Assignment 3

Key Information

Academic Integrity

Your code will be checked against all other students with a similarity checker.

Anything you are able to google can be easily found by the teaching team and compared against your work.

In this assessment, you must not use generative artificial intelligence (genAI) to generate any materials or content in relation to the assessment task. This includes ChatGPT and GitHub Copilot.

You are responsible for all code submitted as a part of your assignment.

How can I avoid academic integrity issues?

- Never copy code from anywhere. If you learn something useful online, rewrite it from scratch. It's also the best way to make sure you have understood it. This includes not submitting code you have submitted in a prior assessment in this or another unit. Self-plagiarism is also a breach of academic integrity.
- If a fellow student asks you for a solution to a question, you can try to help them build their solution, but **do not give them yours**. Giving your solution is just as much of an academic Integrity breach as receiving it.
- Don't leave your devices unlocked and unattended. Properly securing your work is your responsibility.

Tips on working in teams

Attend your applied class

Your teammate(s) is in the same applied class. Make sure to attend it to plan your efforts and work together.

Working together

Pair programming

You and your group member/s can sit together and work on a single computer trying to solve each task. This has the benefit of allowing everyone to discuss ideas to solve each part of the problem, as well as double-checking your work as you go along. You'll learn a lot by doing this, and your program will be better.

You can also achieve the same outcome by pair programming online via Zoom or any other collaborative platform.

Work separately

There may be times when you need to work asynchronously. In these cases, you may wish to work directly on your own copy of the code, or you can work in a collaborative way using an Ed workspace to be shared with your group. These allow each member of the team to work on a single copy of the code online (similarly to editing a Google Doc), that way all changes are shared live.

If all teammates are familiar with other tools like git, you are free to use them (however, we will not be able to provide any support if you do).

In any case, all of you will need to copy your code from the ed workspace to the A2 ed lesson to run the automated tests in order to obtain the checkpoint marks.

Resolving issues

If you run into issues (e.g. disagreement, missing teammate), try to resolve them ASAP with your teammate(s).

If that does not work, ask your TA for assistance. Do this as soon as there are issues that you cannot resolve internally.

Tackle team issues as early as they arise. Every semester, we have teams that have unresolved issues by submission day. This generally affects their mark. If you have an issue, do not wait. Ask your TA and document your efforts to resolve the issue, as we will need to know what happened if our

arbitrage is needed.

Splitting groups

The assignment is designed to be completed by 2 people, thus, it generally is not an option to complete the assignment on your own. However, in some cases where internal group conflicts cannot be resolved, it may be prudent to split the group so that each individual can continue on with their assignment on their own.

For a group split to occur, you must have collected evidence to show why your group should be split, and you will be required to first raise these issues with your TA. After this, if a resolution cannot be found, you should make a post on Ed directed to the admin team, and they will be able to assist you further.

Valid reasons for splitting a group includes:

- Your teammate leaving the unit, or otherwise not continuing to study,
- They are not communicating with you at all,
- There is any unacceptable behaviour, including bullying or harassment.

Please collect evidence of the above so that it can be shared with the admin team should they require it.

Group splits must occur at least 1 week prior to the deadline. Any requests after this point will be rejected except in extreme circumstances.

Tips on working with automated tests

Handling inputs

You can assume that the inputs provided by the tests conform to the specifications in the brief. This means that you won't need to check that the inputs are correct, whether they are provided by the user or by the automated tests. If you really think there is an error in the tests, then please report it on ed.

The automated tests use complex inputs that may make it hard to debug. We recommend you run your programs with simple inputs that you can manually check.

Make sure that the program you are writing actually expects input or not. If you use the `input()` function when unexpected, you will receive a cryptic `EOF error` from the automarker. If you see this, double check all of your inputs are correct.

Formatting your outputs

In your program, ensure that the format of the output is the same as the format of the examples provided. Every character counts, including spaces and new lines.

Printing new lines

The last character is the newline character `'\n'` in every output file in the automated tests.

```
print("This is on line 1...\nThis is on line two!")
```

The code above produces the output below.

```
This is on line 1...
This is on line two!

```

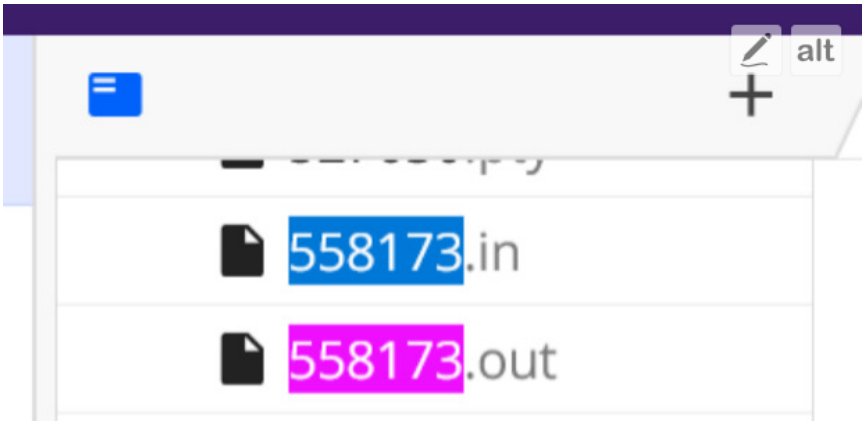
Notice the that last line is empty. This is because the penultimate line is a `print`, which by default ends with a newline character `'\n'`.

Debugging failed tests

If your program does not pass all the tests, look up the ID of a test that fails:

558173

Then you can look for the test in your scaffold (including with ctrl+F):



There are two files for each test under the `public` directory of your scaffold:

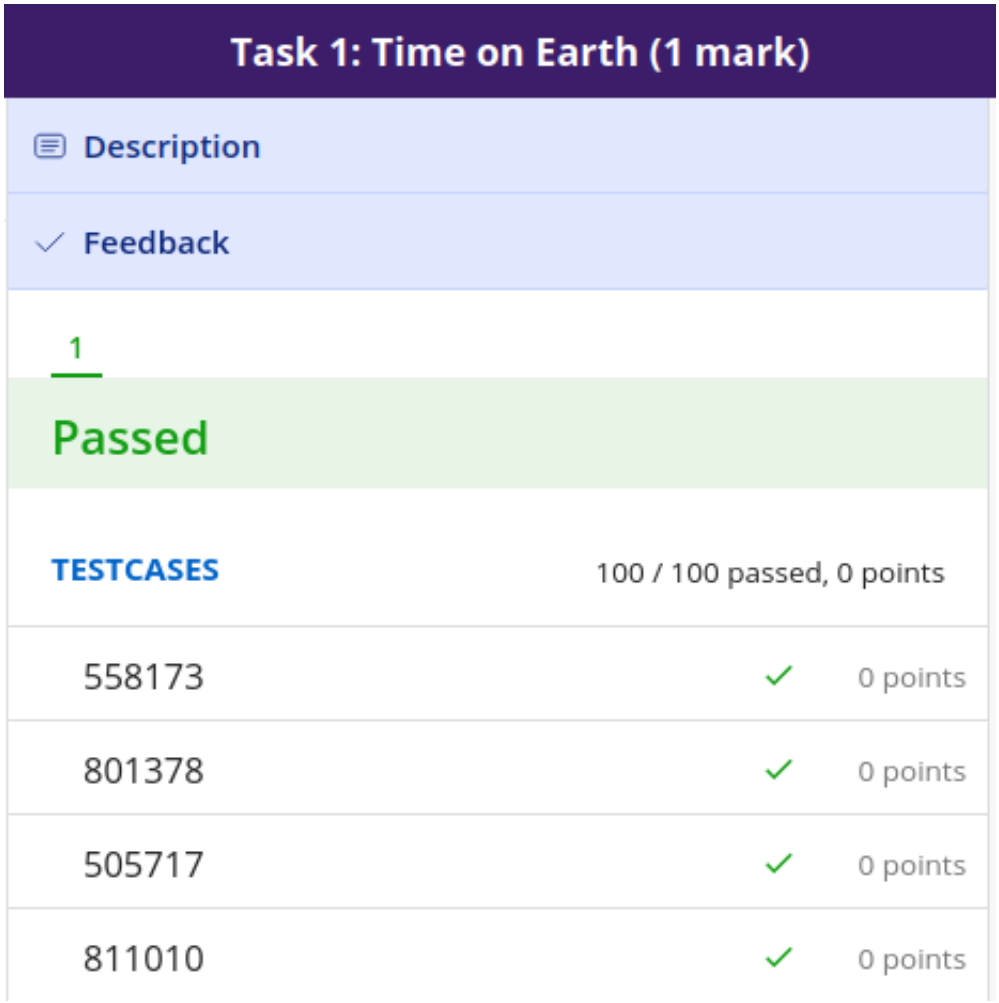
- `.in` files show the input that is fed to your program.
- `.out` files show the output expected. Beware that some newline characters do not appear because they are in the `.in` file.

