对迷宫程序作简单介绍：

1. 首先是初始化。创建存储迷宫的二维数组。createMap函数，通过键盘输入行、列开辟二维内存空间，其中对行列加2保险作用。
2. 初始化迷宫。例如输入5\*5的迷宫，那么对0行0列、6行6列的数组元素都赋值0，对中间都赋值1，中间1就用方块填充。

值得注意的是SetConsoleCursorPosition设置控制台光标这个函数，其横纵坐标表示距离原点纵向和横向的字符个数多一个且对应的横纵坐标相调。使用的是unicode字符，占2个字符宽度。使用单字节字符即可解决，也可以将列坐标乘2。例如你想在10行30列输出unicode字符，那么应令coord.x=2\*(30-1), coord.y=10-1;

1. **画出迷宫路径create1函数。**这是重难点。先在里面5\*5的区域随机选定一个位置，不能随机到边上。2 \* (rand() % (Height / 2) + 1)，其中height是int型变量。此式范围为2~height-1，即随机点不会选到边上，令随机点空格（表示打通）。然后采用一个4\*2的二维数组存储上下左右四个方向，**对得到的随机点（x,y）的下一方向的再下一方向进行判定是否为5\*5范围内且未打通。**若是，对（x,y）的下一方向（x1,y1）再打通，对（x,y）的下一方向的再下一方向即（x2,y2）继续执行create1函数；若不是，对（x,y）换一个方向继续进行打通。详情结合代码看。
2. 闯迷宫开始。迷宫初始化好之后，通过读取键盘方向键，控制\*号在空白路径处的打印，其中坐标点x/y都是不断变化的。直接绘制离开路径时，就从此时的x/y开始。
3. **直接输出路径函数mazePath()。**该函数先创建一个二维的bool表mark[][]，其中每个元素对应着迷宫Map[][]的元素，**当Map[][]中某一点是墙1，那么mark[][]中该点为true；Map[][]中为通道空格0，则mark[][]中为0.**再调用getAdjacentNotVisitedNode()函数遍历相应的为true的点，存到pointStack栈中。但是没找到终点endP遍历是不会结束的，故当遍历到四周都无路可走的路径点时，又依次弹栈，到上一个不同方向继续遍历。直到到达终点。最后打印出路径。