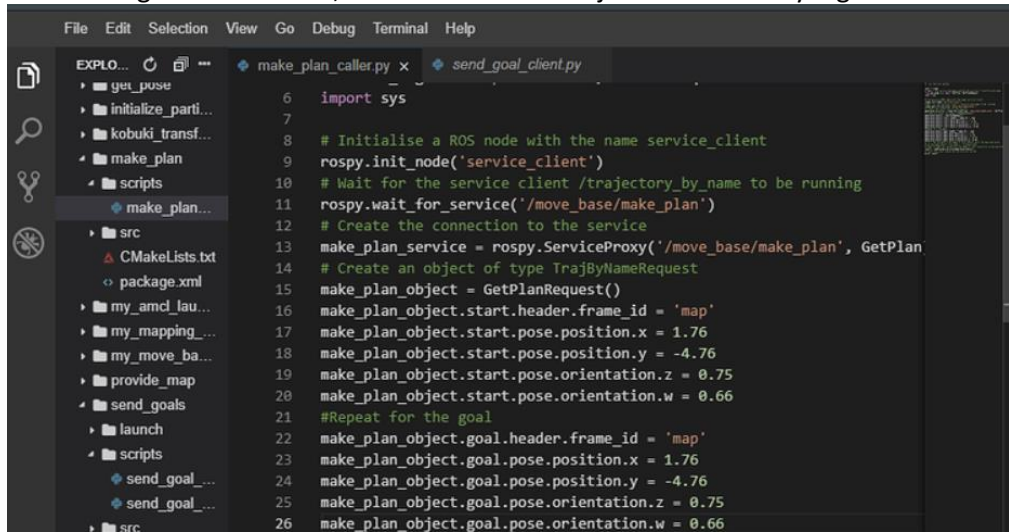


Sample expected output for question 1 (Simulation of Hangman Game)

The correct word in this example is **programming** (1 marks)

Hint in image format below, in text format “what jobs are currently high in demand”



The image shows a screenshot of a ROS2 development environment, likely Visual Studio Code. The interface includes a menu bar (File, Edit, Selection, View, Go, Debug, Terminal, Help), a sidebar with a file explorer, and a main editor window. The file explorer on the left shows a project structure with folders like 'src', 'scripts', and 'launch', and files like 'package.xml' and 'CMakeLists.txt'. The main editor window displays a Python script named 'make_plan_caller.py'. The script is a ROS2 service client that initializes a node, waits for a service, and then sends a goal to a 'make_plan' service. The goal is defined with specific pose parameters: position (x=1.76, y=-4.76), orientation (z=0.75, w=0.66), and a frame_id of 'map'.

```
6 import sys
7
8 # Initialise a ROS node with the name service_client
9 rospy.init_node('service_client')
10 # Wait for the service client /trajectory_by_name to be running
11 rospy.wait_for_service('/move_base/make_plan')
12 # Create the connection to the service
13 make_plan_service = rospy.ServiceProxy('/move_base/make_plan', GetPlan
14 # Create an object of type TrajByNameRequest
15 make_plan_object = GetPlanRequest()
16 make_plan_object.start.header.frame_id = 'map'
17 make_plan_object.start.pose.position.x = 1.76
18 make_plan_object.start.pose.position.y = -4.76
19 make_plan_object.start.pose.orientation.z = 0.75
20 make_plan_object.start.pose.orientation.w = 0.66
21 #Repeat for the goal
22 make_plan_object.goal.header.frame_id = 'map'
23 make_plan_object.goal.pose.position.x = 1.76
24 make_plan_object.goal.pose.position.y = -4.76
25 make_plan_object.goal.pose.orientation.z = 0.75
26 make_plan_object.goal.pose.orientation.w = 0.66
```

```
-----
10 incorrect guesses left.
Guess: a
Correct!
-----a-----
10 incorrect guesses left.
Guess: e
Incorrect!
-----a-----
9 incorrect guesses left.
Guess: i
Correct!
-----a--i--
9 incorrect guesses left.
Guess: o
Correct!
--o--a--i--
9 incorrect guesses left.
Guess: u
Incorrect!
--o--a--i--
8 incorrect guesses left.
Guess: k
Incorrect!
--o--a--i--
7 incorrect guesses left.
Guess: p
Correct!
p-o--a--i--
7 incorrect guesses left.
Guess: r
Correct!
pro-ra--i--
7 incorrect guesses left.
Guess: a
Correct!
pro-ra--i--
7 incorrect guesses left.
Guess: █
```