Using these files, you can:
- Run the command `python3 name_of_your_program.py < public/558173.in > my_output.out` in the terminal and press Enter to write the output of your program to the file `my_output.out`.
- Run the command `diff my_output.out public/558173.out` in the terminal and press Enter to compare the output of your program to the correct output.

You do not have to do this to complete the assignment, but it may help you on your debugging journey.

Understanding passing test cases

Below is an example of code that passes all 100 **public** tests:



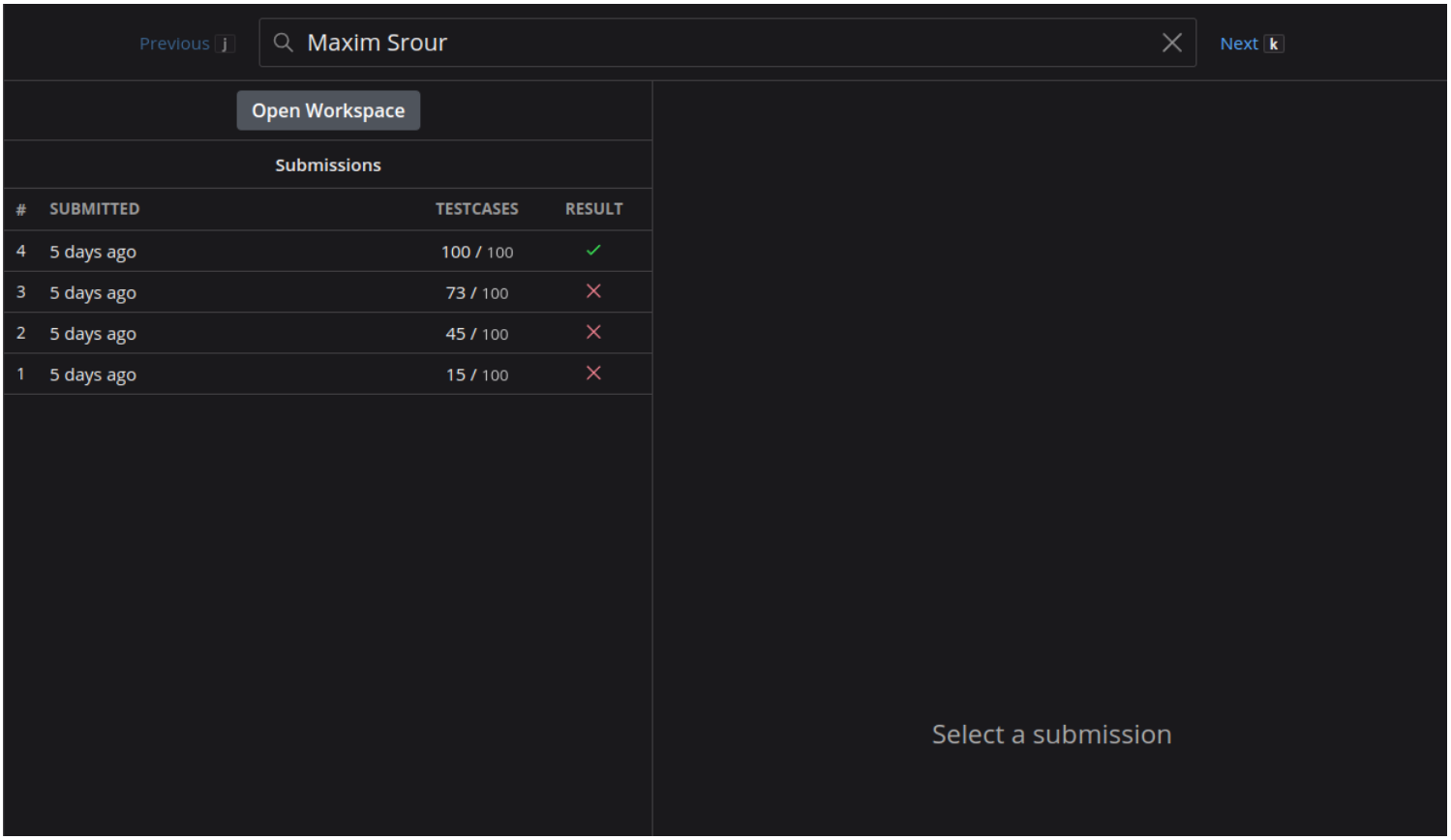
There are two key details to be mindful of. First, is that `100/100 passed` means that this code is passing all 100 of the public tests. If it showed `57/100 passed`, then it would mean only 57 public tests have been passed. Once you pass all the public tests, you will receive a 'green tick' for the whole slide, but this does not necessarily mean that you will pass all private tests and get full marks.

⚠ A green tick for the task also has no bearing on the mark you will receive for the quality of the code, as this is assessed manually by the TAs. Make sure that each task conforms to the requirements independently of the number of passing tests.

You will also see the text `0 points` next to every public test and at the top in the summary. This is because public tests are not assessed, hence why they have zero points allocated to them.

Viewing previous Ed submissions

Every time you run the test cases on Ed, it will generate a new 'submission' which we consider for the checkpoint grades. You can see exactly how many test cases you had passed at a particular point in time, as well as review your code for that submission by clicking on it.



If you have made a change to your code and are now failing more test cases, you can use this to review what you had before or to revert back to it entirely.

How will my work be assessed?

Your assignment will be marked based on 3 criteria:

1. **Program quality**, in terms of simplicity and readability, assessed by a TA who will review your code,
2. **Program correctness**, measured by the output of your programs,
3. **Weekly checkpoints**, to reward timely work.

Automated tests

Both criteria 2 and 3 rely on input/output tests of your programs to measure their correctness. This means we test each of your programs with a series of inputs and measure whether they produce the expected output.

For each task, we put 100 public tests at your disposal to help you create correct programs. They are available both in the scaffold and via the "mark" button at the bottom right of each slide:

However, we will use different, private tests, to evaluate your programs. So, passing a given number of public tests is no guarantee you will achieve a specific mark. Therefore, you can use the public tests to guide you, but, first and foremost, make sure your program follows the assignment brief.

1. Program quality

After you submit on Moodle, a TA will review your code and mark its simplicity and readability. This includes:

1. Clear and meaningful variable names, formatted in snake_case,
2. In-line comments (where required),
3. Documentation (at the top of the file),
4. Code clarity.

The review will occur outside of class, after the Moodle submission deadline.

1.1 Variable names

Variable names should be clear. Following the Python variable naming convention will help you achieve clarity. However, this will not be directly assessed.

Variable names should also be meaningful. This means that the name of the variable should tell a human reader what the role of the variable is in the program, including their type.

```
# Bad variable name
x = int(input("Enter your age: "))

# Good variable name
user_age = int(input("Enter your age: "))
```

The variable names chosen must not override any existing in-built variables. Doing so is bad practice and can lead to difficult-to-debug errors.

```
# This will convert the int 1 into a string '1'
test_conversion = str(1)
print(type(test_conversion))

# This is overriding the existing str() function with a value
# You should NEVER do this
str = "this is a bad idea"

# This code is identical to the code above, but it causes an error because
# the original function has been overriden.
test_conversion = str(1)
print(type(test_conversion))
```

1.2 In-line comments

You are expected to use in-line comments throughout your program to make sure it is clear to the reader. The amount of comments necessary depends on how clear the code is by itself. Therefore, a good habit is to write clear code, so that the amount of in-line comments required is reduced. This will also reduce the number of bugs created.

Comment your code as you write it rather than after reaching a final solution, as it's a lot easier to remember why certain decisions were made and the thought process behind them when the code is fresh in your mind.

Ensure your variable names are meaningful and concise so that your code is understandable at first read. When a reader is going through your program, it should read like a story that is self-explanatory, using in-line comments to explain the logic behind blocks of code and any additional information that is relevant to the functionality of the code.

Avoid repeating yourself by commenting on lines of code that are self-explanatory, and instead reserve your comments to explain higher level logic, such as explaining the overall purpose behind a small block of code.

```
"""
This is an example of over-commented code.
Each line is already quite self-explanatory.
```

```
"""

# This is the first mark
first_mark = float(input("Enter the first mark: "))
# This is the second mark
second_mark = float(input("Enter the second mark: "))
# Computes the average of the two marks
average_mark = (first_mark + second_mark) / 2
# Computes the correctly rounded grade
rounded_grade = int(average_mark + 0.5)
```

1.3 Documentation

You are expected to include documentation at the start of each file you submit. For example:

```
"""
This program reads a name from the user and then greets them.
"""

name = input("What's your name?\n")
print("Greetings,", name)
```

You are also expected to write documentation for every function you define.

```
def name_input() -> str:
    """
    Prompts the user to input their name, and returns it.
    """

    name = input("What's your name?\n")
    return name
```

You are also expected to write documentation for every class and methods you define.

```
class Student:
    """
    A representation of a student.
    """
    def __init__(self, name: str):
        """
        Creates a Student with a name.
        """
        self.name = name

    def __str__(self) -> str:
        """
        Returns a string representating of the Student.
        @return a string representing the Student.
        """
        return "Student " + self.name

print(Student("Robbie"))
```

Documentation should clarify the purpose of a function/file and should provide a high-level overview. Your documentation should not be a rewrite of your comments.

