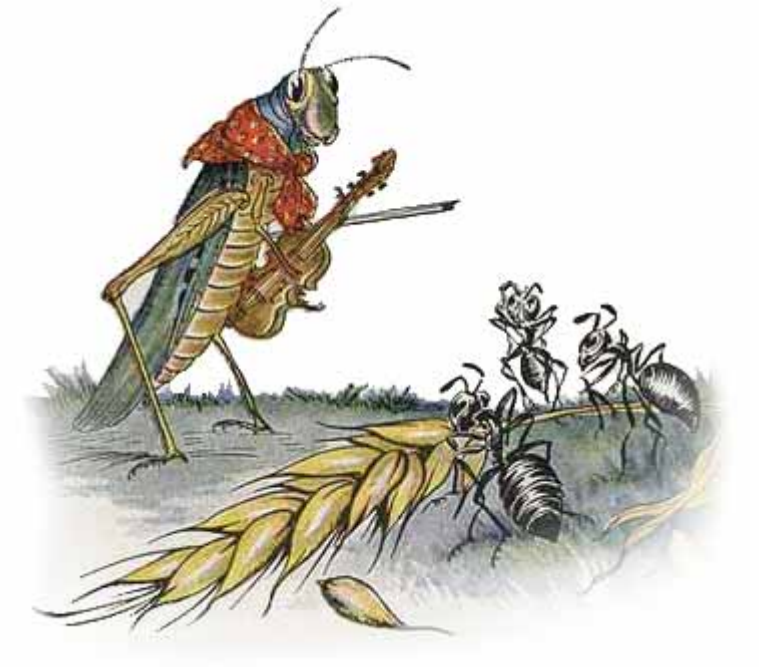
THE UNIVERSITY OF SYDNEY

INFO1110 / COMP9001 Assignment 1

The Ant and the Grasshopper

Due: 31 Aug 2018, 6pm AEST (Week 5, Friday)

Weighting: 0% of the final assessment mark



> One bright day in late autumn a family of Ants were bustling about in the warm sunshine, drying out the grain they had stored up during the summer, when a starving Grasshopper, his fiddle under his arm, came up and humbly begged for a bite to eat.
>
> "What!" cried the Ants in surprise, "haven't you stored anything away for the winter? What in the world were you doing all last summer?"
>
> "I didn't have time to store up any food," whined the Grasshopper; "I was so busy making music that before I knew it the summer was gone."
>
> The Ants shrugged their shoulders in disgust.
>
> "Making music, were you?" they cried. "Very well; now dance!" And they turned their backs on the Grasshopper and went on with their work.

Credits: Library of Congress, adapted from the book *The Aesop for Children: with Pictures by Milo Winter.*

Overview

Description

Unwilling to live out the harsh, cold winter without any food, the Grasshopper decided it would steal the grain from the Ants' nest, to keep for itself.

For this assignment, you will write a simple game *The Ant and the Grasshopper*. The player will control the grasshopper as it walks to the nest, hops over ants, picks up the food inside the nest, and drops it at home.

Implementation details

Your game will be written in Python 3. No packages or modules are to be imported.

Help and feedback

You are encouraged to ask questions about the assignment on the discussion board, on ed.

During your tutorial in Week 5, you also have the opportunity to ask your tutor to review your code. If you choose to get this feedback, you will need to present your code on *paper*. Your code will be printed onto two double-sided pages, using a monospaced font of size 12.

Please ensure your code is comprehensible before requesting feedback. We recommend that your code adheres to the PEP 8 style guide, and is commented appropriately.

After the assignment deadline, your tutor will review your last submission outside the tutorial. They will provide feedback on ed, which will be visible only to you.

Submission

You will submit your code on the assignment page, on ed. You are encouraged to submit multiple times. After each submission, the marking system will automatically check your code against public test cases.

These public tests do not cover all parts of the specification and your code. The complete test suite contains both public and hidden test cases, and your code will not be run through this suite until after the assignment deadline.

Please ensure you carefully follow the assignment specification. Your program output must exactly match the output shown in the examples.

Warning: Any attempts to deceive or disrupt the marking system will result in an immediate zero for the entire assignment. Negative marks can be assigned if you do not properly follow the assignment specifications, or your code is unnecessarily or deliberately obfuscated.

Gameplay

When the game starts, it shows the following "picture" and prompts the player to enter a command.

```
$ python grasshopper.py
[] G . . . a . . a . a . [***]
>
```

The picture, explained

Looking at the picture from left to right:

- `[]` the Grasshopper's home. It is empty at first, because the Grasshopper doesn't have any food.
- `G` the Grasshopper, which is controlled by the player.
- `.` an empty space. The Grasshopper can walk or hop onto empty spaces.
- `a` an Ant. All ants are stationary. The Grasshopper should hop over Ants.
- `[***]` the Ants' nest. Each `*` is a bit of food, which the Grasshopper can pick up.

Commands

The player can instruct the grasshopper to perform each of these actions. After each action is performed, the game updates the picture and shows it to the user.

