4 In a computer game, a player ("O") moves around a maze measuring 10 metres by 11 metres to collect a prize ("P"). The prize is placed at a random position within the maze. The prize position is not where a wall ("X") appears in the maze. An empty position is indicated with a full-stop (".").

The maze is represented on the screen by a rectangular grid. Each square metre of the maze is represented by an x-coordinate and a y-coordinate. The top left square metre of the puzzle display has x = 0 and y = 0.

The player moves left, right, up or down according to a direction entered by the user. The game is turn-based; a user enters the direction, their player moves one position in that direction. If the direction would place the player on a wall, then the player does not move. The maze is displayed after each move.

```
X X X X X X X X X X
X . . . . . . . . X
X . X . X . X X . X
X . X . . P . . . X
X . X X X X X X . X
X . . . O . . . . X
X . X . X X . X . X
X . X . . . . X . X
X . X X . X X X . X
X . . . . . . . . X
X X X X X X X X X X
```

Task 4.1
Write a program to display the maze as shown.

- The maze should be stored in a suitable data structure.
- The data structure will allow fixed loop(s) to be used to display the maze.

The maze is given in the text file MAZE.TXT. You may read in the data from this file **or** place the data in your program using any suitable method.

Evidence 14
Your program code. [6]

Task 4.2
The prize is placed randomly on the maze. It cannot appear in the same grid position as a wall ("X").

Add to your program code to place the prize at a random position.

Take a screenshot of the maze with the prize displayed in it.

Evidence 15
Your program code. [4]

Evidence 16
A screenshot of the maze as output by your program. [1]

The player is represented by the character "O". The character starts the game in a central position on the grid, for example, x = 4 and y = 5.

To move the character, the user is prompted for a direction. The following are valid inputs:

Input character	Action
"U"	Player moves up
"D"	Player moves down
"L"	Player moves left
"R"	Player moves right
""	Continue with previous move. If no previous move, do nothing

If the next position for the player ("O") is a wall ("X"), then the player stays in their current position; this is called collision detection.

When the player enters the move, a new position for the player ("O") is calculated and the maze is displayed. The previous position is changed back to a "." when the player has a new position. The moves are repeated until the player is at the same position as the prize.

Task 4.3
Add to your program code to:

- place the player on the grid at a central position on the grid
- take in and validate a direction
- calculate a new position
- check this position is not a wall
- update the grid so that the previous position of "O" is replaced with a "." and "O" is located in its new position
- continue this until the player is at the same position as the prize.

Evidence 17
Your program code. [16]

When the player and the prize are at the same position, the message "Player has reached the prize" is displayed and the game ends.

Task 4.4
Add to your program, code to end the game when this condition is met, and display the required message. Produce screenshots to show key elements of your program are functioning.

The screenshots required are:

- entering each direction
- player changing position
- end of game

Evidence 18
Your program code. [1]

Evidence 19
Screenshots of:

- entering each direction
- player changing position
- end of game (player wins)

[2]