Documentation should be written using docstrings `""" Like this """`, and they should be placed **inside** functions.

If you use comments `# like this`, you won't be getting marks for documentation, as they aren't included in the .__doc__ attribute.

1.4 Code clarity

Optimise your code for readability. In this unit, we do not care about efficiency. So, if the code can be written in multiple ways, opt for the simplest one. A human reader should be able to understand your code quickly and easily.

2. Program correctness

After you submit on Moodle, we will run automated tests on your Moodle submission.

3. Weekly checkpoints

At the end of each week, we expect you to have completed the tasks of the given week.

To measure this, we will look at the percentage of private automated tests passed (i.e. not failed) by your best ed submission made before the weekly checkpoint deadline.

It is therefore necessary for you to run the automated tests on ed if you want to get these marks. You can do so by clicking the "mark" button of each slide.

You are free to edit your code outside ed if you prefer (e.g., a locally installed PyCharm, VSCode, etc), as long as you copy your code on ed to run the tests.

For you to get marks in the checkpoint, you will need to run the automated tests on your assignment in your local Ed workspace. If you don't, you won't be able to get marks, even if your team mate/s do.

Mark breakdown

The assignment has 15 marks, so 1 mark in the assignment equals one mark in the unit.

- Program quality: 5 marks
- Week 10 (Tasks 1-3):
    - Task 1 correctness: 0.5 marks
    - Task 2 correctness: 2 marks
    - Task 3 correctness: 1.5 marks
    - Weekly checkpoint: 1 mark
    - Subtotal: 5 marks
- Week 11 (Tasks 4-6):
    - Task 4 correctness: 1.5 marks
    - Task 5 correctness: 1.5 marks
    - Task 6 correctness: 1 mark
    - Subtotal: 4 marks
- Correctly formatted Moodle submission: 1 mark
- Total: 15 marks

Program quality

Maximum achievable mark

A maximum of 5 marks can be achieved for program quality for the 6 tasks.

This is further capped depending on how much of the first 5 tasks you have completed.

This is measured by the fraction of the private tests that your programs pass for these 6 tasks.

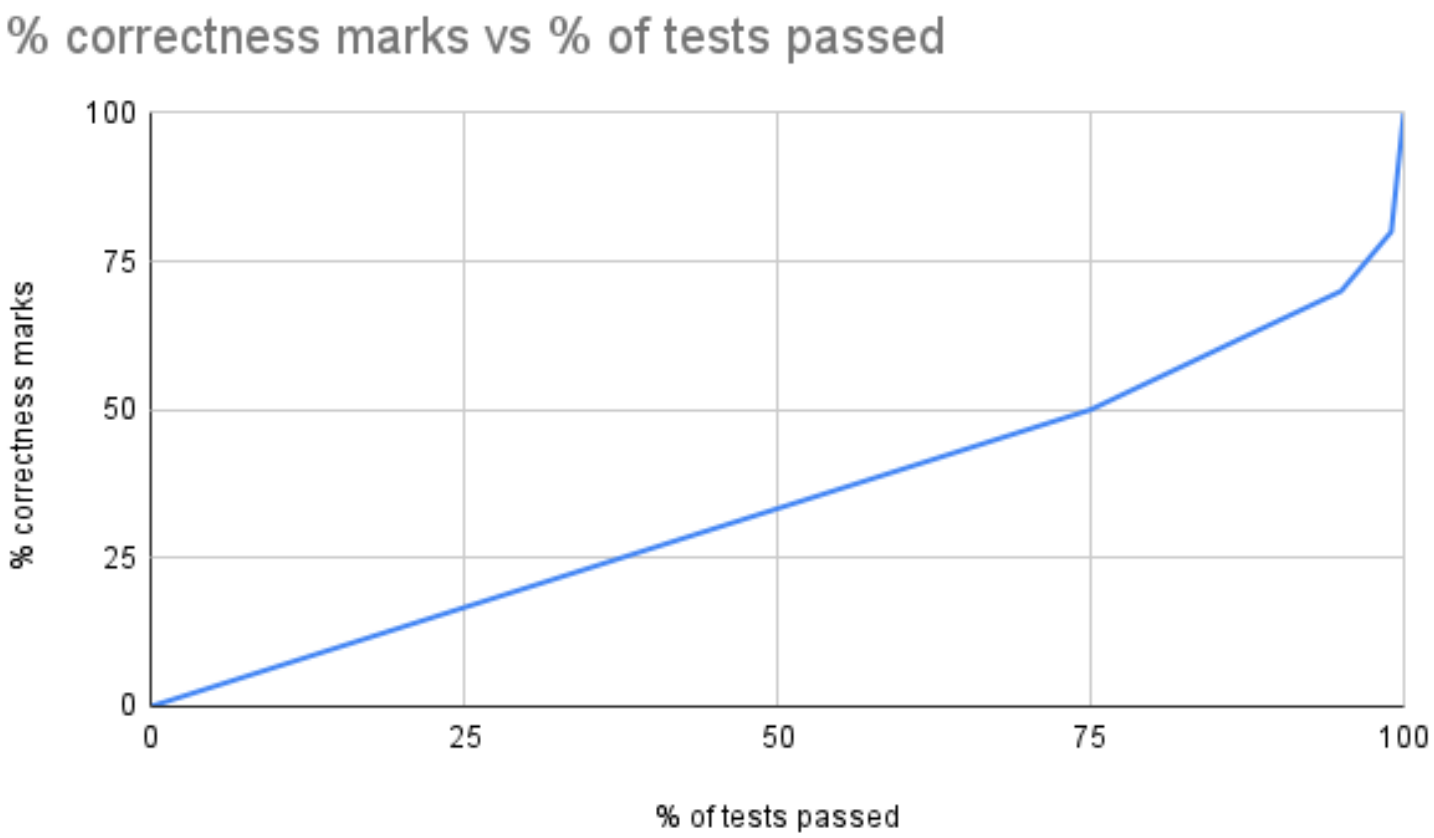
For instance, if your programs pass 100% of the private tests on task 1, 80% of the tests on task 2 and 30% of the tests on task 3, and 0% on tasks 4-6, then the quality mark is capped at (100 + 80 + 30 + 0 + 0 + 0)/500 * 5 = 2.1 marks.

Program correctness

We will evaluate each program with private tests. If your program passes

- 100% of these tests, it receives 100% of the correctness marks,
- 95% to 99% of these tests, it receives 70 to 79% of the correctness marks,
- 75% to 95% of these tests, it receives 50 to 70% of the correctness marks,
- 0% to 75% of these tests, it receives 0 to 50% of the correctness marks.

See chart below.



The rationale is that getting the first 50 tests right is quite easier than getting the last 50 tests right. Furthermore, a program that passes 100% of the tests is more than 1% better than a program that passes 99% of the tests.

Weekly checkpoint

Weekly checkpoint marks are subject to the same curve as program correctness marks.

Penalties

Late penalties

- Up to 7 days: 5% lost per calendar day (or part thereof).
- More than 7 days: Zero (0) marks with no feedback given.

Late penalties are determined using the submission date on Moodle.

Use of external modules

You are allowed to import modules if they are part of the Python Standard Library (e.g. math), but not other modules (e.g. numpy). However, note that the solutions do not require the use of modules.

If your solution to a task uses external modules, it will void (i.e. set to 0) any mark it may have received for this task from program correctness, program quality and weekly checkpoint.

This page is intentionally left blank.

Task 1-6 quality (5 marks)

Write quality code from the get-go.

See how in the slide "How will my work be assessed?".

Task 1: Items and containers (0.5 marks)

In preparation for the summer break, Robbie is coding up an adventure game. He's asked whether you could take care of the looting aspect of the game. With Fibonacci as product manager, what could possibly go wrong?

Task Description

Write a program that reads items and containers from files `items.csv` and `containers.csv`, and prints the list of items.

For this and following tasks, do not create files or write into existing files.

As in previous assignments, you are allowed to use any standard Python feature and module. In particular, `sorted` can be useful here.

We recomend you read all tasks to plan the design of your program so as to minimise the refactoring effort from task to task.

Items

An item has:

- a name, and
- a weight.

We recommend you create a class to represent items.

Containers

A container has:

- a name,
- an empty weight, i.e. their weight when they are empty, and
- a weight capacity, i.e. the maximum combined weight that the container can hold.

- Two copies of the same item or container can exist. If two items or containers have the same name, then they have the same characteristics (e.g. weight).
- Throughout the assignment, all weights and weight-related measures (i.e. weight capacities) are non-negative integers.

We recommend you create a class to represent containers.

Expected output

In the output below, notice "47 items". This is because the containers are included in this count.