All commands are case-insensitive. If an invalid command is entered, the game prints `Invalid command` and prompts the player to enter another command.

```
$ python grasshopper.py
[] G . . . a . . a . a . [***]
> spam
Invalid command
>
```

`WALK L`

The Grasshopper moves one space to the left.

```
[] . . G . a . . a . a . [***]
> WALK L
[] . G . . a . . a . a . [***]
>
```

If the Grasshopper is already next to its home, this command has no effect.

```
[] G . . . a . . a . a . [***]
> WALK L
[] G . . . a . . a . a . [***]
>
```

If the Grasshopper moves onto an Ant, the Grasshopper will be attacked and the game will end.

```
[] . . . . a G . a . a . [***]
> WALK L
Arrgh!! An Ant attacked you. Game over.
```

`WALK R`

The same as `WALK L`, but the Grasshopper moves to the right instead. Like before, this command has no effect if the Grasshopper is already next to the Ants' nest.

`HOP L`

The Grasshopper moves to the left, jumping over one spot and landing on the next.

```
[] . . G . a . . a . a . [***]
> HOP L
[] G . . . a . . a . a . [***]
>
```

```
[] . . . . a G . a . a . [***]
> HOP L
[] . . . G a . . a . a . [***]
>
```

Like `WALK L`, this command has no effect if the Grasshopper is already next to its home.

If hopping left results in the Grasshopper landing on top of its home, the `HOP L` command will instead produce the same effect as `WALK L`.

```
[] . G . . a . . a . a . [***]
> HOP L
[] G . . . a . . a . a . [***]
>
```

If the Grasshopper hops onto an Ant, the Grasshopper will be attacked and the game will end.

```
[] . . . . a . G a . a . [***]
> HOP L
Arrgh!! An Ant attacked you. Game over.
```

`HOP R`

The same as `HOP L`, but the Grasshopper moves to the right instead. Like before, the Grasshopper cannot hop over or onto the Ants' nest.

`PICK UP`

The Grasshopper picks up a bit of food from the Ants' nest, which it can carry back to its own home.

```
[] . . . . a . . a . a G [***]
> PICK UP
[] . . . . a . . a . a G* [**]
>
```

```
[*] . . . . a . . a . a G [**]
> PICK UP
[*] . . . . a . . a . a G* [*]
>
```

```
[**] . . . . a . . a . a G [*]
> PICK UP
[**] . . . . a . . a . a G* []
>
```

This command is valid only when:

- the Grasshopper is next to the Ants' nest,
- the Ants' nest contains food, and
- the Grasshopper isn't already holding food.

If these conditions aren't true, the game prints `Cannot pick up food` and prompts the player to enter another command.

```
[] G . . . a . . a . a . [***]
> PICK UP
Cannot pick up food
>
```

`DROP`

The Grasshopper drops the bit of food it was carrying.

```
[] G* . . . a . . a . a . [**]
> PICK UP
[*] G . . . a . . a . a . [**]
>
```

```
[*] G* . . . a . . a . a . [*]
> PICK UP
[**] G . . . a . . a . a . [*]
>
```

This command is valid only when:

- the Grasshopper is next to its own home, and
- the Grasshopper is already holding food.

If these conditions aren't true, the game prints `Cannot drop food` and prompts the player to enter another command.

```
[] G . . . a . . a . a . [***]
> DROP
Cannot drop food
>
```

When the last bit of food is dropped, the Grasshopper will have succeeded and the game will end.

```
[**] G* . . . a . . a . a . [**]
> DROP
[***] G . . . a . . a . a . [**]
Congratulations Grasshopper, you now have enough food to last the winter!
```

`QUIT`

At any point, the player may choose to end the game.

```
[] . . . . a . . a . a G* [**]
> QUIT
Goodbye Grasshopper!
```

Academic declaration

By submitting this assignment, you declare the following:

I declare that I have read and understood the University of Sydney Student Plagiarism: Coursework Policy and Procedure, and except where specifically acknowledged, the work contained in this assignment/project is my own work, and has not been copied from other sources or been previously submitted for award or assessment.

I understand that failure to comply with the Student Plagiarism: Coursework Policy and Procedure can lead to severe penalties as outlined under Chapter 8 of the University of Sydney By-Law 1999 (as amended). These penalties may be imposed in cases where any significant portion of my submitted work has been copied without proper acknowledgment from other sources, including published works, the Internet, existing programs, the work of other students, or work previously submitted for other awards or assessments.

I realise that I may be asked to identify those portions of the work contributed by me and required to demonstrate my knowledge of the relevant material by answering oral questions or by undertaking supplementary work, either written or in the laboratory, in order to arrive at the final assessment mark.

I acknowledge that the School of Information Technologies, in assessing this assignment, may reproduce it entirely, may provide a copy to another member of faculty, and/or communicate a copy of this assignment to a plagiarism checking service or in-house computer program, and that a copy of the assignment may be maintained by the service or the School of IT for the purpose of future plagiarism checking.