

Exercise - Prompt Copilot to generate code

10 minutes

Sometimes Copilot might not suggest the exact code you need. You can use more specific prompts with Copilot to generate the results you need. In this exercise, you'll use Copilot to generate new methods in the Trie project. Let's get started!

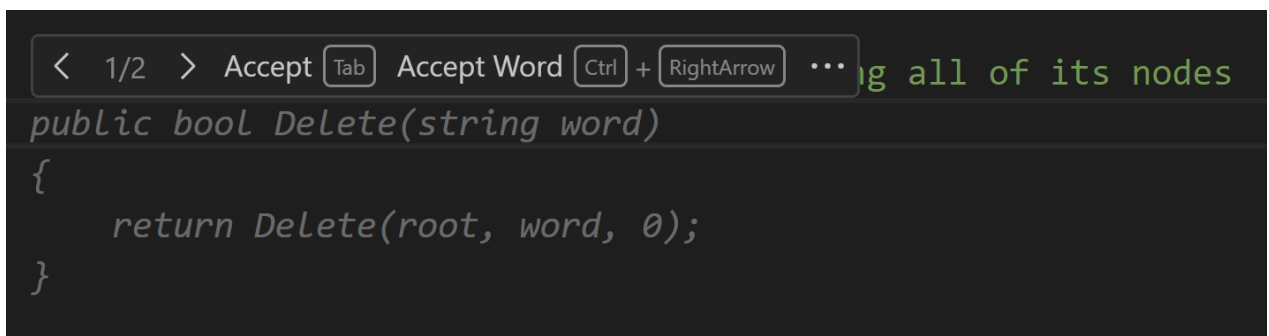
1. Enter a new line after the `GetAllWordsWithPrefix` method, then enter the following code:

c#

```
// Delete a word from the trie  
public bool Delete(string word)  
{
```

2. Wait for Copilot to generate code, then press `Tab` or click **Accept** to apply the code suggestion.

You might notice that Copilot has multiple suggestions. You can click the arrow to navigate through suggestions.



3. Take a moment to consider the code that Copilot autocompleted.

Feel free to prompt Copilot to generate comments to explain the code by entering a new line with forward slashes.

You might have a method similar to the following code:

c#

```
// Delete a word from the trie  
public bool Delete(string word)
```

```
{
    TrieNode current = root;
    foreach (char c in word)
    {
        if (!current.HasChild(c))
        {
            // Word doesn't exist in trie
            return false;
        }
        current = current.Children[c];
    }
    if (!current.IsEndOfWord)
    {
        // Word doesn't exist in trie
        return false;
    }
    // Word exists in trie
    // Set IsEndOfWord to false
    current.IsEndOfWord = false;
    return true;
}
```

Notice that this code sets the value of `IsEndOfWord` to false, but it doesn't remove the nodes of the word from the trie. To delete these nodes, you need to prompt Copilot to be more specific.

Prompt Copilot to generate specific code

1. Delete the previously generated `Delete` method.
2. On a new line, type the following comment:

c#

```
// Helper method to delete a word from the trie by recursively removing its nodes
```

3. Enter a new line and wait for Copilot to suggest code.

If you're having trouble getting Copilot to autocomplete the method, you can add the following parameters and wait for Copilot to complete the method code:

c#

```
private bool _delete(TrieNode current, string word, int index)
{
```

4. Take a moment to consider the code that Copilot suggested.

You might want to prompt Copilot to generate comments to explain the code by entering a new line with forward slashes.

The code generated should perform the following tasks:

- Determine if the word to be deleted exists in the trie.
- Update `IsEndOfWord` of the last node in the word.
- Verify that the nodes not part of another word before deleting them.
- Return true if the word was deleted, otherwise return false.

c#

```
private bool _delete(TrieNode root, string word, int index)
{
    // Base case: we've reached the end of the word
    if (index == word.Length)
    {
        // If the current node is not the end of a word, we can't
delete it
        if (!root.IsEndOfWord)
        {
            return false;
        }
        // Otherwise, we can delete it by setting the end of word flag
to false
        root.IsEndOfWord = false;
        // And returning true to indicate that the node can be deleted
        return true;
    }
    // Recursive case: we haven't reached the end of the word yet
    char currentChar = word[index];
    // If the current node doesn't have a child with the current char-
acter, we can't delete it
    if (!root.HasChild(currentChar))
    {
        return false;
    }
    // Otherwise, we can delete it if its child can be deleted
    bool canDelete = _delete(root.Children[currentChar], word, index +
1);
    // If the child can be deleted and the current node is not the end
of a word, we can delete it
    if (canDelete && !root.IsEndOfWord)
    {
        root.Children.Remove(currentChar);
        return true;
    }
    // Otherwise, we can't delete it
    return false;
}
```

5. Press `Tab` or click **Accept** to apply the code suggestion.
6. On a new line, enter `public bool Delete` and wait for Copilot to suggest code.

Copilot should suggest code that calls the `_delete` method.
7. Press `Tab` or click **Accept** to apply the code suggestion.

Check your work

In this task, you'll test the methods you created with Copilot to verify that they work as expected.

1. Open the **Program.cs** file.
2. Comment any method calls and uncomment the `DeleteWord` method call:

```
c#  
  
Trie dictionary = InitializeTrie(words);  
// SearchWord();  
// PrefixAutocomplete();  
DeleteWord();  
// GetSpellingSuggestions();
```

3. Navigate to the `DeleteWord()` method and uncomment the following code:

```
c#  
  
/*  
if (input != null && dictionary.Search(input))  
{  
    dictionary.Delete(input);  
    Console.WriteLine($"Deleted \"{input}\" from dictionary\n");  
    PrintTrie(dictionary);  
}  
*/
```

4. In the file explorer, right click the **Program.cs** file and click **Open in Integrated Terminal**
5. Enter `dotnet run` to run the program.
6. Enter a word to delete such as "follows" or "beans"
7. Verify that your output is similar to the following

Output

```
The dictionary contains the following words:  
as, astronaut, asteroid, are, around, cat, cars, cares, careful, care-  
fully, for, forgot, follows, from, front, mellow, mean, money, monday,  
monster, place, plan, planet, planets, plans, the, their, they, there,  
towards,
```

```
Enter a word to delete, or press Enter to exit.
```

```
follows
```

```
Deleted "follows" from dictionary
```

```
The dictionary contains the following words:  
as, astronaut, asteroid, are, around, cat, cares, careful, carefully,  
for, forgot, follows, from, front, mellow, mean, money, monday, mon-  
ster, place, plan, planet, planets, plans, the, their, they, there,  
towards,
```

```
Enter a word to delete, or press Enter to exit.
```

```
beans
```

```
Did not find "beans" in dictionary
```

```
Enter a word to delete, or press Enter to exit.
```

If your code displays different results, review your code to find your error and make updates. Run the code again to see if you've fixed the problem. Continue updating and running your code until your code produces the expected results.

8. Comment out the Delete method call.