There are no user inputs in the example below.

```
Initialised 47 items including 15 containers.

Items:
A normal cheese platter (weight: 1000)
A rock (weight: 1)
Bhagya's publications (weight: 8002)
Charlie's unread emails (weight: 247)
Chloe's half baked ideas (weight: 5)
Chloe's late assignments (weight: 999999)
Crimpy's destroyed cat toys (weight: 27)
Ed's forum posts (weight: 678)
Elena's fishing count (weight: 3500)
Fibonnaci's rabbytes family tree (weight: 144)
Fibonnaci's recursive call count (weight: 10946)
Gabe's Steam game library (weight: 0)
Hui's Hidden Hamster Hoard (weight: 3141)
Lifi's browser tabs (weight: 1337)
Liz's brain cell cluster (weight: 3)
Michael's stack of unmarked assignments (weight: 10000)
Paul's cringe tiktok compilation (weight: 23)
Paul's missing aura points (weight: 22)
Paul's only frontal lobe (weight: 9)
Pierre's daily cheese wheel (weight: 100)
Pierre's funny meme collection (weight: 0)
Pierre's meme collection (weight: 9001)
Pierre's outdated meme collection (weight: 9001)
Pierre's secret meme collection (weight: 420)
Rehan's Book collection (weight: 7005)
Robbie's final drop of sanity (weight: 0)
Robbie's shower thoughts (weight: 150)
Sam's water pouch (weight: 1)
Tan's Tamagotchi Support Group (weight: 410)
Taylor's ex-lovers list (weight: 999)
Vanessa's hit list (weight: 299)
Vanessa's secret gold stash (weight: 2028)

Containers:
A backpack (total weight: 40, empty weight: 40, capacity: 0/5000)
A carrybag for pets (total weight: 50, empty weight: 50, capacity: 0/2000)
A coles shopping bag (total weight: 1, empty weight: 1, capacity: 0/1000)
A container (total weight: 100, empty weight: 100, capacity: 0/250000)
A delicate flower vase (total weight: 5, empty weight: 5, capacity: 0/25)
A full bag of chips (total weight: 2, empty weight: 2, capacity: 0/5)
```

```
A large pouch (total weight: 3, empty weight: 3, capacity: 0/80)
A medium pocket (total weight: 0, empty weight: 0, capacity: 0/200)
A small pocket (total weight: 0, empty weight: 0, capacity: 0/100)
A small pouch (total weight: 1, empty weight: 1, capacity: 0/20)
A suitcase (total weight: 100, empty weight: 100, capacity: 0/20000)
A wheelbarrow (total weight: 100, empty weight: 100, capacity: 0/10000)
A woolworths shopping bag (total weight: 1, empty weight: 1, capacity: 0/1200)
Joey's water bowl (total weight: 2, empty weight: 2, capacity: 0/20)
One of those blue ikea tote bags (total weight: 3, empty weight: 3, capacity: 0/8000)
```

Task 2: Looting items (2 marks)

"Now all we need to do is get the user to pick up loot and store it in the container!" Robbie explains. "Bada bing, bada boom. It's in the bag! Get it?".

You do.

Task Description

After reading all items and containers, do not print them, but instead ask the user for a container to pick for the adventure. For example,

```
Enter the name of the container: A backpack
```

Main menu

Then, a menu with three options will be shown repeatedly. The program prompt of the main menu looks like:

```
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
```

1. Loot item.

Upon entering **1**, the program will then ask for the name of an item to loot. If the item can fit in the container given the remaining capacity, the program indicates so, as shown below.

```
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: A rock
Success! Item "A rock" stored in container "A backpack".
```

If, instead, the remaining capacity is not sufficient to store the item, the item is not looted:

```
==================================
```

```
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Fibonnaci's recursive call count
Failure! Item "Fibonnaci's recursive call count" NOT stored in container "A backpack".
```

If the item's name is not one of the known items, then the user is asked for the name again:

```
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: A smaller Fibonnaci's recursive call count
"A smaller Fibonnaci's recursive call count" not found. Try again.
Enter the name of the item: Fibonnaci's rabbytes family tree
Success! Item "Fibonnaci's rabbytes family tree" stored in container "A backpack".
```

See example 1 for a complete example.

✅ Consider using exceptions (including custom ones) to handle the case where a container cannot store an item.

2. List looted items.

Upon entering **2**, the program will then print the container and the list of content, in the order they have been looted:

```
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
A backpack (total weight: 186, empty weight: 40, capacity: 146/5000)
   A rock (weight: 1)
   Fibonnaci's rabbytes family tree (weight: 144)
   A rock (weight: 1)
```

✅ Notice how the total weight and capacity of the backpack are updated based on the contents.

Examples

User inputs are in **bold font** below.

Example 1

```
Initialised 47 items including 15 containers.

Enter the name of the container: A backpack
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: A rock
Success! Item "A rock" stored in container "A backpack".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Fibonnaci's recursive call count
Failure! Item "Fibonnaci's recursive call count" NOT stored in container "A backpack".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
A backpack (total weight: 41, empty weight: 40, capacity: 1/5000)
   A rock (weight: 1)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: A smaller Fibonnaci's recursive call count
"A smaller Fibonnaci's recursive call count" not found. Try again.
Enter the name of the item: Fibonnaci's rabbytes family tree
Success! Item "Fibonnaci's rabbytes family tree" stored in container "A backpack".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
A backpack (total weight: 185, empty weight: 40, capacity: 145/5000)
   A rock (weight: 1)
   Fibonnaci's rabbytes family tree (weight: 144)
```

```
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: A rock
Success! Item "A rock" stored in container "A backpack".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
A backpack (total weight: 186, empty weight: 40, capacity: 146/5000)
   A rock (weight: 1)
   Fibonnaci's rabbytes family tree (weight: 144)
   A rock (weight: 1)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
0
```

Example 2

```
Initialised 47 items including 15 containers.

Enter the name of the container: A bag
"A bag" not found. Try again.
Enter the name of the container: A rock
"A rock" not found. Try again.
Enter the name of the container: Joey's water bowl
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Paul's only frontal lobe
Success! Item "Paul's only frontal lobe" stored in container "Joey's water bowl".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
```

1
Enter the name of the item: **Liz's brain cell cluster**
Success! Item "Liz's brain cell cluster" stored in container "Joey's water bowl".
==========
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==========
1
Enter the name of the item: **Paul's only frontal lobe**
Failure! Item "Paul's only frontal lobe" NOT stored in container "Joey's water bowl".
==========
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==========
1
Enter the name of the item: **Liz's brain cell cluster**
Success! Item "Liz's brain cell cluster" stored in container "Joey's water bowl".
==========
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==========
2
Joey's water bowl (total weight: 17, empty weight: 2, capacity: 15/20)
   Paul's only frontal lobe (weight: 9)
   Liz's brain cell cluster (weight: 3)
   Liz's brain cell cluster (weight: 3)
==========
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==========
1
Enter the name of the item: **Chloe's half baked ideas**
Success! Item "Chloe's half baked ideas" stored in container "Joey's water bowl".
==========
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==========
1
Enter the name of the item: **Sam's water pouch**
Failure! Item "Sam's water pouch" NOT stored in container "Joey's water bowl".
==========
Enter your choice:
1. Loot item.
2. List looted items.

```
0. Quit.
==================================
1
Enter the name of the item: Pierre's funny meme collection
Success! Item "Pierre's funny meme collection" stored in container "Joey's water bowl".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
Joey's water bowl (total weight: 22, empty weight: 2, capacity: 20/20)
   Paul's only frontal lobe (weight: 9)
   Liz's brain cell cluster (weight: 3)
   Liz's brain cell cluster (weight: 3)
   Chloe's half baked ideas (weight: 5)
   Pierre's funny meme collection (weight: 0)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
0
```

Task 3: Multiple compartments (1.5 marks)

"... then the user picks up the loot, and it's in the bag", finishes Robbie, with a smirk. "So, how do you like the game?"

"Forget it! A container with a single compartment is not realistic," opines Fibonacci. "Even in a video game!"

Robbie actually sort of agrees. Looks like you're not done yet...

Task Description

An additional file, `multi_containers.csv`, now provides the description of containers that have multiple compartments, each behaving like an independent container. The menu itself does not change.

Container with multiple compartments

The empty weight of a container with multiple compartments is the sum of the empty weights of the compartments.

When looting an item, the item can be looted if it fits in one compartment. If it fits in multiple compartments, it is placed in the first one (in the order listed in the file). Once an item is placed in a compartment, it is not moved, even if that would allow looting more items.

Displaying a container with multiple compartments now shows the items stored in each compartment:

```
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
A coles shopping cart (total weight: 3720, empty weight: 43, capacity: 0/0)
   A coles shopping bag (total weight: 1001, empty weight: 1, capacity: 1000/1000)
      A normal cheese platter (weight: 1000)
   A backpack (total weight: 2717, empty weight: 40, capacity: 2677/5000)
      Taylor's ex-lovers list (weight: 999)
      Ed's forum posts (weight: 678)
      A normal cheese platter (weight: 1000)
   A woolworths shopping bag (total weight: 1, empty weight: 1, capacity: 0/1200)
   A coles shopping bag (total weight: 1, empty weight: 1, capacity: 0/1000)
```

See Example 1 for a complete example.

Consider using classes and inheritance in your program.

Your program must retain the same functionality as in Task 2, which is to say that it should be possible to select at the beginning of the program either a standard container, or a container with multiple compartments. See Example 2.

