
155         printf("\nInvalid move");
156         printf("\npress enter key...");
157         getchar();
158     }
159     else
```

```
159     else
160     {
161         swap(&arr[i][j], &arr[i + 1][j]);
162     }
163     system("cls");
164 }
165 void arrowleft() // for left arrow key
166 {
167     int i, j;
168     for (i = 0; i < 4 && stop; i++)
169     {
170         for (j = 0; j < 4; j++)
171         {
172             if (arr[i][j] == 0)
173             {
174                 stop = 0;
175                 break;
176             }
177         }
178     }
179     i--;
180     if (j == 0)
181     {
182         system("cls");
183         print();
184         printf("\033[1;31m");
185         printf("\nInvalid move");
186         printf("\npress enter key...");
187         getchar();
188     }
189     else
190     {
```

matrixpuzzle.c > main()

```
180     if (j == 0)
181     {
182         system("cls");
183         print();
184         printf("\033[1;31m");
185         printf("\nInvalid move");
186         printf("\npress enter key...");
187         getchar();
188     }
189     else
190     {
191         swap(&arr[i][j], &arr[i][j - 1]);
192     }
193     system("cls");
194 }
195 void arrowright() // for right arrow key
196 {
197     int i, j;
198     for (i = 0; i < 4 && stop; i++)
199     {
200         for (j = 0; j < 4; j++)
201         {
202             if (arr[i][j] == 0)
203             {
204                 stop = 0;
205                 break;
206             }
207         }
208     }
209     i--;
210     if (j == 3)
211     {
```

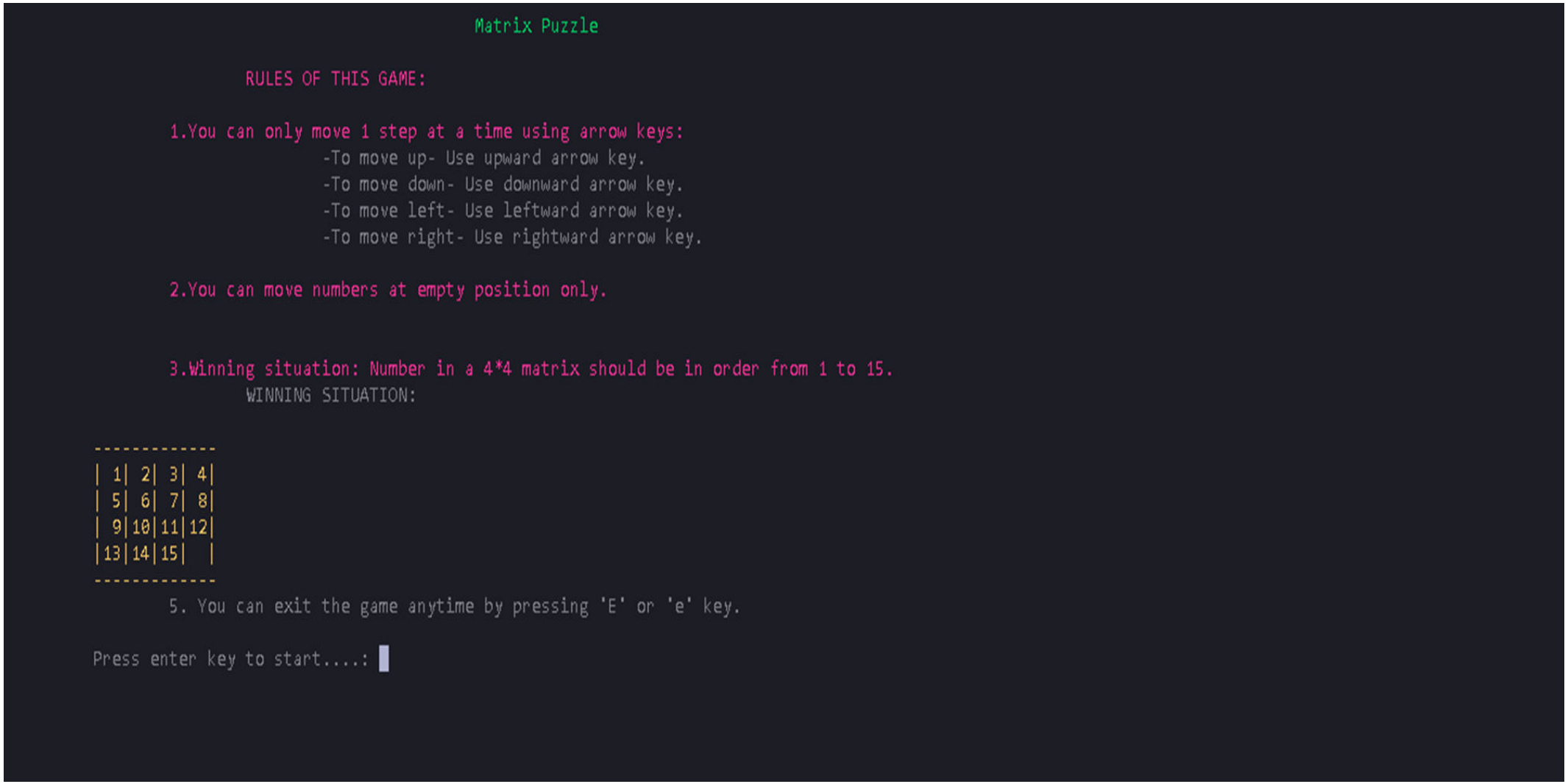
```
212         system("cls");
213 
214         print();
215         printf("\033[1;31m");
216         printf("\nInvalid move");
217         printf("\npress enter key...");
218         getchar();
219     }
220     else
221     {
222         swap(&arr[i][j], &arr[i][j + 1]);
223     }
224     system("cls");
225 }
226 int win_matrix() // matrix should be equal to this matrix inorder to win
227 {
228     int i, j;
229     int win[4][4] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 0};
230     for (i = 0; i < 4; i++)
231     {
232         for (j = 0; j < 4; j++)
233         {
234             if (win[i][j] != arr[i][j])
235                 return 0;
236         }
237     }
238     return 1;
239 }
240 
241 void main()
242 {
243     createarray();
```

