



حال با استفاده از یک روش جدول را به صورت زیر پر کنید:

14	13	12	11	10	9	8	7	7	8	9	10	11	12	13	14
13	12	11	10	9	8	7	6	6	7	8	9	10	11	12	13
12	11	10	9	8	7	6	5	5	6	7	8	9	10	11	12
11	10	9	8	7	6	5	4	4	5	6	7	8	9	10	11
10	9	8	7	6	5	4	3	3	4	5	6	7	8	9	10
9	8	7	6	5	4	3	2	2	3	4	5	6	7	8	9
8	7	6	5	4	3	2	1	1	2	3	4	5	6	7	8
7	6	5	4	3	2	1	0	0	1	2	3	4	5	6	7
7	6	5	4	3	2	1	0	0	1	2	3	4	10	6	7
8	7	6	5	4	3	2	1	1	2	3	4	5	6	7	8
9	8	7	6	5	4	3	2	2	3	4	5	6	7	8	9
10	9	8	7	6	5	4	3	3	4	5	6	7	8	9	10
11	10	9	8	7	6	5	4	4	5	6	7	8	9	10	11
12	11	10	9	8	7	6	5	5	6	7	8	9	10	11	12
13	12	11	10	9	8	7	6	6	7	8	9	10	11	12	13
14	13	12	11	10	9	8	7	7	8	9	10	11	12	13	14

این کار را میتوانید به چند روش انجام دهید

روش اول: مقدار دهی مستقیم و تک به تک:

$T_{\text{maze}}[0,0]=14;$

$T_{\text{maze}}[0,1]=13;$

$T_{\text{maze}}[0,1]=12;$

$T_{\text{maze}}[0,1]=11;$

$T_{\text{maze}}[0,1]=10;$

$T_{\text{maze}}[0,1]=9;$

.....

که استفاده از این روش فضای حافظه زیادی را اشغال میکند و استفاده از آن توصیه نمیشود .

روش دوم : مقدار دهی یک جا با استفاده از خاصیت چند بعدی ها :

```
unsigned char Tmaze [16][16]=
{
{14,13,12,11,10,9,8,7,7,8,9,10,11,12,13,14}
,{13,12,11,10,9,8,7,6,6,7,8,9,10,11,12,13}
,{12,11,10,9,8,7,6,5,5,6,7,8,9,10,11,12}
,{11,10,9,8,7,6,5,4,4,5,6,7,8,9,10,11}
,{10,9,8,7,6,5,4,3,3,4,5,6,7,8,9,10}
,{9,8,7,6,5,4,3,2,2,3,4,5,6,7,8,9}
,{8,7,6,5,4,3,2,1,1,2,3,4,5,6,7,8}
,{7,6,5,4,3,2,1,0,0,1,2,3,4,5,6,7}
,{7,6,5,4,3,2,1,0,0,1,2,3,4,5,6,7}
,{8,7,6,5,4,3,2,1,1,2,3,4,5,6,7,8}
,{9,8,7,6,5,4,3,2,2,3,4,5,6,7,8,9}
,{10,9,8,7,6,5,4,3,3,4,5,6,7,8,9,10}
,{11,10,9,8,7,6,5,4,4,5,6,7,8,9,10,11}
,{12,11,10,9,8,7,6,5,5,6,7,8,9,10,11,12}
,{13,12,11,10,9,8,7,6,6,7,8,9,10,11,12,13}
,{14,13,12,11,10,9,8,7,7,8,9,10,11,12,13,14}
};
```

روش سوم : در این روش می توانید در ابتدا این متغیر را تعریف کنید و سپس با استفاده از یک برنامه و الگوریتم به پر کردن آن بپردازید ، یکی از بهترین روشهاست .

```
char Tmaze[16,16];
char info[16,16]; // information
unsigned int value=0x0E; // همون عدد 14 است
char xtp,ytp;
for (xtp=0;xtp<16;xtp++){
for (ytp=0;ytp<16;ytp++){
Tmaze[xtp,ytp]=value;
if (ytp<=7) value--;
if (ytp>=7) value++;
}
value--;
if (xtp<=7) value--;
if (xtp>=7) value++;
}
}
```

بعد از مقدار دهی جدول Tmaze باید تمام خانه های جدول Info\_maze را که مربوط به اطلاعا هر خانه است را صفر کنید .

اطلاعاتی که در جدول دوم قرار میگیرد به از این قرار است

- 1- وجود دیواره جلو
- 2- وجود دیواره عقب
- 3- وجود دیواره سمت راست
- 4- وجود دیواره سمت چپ
- 5- خانه ای که یک بار رفته شده
- 6- خالی
- 7- خالی
- 8- خالی

با توجه به اینکه جدول دوم از 256 خانه 8 بیتی تشکیل شده و هر 8 بیت مربوط به یک خانه است میتوانیم برای هر خانه تا 8 بیت اطلاعات ذخیره کنیم مثل بالا

حال اگر به عنوان مثال خانه  $x,y$  را در نظر بگیریم با قراردادن آنها در  $Tmaze[x,y]$  عدد داخل خانه را بدست می آوریم و با قرار دادن آنها در  $Info\_maze[x,y]$  اطلاعات مربوط به هر خانه را که مثلا این خانه را قبلا آمده ایم یا نه را بدست می آورید .