Examples

User inputs are in **bold font** below.

Example 1

```
Initialised 52 items including 20 containers.

Enter the name of the container: A coles shopping cart
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
A coles shopping cart (total weight: 43, empty weight: 43, capacity: 0/0)
   A coles shopping bag (total weight: 1, empty weight: 1, capacity: 0/1000)
   A backpack (total weight: 40, empty weight: 40, capacity: 0/5000)
   A woolworths shopping bag (total weight: 1, empty weight: 1, capacity: 0/1200)
   A coles shopping bag (total weight: 1, empty weight: 1, capacity: 0/1000)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Pierre's outdated meme collection
Failure! Item "Pierre's outdated meme collection" NOT stored in container "A coles shopping cart".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: 1
"1" not found. Try again.
Enter the name of the item: A normal cheese platter
Success! Item "A normal cheese platter" stored in container "A coles shopping cart".
==================================
```

```
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Taylor's ex-lovers list
Success! Item "Taylor's ex-lovers list" stored in container "A coles shopping cart".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Ed's forum posts
Success! Item "Ed's forum posts" stored in container "A coles shopping cart".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
A coles shopping cart (total weight: 2720, empty weight: 43, capacity: 0/0)
   A coles shopping bag (total weight: 1001, empty weight: 1, capacity: 1000/1000)
      A normal cheese platter (weight: 1000)
   A backpack (total weight: 1717, empty weight: 40, capacity: 1677/5000)
      Taylor's ex-lovers list (weight: 999)
      Ed's forum posts (weight: 678)
   A woolworths shopping bag (total weight: 1, empty weight: 1, capacity: 0/1200)
   A coles shopping bag (total weight: 1, empty weight: 1, capacity: 0/1000)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Elena's fishing count
Failure! Item "Elena's fishing count" NOT stored in container "A coles shopping cart".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: A normal cheese platter
Success! Item "A normal cheese platter" stored in container "A coles shopping cart".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
```

```
0. Quit.
==================================
2
A coles shopping cart (total weight: 3720, empty weight: 43, capacity: 0/0)
   A coles shopping bag (total weight: 1001, empty weight: 1, capacity: 1000/1000)
      A normal cheese platter (weight: 1000)
   A backpack (total weight: 2717, empty weight: 40, capacity: 2677/5000)
      Taylor's ex-lovers list (weight: 999)
      Ed's forum posts (weight: 678)
      A normal cheese platter (weight: 1000)
   A woolworths shopping bag (total weight: 1, empty weight: 1, capacity: 0/1200)
   A coles shopping bag (total weight: 1, empty weight: 1, capacity: 0/1000)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
0
```

Example 2

```
Initialised 52 items including 20 containers.

Enter the name of the container: A pouch
"A pouch" not found. Try again.
Enter the name of the container: A suitcase
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: A rock
Success! Item "A rock" stored in container "A suitcase".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
A suitcase (total weight: 101, empty weight: 100, capacity: 1/20000)
   A rock (weight: 1)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
0
```

Week 10 checkpoint (1 mark)

The tasks of week 10 are Tasks 1, 2 and 3.

The checkpoint is Friday, 4 October 2024, 23:55 (Melbourne time), 21:55 (Malaysia time).

This page is intentionally left blank.

Task 4: Magic containers (1.5 marks)

"It's not a proper adventure game without magic items," asserts Fibo. Robbie concedes, reluctantly. You know what must be done

Task Description

An additional file, `magic_containers.csv`, now provides the description of containers that behave like containers with a single compartment, but that do no increase in weight if items are stored in them. They still have a maximum capacity, though.

Magic containers

Displaying a magic container looks like:

```
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
Bag of Holding (total weight: 40, empty weight: 40, capacity: 4910/5000)
   A normal cheese platter (weight: 1000)
   Elena's fishing count (weight: 3500)
   Tan's Tamagotchi Support Group (weight: 410)
   Pierre's funny meme collection (weight: 0)
```

The total weight remains equal to the empty way. The capacity is computed as for a non-magical container. See Example 1 for a complete example.

> Your program must retain the same functionality as in Task 3, which is to say that it should be possible to select at the beginning any of the containers covered so far. See Example 2.

Examples

User inputs are in **bold font** below.

Example 1

```
Initialised 57 items including 25 containers.

Enter the name of the container: Bag of Holding
```

```
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
Bag of Holding (total weight: 40, empty weight: 40, capacity: 0/5000)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: A normal cheese platter
Success! Item "A normal cheese platter" stored in container "Bag of Holding".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
Bag of Holding (total weight: 40, empty weight: 40, capacity: 1000/5000)
   A normal cheese platter (weight: 1000)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Elena's fishing count
Success! Item "Elena's fishing count" stored in container "Bag of Holding".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Vanessa's secret gold stash
Failure! Item "Vanessa's secret gold stash" NOT stored in container "Bag of Holding".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
Bag of Holding (total weight: 40, empty weight: 40, capacity: 4500/5000)
   A normal cheese platter (weight: 1000)
   Elena's fishing count (weight: 3500)
```

```
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Tan's Tamagotchi Support Group
Success! Item "Tan's Tamagotchi Support Group" stored in container "Bag of Holding".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Pierre's funny meme collection
Success! Item "Pierre's funny meme collection" stored in container "Bag of Holding".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
Bag of Holding (total weight: 40, empty weight: 40, capacity: 4910/5000)
   A normal cheese platter (weight: 1000)
   Elena's fishing count (weight: 3500)
   Tan's Tamagotchi Support Group (weight: 410)
   Pierre's funny meme collection (weight: 0)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
0
```

Example 2

```
Initialised 57 items including 25 containers.

Enter the name of the container: A lab coat
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
A lab coat (total weight: 0, empty weight: 0, capacity: 0/0)
   A small pocket (total weight: 0, empty weight: 0, capacity: 0/100)
```

```
   A medium pocket (total weight: 0, empty weight: 0, capacity: 0/200)
   A small pocket (total weight: 0, empty weight: 0, capacity: 0/100)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Gabe's Steam game library
Success! Item "Gabe's Steam game library" stored in container "A lab coat".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Robbie's final drop of sanity
Success! Item "Robbie's final drop of sanity" stored in container "A lab coat".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
A lab coat (total weight: 0, empty weight: 0, capacity: 0/0)
   A small pocket (total weight: 0, empty weight: 0, capacity: 0/100)
      Gabe's Steam game library (weight: 0)
      Robbie's final drop of sanity (weight: 0)
   A medium pocket (total weight: 0, empty weight: 0, capacity: 0/200)
   A small pocket (total weight: 0, empty weight: 0, capacity: 0/100)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Robbie's shower thoughts
Success! Item "Robbie's shower thoughts" stored in container "A lab coat".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
A lab coat (total weight: 150, empty weight: 0, capacity: 0/0)
   A small pocket (total weight: 0, empty weight: 0, capacity: 0/100)
      Gabe's Steam game library (weight: 0)
      Robbie's final drop of sanity (weight: 0)
   A medium pocket (total weight: 150, empty weight: 0, capacity: 150/200)
```

```
      Robbie's shower thoughts (weight: 150)
   A small pocket (total weight: 0, empty weight: 0, capacity: 0/100)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
0
```

Task 5: Magic multiple compartments (1.5 mark)

"But what about these containers with multiple compartments? Can't they be magic? That's discriminatory!" explodes Fibonacci □□□

"Beep", sighs Robbie 🙄

Task Description

An additional file, `magic_multi_containers.csv`, now provides the description of containers that behave like containers with multiple compartments, and, like the other magic containers, do no increase in weight if items are stored in them. Each compartment still has a maximum capacity.

Magic containers with multiple compartments

Displaying a magic container looks like:

```
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
Professor Farnsworth's Lab Coat (total weight: 0, empty weight: 0, capacity: 0/0)
   A small pocket (total weight: 100, empty weight: 0, capacity: 100/100)
      Pierre's daily cheese wheel (weight: 100)
      Gabe's Steam game library (weight: 0)
      Robbie's final drop of sanity (weight: 0)
   A medium pocket (total weight: 186, empty weight: 0, capacity: 186/200)
      Paul's only frontal lobe (weight: 9)
      Robbie's shower thoughts (weight: 150)
      Crimpy's destroyed cat toys (weight: 27)
   A small pocket (total weight: 27, empty weight: 0, capacity: 27/100)
      Crimpy's destroyed cat toys (weight: 27)
==================================
```

The total weight remains equal to the empty way. The capacity is computed as for a non-magical container with multiple compartments, which is that it is computed and displayed at the compartment level. See Example 1 for a complete example.

> Your program must retain the same functionality as in Task 4, which is to say that it should be possible to select at the beginning any of the containers covered so far. See Example 2.

Examples

User inputs are in **bold font** below.