```
243     createarray();
244     display();
245     char name[20];
246     printf("\n\n\tEnter player name: ");
247     fflush(stdin);
248     scanf("%[^\n]s", name);
249     printf("%s", name);
250     printf("\033[0;35m");
251     printf("\t\t\n\nALL THE BEST!");
252     printf("\033[0;30m");
253     system("cls");
254     int choice;
255     while (1)
256     {
257         if (win_matrix())
258         {
259             printf("\n\n\n\n\n\t\t\tCONGRATULATIONS\n\n\t\t\tYou won!  press enter key to exit: ");
260             getchar();
261             exit(0);
262         }
263 
264         else
265         {
266             printf("\033[0;33m");
267             printf("\t\tPlayer name: %s\n\n", name);
268             print();
269             fflush(stdin);
270             printf("\033[0;35m");
271             printf("\n\n\tEnter your choice: ");
272             choice = readEnteredKey(); // takes arrow key input from user
273             switch (choice)
274             {
275             case 69:
```

```
269             fflush(stdin);
270             printf("\033[0;35m");
271             printf("\n\n\tEnter your choice: ");
272             choice = readEnteredKey(); // takes arrow key input from user
273             switch (choice)
274             {
275             case 69:
276             case 101:
277                 printf("\033[1;31m");
278                 system("cls");
279                 printf("\n\n\n\n\t\t\t     Thanks for Playing ! \n\a");
280                 printf("\n\n\n\n\n\t\t\t Hit 'Enter' to exit the game \n");
281                 getchar();
282                 exit(0);
283             case 72:
284                 print();
285                 arrowup();
286                 stop = 1;
287                 break;
288             case 80:
289                 print();
290                 arrowdown();
291                 stop = 1;
292                 break;
293             case 75:
294                 print();
295                 arrowleft();
296                 stop = 1;
297                 break;
298             case 77:
299
300                 print();
301                 arrowright();
```

```
302                 stop = 1;
303                 break;
304             default:
305                 print();
306                 printf("\033[1;31m");
307                 printf("\nInvalid move");
308                 printf("\npress enter key...");
309                 break;
310             }
311         }
312     }
313 }
```

OUTPUT -


Matrix Puzzle
RULES OF THIS GAME:
1.You can only move 1 step at a time using arrow keys:
-To move up- Use upward arrow key.
-To move down- Use downward arrow key.
-To move left- Use leftward arrow key.
-To move right- Use rightward arrow key.
2.You can move numbers at empty position only.
3.Winning situation: Number in a 4*4 matrix should be in order from 1 to 15.
WINNING SITUATION:
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	
5. You can exit the game anytime by pressing 'E' or 'e' key.
Press enter key to start....:

```
Enter player name:
```

```
              Player name: Jeevanjot


-------------
|12| 3| 5|11|
|15| 4| 8| 6|
| 2|13| 7|10|
| 9|14| 1|  |
-------------

        Enter your choice:
```

```
               Player name: Jeevanjot


-------------
|12| 3| 5|11|
|15| 4| 8| 6|
| 2|13| 7|10|
| 9|14|  | 1|
-------------

         Enter your choice:
```

-Down arrow key was pressed(which was an invalid move).

-All the elements were arranged sequentially.