قوانین در حل ماز به این روش :

1- دور تا دور زمین دیواره کشیده شده است ، یعنی سمت پایین خانه هایی که در سطر  $y=0$  قرار دارند دارای دیواره

هستن و همچنین سمت بالای خانه هایی که در سطر  $y=15$ ، سمت چپ ستون  $x=0$  و سمت راست  $x=15$  دارای دیواره هستند که بهتر است قبل از شروع به کار روبات مقدار دهی اولیه شوند یعنی در جدول دوم در قسمت مربوط به دیواره های این خانه ها عدد یک که نشان دهنده وجود دیواره است را یادداشت کنید .

2- وارد هر خانه ای که میشود در 8 بیتی مربوط به آن خانه در جدول دوم ، بیتی که مربوط به رفته شدن یا نشدن است را به نشانه رفته شدن این خانه که همکنون در آن هستید یک کنید .

3- بعد از توقف در هر خانه با استفاده از سنسورها دیواره ها را بخوانید (وجود یا عدم وجود دیواره) و اگر دیواره موجود بود در جدول دوم علامت بزنید .

4- باید همیشه یک متغیر داشته باشید که سمت روبات را نشان دهد که میتواند از جهت های بالا، پایین ، راست ، چپ یا شمال ، جنوب ، شرق ، غرب استفاده کنید

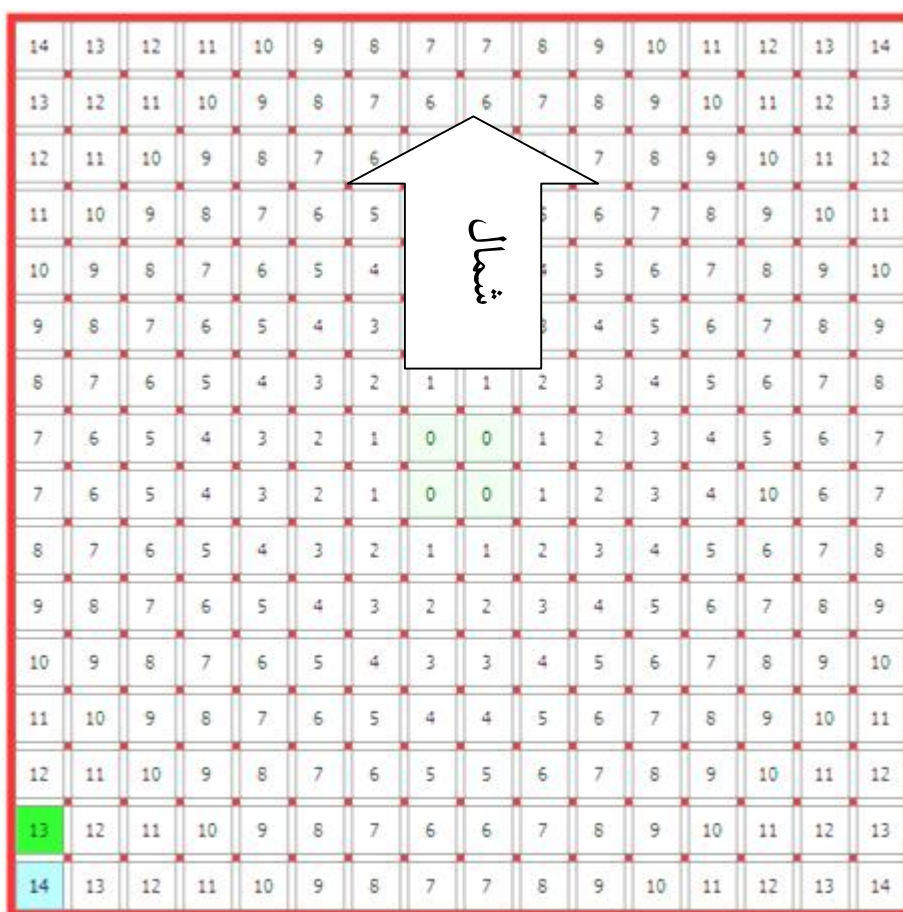
5- همیشه باید وارد خانه ای شوید که اولاً آنرا نرفته باشید و دوّمَا عدد آن کمتر از خانه ای باشد که در آن هستید و در نهایت اگر خانه مقصدتان در شرایط بالا قرار نگرفت یعنی دو خانه بود که هم نرفته بودید و هم عددشان باهم برابر باشد و هم کوچکتر از خانه ایست که در آن هستیم تکلیف چیست ؟

6- در شرایط قسمت قبل باید ورود به خانه را تحت یک اولیوت انجام دهیم مثلا :

اول بالا - دوم راست - سوم چپ - چهارم عقب

مثال اگر در ابتدا شروع به حرکت میکنید روی روبات به سمت شمال است و چرخش به سمت راست روی روبات

را به سمت شرق میکند



7- اگر عدد خانه ای که می خواهید وارد آن شوید از عدد خانه ای که در آن هستید بیشتر است باید به تابع بروز رسانی مراجعه کنید (این تابع بعدا توضیح داده میشود)