Example 1

```
Initialised 60 items including 28 containers.

Enter the name of the container: Professor Farnsworth's Lab Coat
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
Professor Farnsworth's Lab Coat (total weight: 0, empty weight: 0, capacity: 0/0)
   A small pocket (total weight: 0, empty weight: 0, capacity: 0/100)
   A medium pocket (total weight: 0, empty weight: 0, capacity: 0/200)
   A small pocket (total weight: 0, empty weight: 0, capacity: 0/100)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Pierre's daily cheese wheel
Success! Item "Pierre's daily cheese wheel" stored in container "Professor Farnsworth's Lab Coat".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Gabe's Steam game library
Success! Item "Gabe's Steam game library" stored in container "Professor Farnsworth's Lab Coat".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
Professor Farnsworth's Lab Coat (total weight: 0, empty weight: 0, capacity: 0/0)
   A small pocket (total weight: 100, empty weight: 0, capacity: 100/100)
      Pierre's daily cheese wheel (weight: 100)
      Gabe's Steam game library (weight: 0)
   A medium pocket (total weight: 0, empty weight: 0, capacity: 0/200)
   A small pocket (total weight: 0, empty weight: 0, capacity: 0/100)
==================================
```

```
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Paul's only frontal lobe
Success! Item "Paul's only frontal lobe" stored in container "Professor Farnsworth's Lab Coat".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
Professor Farnsworth's Lab Coat (total weight: 0, empty weight: 0, capacity: 0/0)
   A small pocket (total weight: 100, empty weight: 0, capacity: 100/100)
      Pierre's daily cheese wheel (weight: 100)
      Gabe's Steam game library (weight: 0)
   A medium pocket (total weight: 9, empty weight: 0, capacity: 9/200)
      Paul's only frontal lobe (weight: 9)
   A small pocket (total weight: 0, empty weight: 0, capacity: 0/100)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Robbie's final drop of sanity
Success! Item "Robbie's final drop of sanity" stored in container "Professor Farnsworth's Lab Coat"
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
Professor Farnsworth's Lab Coat (total weight: 0, empty weight: 0, capacity: 0/0)
   A small pocket (total weight: 100, empty weight: 0, capacity: 100/100)
      Pierre's daily cheese wheel (weight: 100)
      Gabe's Steam game library (weight: 0)
      Robbie's final drop of sanity (weight: 0)
   A medium pocket (total weight: 9, empty weight: 0, capacity: 9/200)
      Paul's only frontal lobe (weight: 9)
   A small pocket (total weight: 0, empty weight: 0, capacity: 0/100)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Robbie's shower thoughts
```

```
Success! Item "Robbie's shower thoughts" stored in container "Professor Farnsworth's Lab Coat".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
Professor Farnsworth's Lab Coat (total weight: 0, empty weight: 0, capacity: 0/0)
   A small pocket (total weight: 100, empty weight: 0, capacity: 100/100)
      Pierre's daily cheese wheel (weight: 100)
      Gabe's Steam game library (weight: 0)
      Robbie's final drop of sanity (weight: 0)
   A medium pocket (total weight: 159, empty weight: 0, capacity: 159/200)
      Paul's only frontal lobe (weight: 9)
      Robbie's shower thoughts (weight: 150)
   A small pocket (total weight: 0, empty weight: 0, capacity: 0/100)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Crimpy's destroyed cat toys
Success! Item "Crimpy's destroyed cat toys" stored in container "Professor Farnsworth's Lab Coat".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
Professor Farnsworth's Lab Coat (total weight: 0, empty weight: 0, capacity: 0/0)
   A small pocket (total weight: 100, empty weight: 0, capacity: 100/100)
      Pierre's daily cheese wheel (weight: 100)
      Gabe's Steam game library (weight: 0)
      Robbie's final drop of sanity (weight: 0)
   A medium pocket (total weight: 186, empty weight: 0, capacity: 186/200)
      Paul's only frontal lobe (weight: 9)
      Robbie's shower thoughts (weight: 150)
      Crimpy's destroyed cat toys (weight: 27)
   A small pocket (total weight: 0, empty weight: 0, capacity: 0/100)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Crimpy's destroyed cat toys
Success! Item "Crimpy's destroyed cat toys" stored in container "Professor Farnsworth's Lab Coat".
==================================
Enter your choice:
```

```
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
Professor Farnsworth's Lab Coat (total weight: 0, empty weight: 0, capacity: 0/0)
   A small pocket (total weight: 100, empty weight: 0, capacity: 100/100)
      Pierre's daily cheese wheel (weight: 100)
      Gabe's Steam game library (weight: 0)
      Robbie's final drop of sanity (weight: 0)
   A medium pocket (total weight: 186, empty weight: 0, capacity: 186/200)
      Paul's only frontal lobe (weight: 9)
      Robbie's shower thoughts (weight: 150)
      Crimpy's destroyed cat toys (weight: 27)
   A small pocket (total weight: 27, empty weight: 0, capacity: 27/100)
      Crimpy's destroyed cat toys (weight: 27)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
0
```

Example 2

```
Initialised 60 items including 28 containers.

Enter the name of the container: Hermione's Satchel
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
Hermione's Satchel (total weight: 3, empty weight: 3, capacity: 0/80)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Fibonnaci's rabbytes family tree
Failure! Item "Fibonnaci's rabbytes family tree" NOT stored in container "Hermione's Satchel".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
```

```
==================================
1
Enter the name of the item: Sam's water pouch
Success! Item "Sam's water pouch" stored in container "Hermione's Satchel".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
Hermione's Satchel (total weight: 3, empty weight: 3, capacity: 1/80)
   Sam's water pouch (weight: 1)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
0
```

Task 6: Containers are items too (1 mark)

"What do you mean the program won't let me pick up a container? Containers are items too! Always have been!" fulminates Fibonacci. "Anyway, if the program doesn't allow it yet, it's only a small matter of programming."

"And we should be able to make pyramids of them and create an infinite capacity loophole. Handy Haversacks go brrrrrr," deadpans Robbie.

"Ooh, recursion, that's brilliant! Let's get the student to do that! Add to cart!", exults Fibo.

A few moments later

"I'm sorry, mate", apologises Robbie. "This one is kind of on me. I understand if you've had enough. You probably have better things to do. Anyway, you know the green tick is a lie, right?"

Robbie makes a valid point. This is going to take hours. What about your other assignments?

Task Description

Expand your program so that:

- Containers can be looted and stored just like other items.
- When looting and storing an item (including containers), it is placed in the first possible container:
    - if it can be stored in the current container, it is,
    - otherwise, containers (and other containers within) placed in the current container are tested in the order in which they were stored to determine if the item can be stored there, and
    - any container inside the current container is tested before any container at the same level as the current container.
- Items more than one level inside a container affect the weight of all containers containing this item (but a magic container will also affect this). This means that if item X is put in non-magic container B, itself inside non-magic container A, then the weight of X both applies to A and B, and they both must have enough capacity. However, if B is magic, then the weight of X only applies to B. See Examples 1 and 2.

This type of recursive search is called a preorder depth-first search.

Examples

User inputs are in **bold font** below.

Example 1

```
Initialised 60 items including 28 containers.

Enter the name of the container: One of those blue ikea tote bags
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: A container
Success! Item "A container" stored in container "One of those blue ikea tote bags".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
One of those blue ikea tote bags (total weight: 103, empty weight: 3, capacity: 100/8000)
   A container (total weight: 100, empty weight: 100, capacity: 0/250000)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Hui's Hidden Hamster Hoard
Success! Item "Hui's Hidden Hamster Hoard" stored in container "One of those blue ikea tote bags".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
One of those blue ikea tote bags (total weight: 3244, empty weight: 3, capacity: 3241/8000)
   A container (total weight: 100, empty weight: 100, capacity: 0/250000)
   Hui's Hidden Hamster Hoard (weight: 3141)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Fibonnaci's recursive call count
Failure! Item "Fibonnaci's recursive call count" NOT stored in container "One of those blue ikea to
```

```
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Rehan's Book collection
Failure! Item "Rehan's Book collection" NOT stored in container "One of those blue ikea tote bags".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
One of those blue ikea tote bags (total weight: 3244, empty weight: 3, capacity: 3241/8000)
   A container (total weight: 100, empty weight: 100, capacity: 0/250000)
   Hui's Hidden Hamster Hoard (weight: 3141)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
0
```

