```python
import random

def make_maze(yoko, tate):
    XP = [ 0, 1, 0, -1]
    YP = [-1, 0, 1,  0]

    maze_lst = []
    for y in range(tate):
        maze_lst.append([0]*yoko)
    for x in range(yoko):
        maze_lst[0][x] = 1
        maze_lst[tate-1][x] = 1
    for y in range(1, tate-1):
        maze_lst[y][0] = 1
        maze_lst[y][yoko-1] = 1
    for y in range(2, tate-2, 2):
        for x in range(2, yoko-2, 2):
            maze_lst[y][x] = 1
    for y in range(2, tate-2, 2):
        for x in range(2, yoko-2, 2):
            if x > 2: rnd = random.randint(0, 2)
            else:     rnd = random.randint(0, 3)
            maze_lst[y+YP[rnd]][x+XP[rnd]] = 1

    return maze_lst

def sarch(maze_bg,canvas):
    global mx,my,px,py
    lx=len(maze_bg[0])
    ly=len(maze_bg)

    k=[]
    p=0
```

```
33      for y in range(1,ly-1):
34          for x in range(1,lx-1):
35              if maze_bg[y][x]==0 and ((maze_bg[y-1][x]==1 and maze_bg[y+1][x]==1 and maze_bg[y]
36              (maze_bg[y-1][x]==1 and maze_bg[y+1][x]==1 and maze_bg[y][x+1]==1) or
37              (maze_bg[y-1][x]==1 and maze_bg[y][x-1]==1 and maze_bg[y][x-1]==1) or
38              (maze_bg[y+1][x]==1 and maze_bg[y][x+1]==1 and maze_bg[y][x-1]==1)):
39                  k.append([x,y])
40      for j in k:
41          m=(j[0]-1)**2+(j[1]-1)**2
42          if p<m:
43              p=m
44              px=j[0]
45              py=j[1]
46      canvas.create_rectangle(px*50, py*50, px*50+50, py*50+50,
47                                      fill="red")
48      return px,py
49
50  def show_maze(canvas, maze_lst):
51      color = ["white", "gray","red"]
52      for y in range(len(maze_lst)):
53          for x in range(len(maze_lst[y])):
54              canvas.create_rectangle(x*50, y*50, x*50+50, y*50+50,
55                                      fill=color[maze_lst[y][x]])
56      sarch(maze_lst,canvas)
57
58
```