Example 2

```
Initialised 60 items including 28 containers.

Enter the name of the container: A small pouch
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Bag of Holding
Failure! Item "Bag of Holding" NOT stored in container "A small pouch".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Hermione's Satchel
Success! Item "Hermione's Satchel" stored in container "A small pouch".
==================================
```

```
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
A small pouch (total weight: 4, empty weight: 1, capacity: 3/20)
   Hermione's Satchel (total weight: 3, empty weight: 3, capacity: 0/80)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Paul's only frontal lobe
Success! Item "Paul's only frontal lobe" stored in container "A small pouch".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
A small pouch (total weight: 13, empty weight: 1, capacity: 12/20)
   Hermione's Satchel (total weight: 3, empty weight: 3, capacity: 0/80)
   Paul's only frontal lobe (weight: 9)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Paul's only frontal lobe
Success! Item "Paul's only frontal lobe" stored in container "A small pouch".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
A small pouch (total weight: 13, empty weight: 1, capacity: 12/20)
   Hermione's Satchel (total weight: 3, empty weight: 3, capacity: 9/80)
      Paul's only frontal lobe (weight: 9)
   Paul's only frontal lobe (weight: 9)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
```

```
Enter the name of the item: Chloe's half baked ideas
Success! Item "Chloe's half baked ideas" stored in container "A small pouch".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Liz's brain cell cluster
Success! Item "Liz's brain cell cluster" stored in container "A small pouch".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
A small pouch (total weight: 21, empty weight: 1, capacity: 20/20)
   Hermione's Satchel (total weight: 3, empty weight: 3, capacity: 9/80)
      Paul's only frontal lobe (weight: 9)
   Paul's only frontal lobe (weight: 9)
   Chloe's half baked ideas (weight: 5)
   Liz's brain cell cluster (weight: 3)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Gabe's Steam game library
Success! Item "Gabe's Steam game library" stored in container "A small pouch".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
A small pouch (total weight: 21, empty weight: 1, capacity: 20/20)
   Hermione's Satchel (total weight: 3, empty weight: 3, capacity: 9/80)
      Paul's only frontal lobe (weight: 9)
   Paul's only frontal lobe (weight: 9)
   Chloe's half baked ideas (weight: 5)
   Liz's brain cell cluster (weight: 3)
   Gabe's Steam game library (weight: 0)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
```

```
Enter the name of the item: Robbie's shower thoughts
Failure! Item "Robbie's shower thoughts" NOT stored in container "A small pouch".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
A small pouch (total weight: 21, empty weight: 1, capacity: 20/20)
   Hermione's Satchel (total weight: 3, empty weight: 3, capacity: 9/80)
      Paul's only frontal lobe (weight: 9)
   Paul's only frontal lobe (weight: 9)
   Chloe's half baked ideas (weight: 5)
   Liz's brain cell cluster (weight: 3)
   Gabe's Steam game library (weight: 0)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Vanessa's hit list
Failure! Item "Vanessa's hit list" NOT stored in container "A small pouch".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Bag of Holding
Success! Item "Bag of Holding" stored in container "A small pouch".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
A small pouch (total weight: 21, empty weight: 1, capacity: 20/20)
   Hermione's Satchel (total weight: 3, empty weight: 3, capacity: 49/80)
      Paul's only frontal lobe (weight: 9)
      Bag of Holding (total weight: 40, empty weight: 40, capacity: 0/5000)
   Paul's only frontal lobe (weight: 9)
   Chloe's half baked ideas (weight: 5)
   Liz's brain cell cluster (weight: 3)
   Gabe's Steam game library (weight: 0)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
```

```
==================================
1
Enter the name of the item: Vanessa's hit list
Success! Item "Vanessa's hit list" stored in container "A small pouch".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
A small pouch (total weight: 21, empty weight: 1, capacity: 20/20)
   Hermione's Satchel (total weight: 3, empty weight: 3, capacity: 49/80)
      Paul's only frontal lobe (weight: 9)
      Bag of Holding (total weight: 40, empty weight: 40, capacity: 299/5000)
         Vanessa's hit list (weight: 299)
   Paul's only frontal lobe (weight: 9)
   Chloe's half baked ideas (weight: 5)
   Liz's brain cell cluster (weight: 3)
   Gabe's Steam game library (weight: 0)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Arnaud's Oven of Crispiness
Success! Item "Arnaud's Oven of Crispiness" stored in container "A small pouch".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Lifi's browser tabs
Success! Item "Lifi's browser tabs" stored in container "A small pouch".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
A small pouch (total weight: 21, empty weight: 1, capacity: 20/20)
   Hermione's Satchel (total weight: 3, empty weight: 3, capacity: 49/80)
      Paul's only frontal lobe (weight: 9)
      Bag of Holding (total weight: 40, empty weight: 40, capacity: 1736/5000)
         Vanessa's hit list (weight: 299)
         Arnaud's Oven of Crispiness (total weight: 100, empty weight: 100, capacity: 0/250000)
         Lifi's browser tabs (weight: 1337)
   Paul's only frontal lobe (weight: 9)
   Chloe's half baked ideas (weight: 5)
```

```
   Liz's brain cell cluster (weight: 3)
   Gabe's Steam game library (weight: 0)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Pierre's outdated meme collection
Success! Item "Pierre's outdated meme collection" stored in container "A small pouch".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
A small pouch (total weight: 21, empty weight: 1, capacity: 20/20)
   Hermione's Satchel (total weight: 3, empty weight: 3, capacity: 49/80)
      Paul's only frontal lobe (weight: 9)
      Bag of Holding (total weight: 40, empty weight: 40, capacity: 1736/5000)
         Vanessa's hit list (weight: 299)
         Arnaud's Oven of Crispiness (total weight: 100, empty weight: 100, capacity: 9001/250000)
            Pierre's outdated meme collection (weight: 9001)
         Lifi's browser tabs (weight: 1337)
   Paul's only frontal lobe (weight: 9)
   Chloe's half baked ideas (weight: 5)
   Liz's brain cell cluster (weight: 3)
   Gabe's Steam game library (weight: 0)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
0
```

Example 3

```
Initialised 60 items including 28 containers.

Enter the name of the container: The Stomach of that Penguin from Madagascar
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: A small pouch
Success! Item "A small pouch" stored in container "The Stomach of that Penguin from Madagascar".
```

```
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Rambo's Cargo Pants
Success! Item "Rambo's Cargo Pants" stored in container "The Stomach of that Penguin from Madagasca
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
The Stomach of that Penguin from Madagascar (total weight: 3, empty weight: 3, capacity: 1/80)
   A small pouch (total weight: 1, empty weight: 1, capacity: 0/20)
   Rambo's Cargo Pants (total weight: 0, empty weight: 0, capacity: 0/0)
      A small pocket (total weight: 0, empty weight: 0, capacity: 0/100)
      A medium pocket (total weight: 0, empty weight: 0, capacity: 0/200)
      A medium pocket (total weight: 0, empty weight: 0, capacity: 0/200)
      A small pocket (total weight: 0, empty weight: 0, capacity: 0/100)
      A medium pocket (total weight: 0, empty weight: 0, capacity: 0/200)
      A medium pocket (total weight: 0, empty weight: 0, capacity: 0/200)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Robbie's shower thoughts
Success! Item "Robbie's shower thoughts" stored in container "The Stomach of that Penguin from Mada
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Pierre's daily cheese wheel
Success! Item "Pierre's daily cheese wheel" stored in container "The Stomach of that Penguin from M
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
The Stomach of that Penguin from Madagascar (total weight: 3, empty weight: 3, capacity: 1/80)
   A small pouch (total weight: 1, empty weight: 1, capacity: 0/20)
   Rambo's Cargo Pants (total weight: 0, empty weight: 0, capacity: 0/0)
      A small pocket (total weight: 100, empty weight: 0, capacity: 100/100)
```

```
         Pierre's daily cheese wheel (weight: 100)
      A medium pocket (total weight: 150, empty weight: 0, capacity: 150/200)
         Robbie's shower thoughts (weight: 150)
      A medium pocket (total weight: 0, empty weight: 0, capacity: 0/200)
      A small pocket (total weight: 0, empty weight: 0, capacity: 0/100)
      A medium pocket (total weight: 0, empty weight: 0, capacity: 0/200)
      A medium pocket (total weight: 0, empty weight: 0, capacity: 0/200)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Paul's cringe tiktok compilation
Success! Item "Paul's cringe tiktok compilation" stored in container "The Stomach of that Penguin f
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Crimpy's destroyed cat toys
Success! Item "Crimpy's destroyed cat toys" stored in container "The Stomach of that Penguin from M
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
The Stomach of that Penguin from Madagascar (total weight: 3, empty weight: 3, capacity: 51/80)
   A small pouch (total weight: 1, empty weight: 1, capacity: 0/20)
   Rambo's Cargo Pants (total weight: 0, empty weight: 0, capacity: 0/0)
      A small pocket (total weight: 100, empty weight: 0, capacity: 100/100)
         Pierre's daily cheese wheel (weight: 100)
      A medium pocket (total weight: 150, empty weight: 0, capacity: 150/200)
         Robbie's shower thoughts (weight: 150)
      A medium pocket (total weight: 0, empty weight: 0, capacity: 0/200)
      A small pocket (total weight: 0, empty weight: 0, capacity: 0/100)
      A medium pocket (total weight: 0, empty weight: 0, capacity: 0/200)
      A medium pocket (total weight: 0, empty weight: 0, capacity: 0/200)
   Paul's cringe tiktok compilation (weight: 23)
   Crimpy's destroyed cat toys (weight: 27)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Paul's missing aura points
Success! Item "Paul's missing aura points" stored in container "The Stomach of that Penguin from Ma
```

```
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
The Stomach of that Penguin from Madagascar (total weight: 3, empty weight: 3, capacity: 73/80)
   A small pouch (total weight: 1, empty weight: 1, capacity: 0/20)
   Rambo's Cargo Pants (total weight: 0, empty weight: 0, capacity: 0/0)
      A small pocket (total weight: 100, empty weight: 0, capacity: 100/100)
         Pierre's daily cheese wheel (weight: 100)
      A medium pocket (total weight: 150, empty weight: 0, capacity: 150/200)
         Robbie's shower thoughts (weight: 150)
      A medium pocket (total weight: 0, empty weight: 0, capacity: 0/200)
      A small pocket (total weight: 0, empty weight: 0, capacity: 0/100)
      A medium pocket (total weight: 0, empty weight: 0, capacity: 0/200)
      A medium pocket (total weight: 0, empty weight: 0, capacity: 0/200)
   Paul's cringe tiktok compilation (weight: 23)
   Crimpy's destroyed cat toys (weight: 27)
   Paul's missing aura points (weight: 22)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Paul's only frontal lobe
Success! Item "Paul's only frontal lobe" stored in container "The Stomach of that Penguin from Mada
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
The Stomach of that Penguin from Madagascar (total weight: 3, empty weight: 3, capacity: 73/80)
   A small pouch (total weight: 1, empty weight: 1, capacity: 0/20)
   Rambo's Cargo Pants (total weight: 0, empty weight: 0, capacity: 0/0)
      A small pocket (total weight: 100, empty weight: 0, capacity: 100/100)
         Pierre's daily cheese wheel (weight: 100)
      A medium pocket (total weight: 159, empty weight: 0, capacity: 159/200)
         Robbie's shower thoughts (weight: 150)
         Paul's only frontal lobe (weight: 9)
      A medium pocket (total weight: 0, empty weight: 0, capacity: 0/200)
      A small pocket (total weight: 0, empty weight: 0, capacity: 0/100)
      A medium pocket (total weight: 0, empty weight: 0, capacity: 0/200)
      A medium pocket (total weight: 0, empty weight: 0, capacity: 0/200)
   Paul's cringe tiktok compilation (weight: 23)
   Crimpy's destroyed cat toys (weight: 27)
   Paul's missing aura points (weight: 22)
==================================
Enter your choice:
```

```
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Robbie's final drop of sanity
Success! Item "Robbie's final drop of sanity" stored in container "The Stomach of that Penguin from
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
The Stomach of that Penguin from Madagascar (total weight: 3, empty weight: 3, capacity: 73/80)
   A small pouch (total weight: 1, empty weight: 1, capacity: 0/20)
   Rambo's Cargo Pants (total weight: 0, empty weight: 0, capacity: 0/0)
      A small pocket (total weight: 100, empty weight: 0, capacity: 100/100)
         Pierre's daily cheese wheel (weight: 100)
      A medium pocket (total weight: 159, empty weight: 0, capacity: 159/200)
         Robbie's shower thoughts (weight: 150)
         Paul's only frontal lobe (weight: 9)
      A medium pocket (total weight: 0, empty weight: 0, capacity: 0/200)
      A small pocket (total weight: 0, empty weight: 0, capacity: 0/100)
      A medium pocket (total weight: 0, empty weight: 0, capacity: 0/200)
      A medium pocket (total weight: 0, empty weight: 0, capacity: 0/200)
   Paul's cringe tiktok compilation (weight: 23)
   Crimpy's destroyed cat toys (weight: 27)
   Paul's missing aura points (weight: 22)
   Robbie's final drop of sanity (weight: 0)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: A small pouch
Success! Item "A small pouch" stored in container "The Stomach of that Penguin from Madagascar".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
The Stomach of that Penguin from Madagascar (total weight: 3, empty weight: 3, capacity: 74/80)
   A small pouch (total weight: 1, empty weight: 1, capacity: 0/20)
   Rambo's Cargo Pants (total weight: 0, empty weight: 0, capacity: 0/0)
      A small pocket (total weight: 100, empty weight: 0, capacity: 100/100)
         Pierre's daily cheese wheel (weight: 100)
      A medium pocket (total weight: 159, empty weight: 0, capacity: 159/200)
         Robbie's shower thoughts (weight: 150)
         Paul's only frontal lobe (weight: 9)
```

```
      A medium pocket (total weight: 0, empty weight: 0, capacity: 0/200)
      A small pocket (total weight: 0, empty weight: 0, capacity: 0/100)
      A medium pocket (total weight: 0, empty weight: 0, capacity: 0/200)
      A medium pocket (total weight: 0, empty weight: 0, capacity: 0/200)
   Paul's cringe tiktok compilation (weight: 23)
   Crimpy's destroyed cat toys (weight: 27)
   Paul's missing aura points (weight: 22)
   Robbie's final drop of sanity (weight: 0)
   A small pouch (total weight: 1, empty weight: 1, capacity: 0/20)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Handy Haversack
Success! Item "Handy Haversack" stored in container "The Stomach of that Penguin from Madagascar".
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
2
The Stomach of that Penguin from Madagascar (total weight: 3, empty weight: 3, capacity: 79/80)
   A small pouch (total weight: 1, empty weight: 1, capacity: 0/20)
   Rambo's Cargo Pants (total weight: 0, empty weight: 0, capacity: 0/0)
      A small pocket (total weight: 100, empty weight: 0, capacity: 100/100)
         Pierre's daily cheese wheel (weight: 100)
      A medium pocket (total weight: 159, empty weight: 0, capacity: 159/200)
         Robbie's shower thoughts (weight: 150)
         Paul's only frontal lobe (weight: 9)
      A medium pocket (total weight: 0, empty weight: 0, capacity: 0/200)
      A small pocket (total weight: 0, empty weight: 0, capacity: 0/100)
      A medium pocket (total weight: 0, empty weight: 0, capacity: 0/200)
      A medium pocket (total weight: 0, empty weight: 0, capacity: 0/200)
   Paul's cringe tiktok compilation (weight: 23)
   Crimpy's destroyed cat toys (weight: 27)
   Paul's missing aura points (weight: 22)
   Robbie's final drop of sanity (weight: 0)
   A small pouch (total weight: 1, empty weight: 1, capacity: 0/20)
   Handy Haversack (total weight: 5, empty weight: 5, capacity: 0/0)
      A small pouch (total weight: 1, empty weight: 1, capacity: 0/20)
      A large pouch (total weight: 3, empty weight: 3, capacity: 0/80)
      A small pouch (total weight: 1, empty weight: 1, capacity: 0/20)
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
1
Enter the name of the item: Vanessa's hit list
```

```
Failure! Item "Vanessa's hit list" NOT stored in container "The Stomach of that Penguin from Madaga
==================================
Enter your choice:
1. Loot item.
2. List looted items.
0. Quit.
==================================
0
```

Final submission on Moodle (1 mark)

Only one teammate per team needs to submit on Moodle, on behalf of the whole team.

Your final submission must be one zip file with the necessary and sufficient files, as shown below.

```
A3_<student_1_id>_<student_2_id>.zip
└─> items_and_containers.py
└─> looting_items_containers.py
└─> magic_containers.py
└─> magic_multi_containers.py
└─> multi_containers.py
└─> recursive_containers.py
```

Alternatively, you may also submit in the following format.

```
A3_<student_1_id>_<student_2_id>.zip
└─> A3_<student_1_id>_<student_2_id>
    └─> items_and_containers.py
    └─> looting_items_containers.py
    └─> magic_containers.py
    └─> magic_multi_containers.py
    └─> multi_containers.py
    └─> recursive_containers.py
```

If your submission follows the structures above, you will receive one mark. Make sure that you replace `<student_i_id>` with your actual student ID, and ensure that you use the correct names for each of the files, including the file extension.

For more information, please check the Ed post.

If you are by yourself in your team, use A3_<student_id> for the file and directory names.
If you are in a team of 3, use A3_<student_1_id>_<student_2_id>_<student_3_id>

Some operating systems hide the file extension, so be sure to check if that's the case before "adding it back in".

Do not include the .csv files or confuse them with the .py files.

This page is intentionally left blank.

Changelog (5 changes)

Change 5 - 07/10/2024

In Task 6, added the following sentence

- any container inside the current container is tested before any container at the same level as the current container.

and the following help bubble:

.

Change 4 - 04/10/2024

Changed the formula used to compute the "Maximum achievable mark" in program quality. Task 6 is now excluded from the computation, which means you can get full marks in program quality without attempting Task 6. (This change cannot decrease your mark).

Change 3 - 03/10/2024

Added the highlighted text to Task 6:

- Items more than one level inside a container affect the weight of all containers containing this item (but a magic container will also affect this). This means that if item X is put in non-magic container B, itself inside non-magic container A, then the weight of X both applies to A and B, and they both must have enough capacity. However, if B is magic, then the weight of X only applies to B. See Examples 1 and 2.

Change 2 - 01/10/2024

Added the following to 1.3 Documentation under "How will my work be assessed?"

You are also expected to write documentation for every class and methods you define.

Run PYTHON

```python
class Student:
    """
    A representation of a student.
    """
    def __init__(self, name: str):
        """
        Creates a Student with a name.
        """
        self.name = name

    def __str__(self) -> str:
        """
        Returns a string representating of the Student.
        @return a string representing the Student.
```

Thanks Justin for asking!

Change 1 - 01/10/2024

Added the following help bubble to Task 1.