

Ham Sandwich Maker Machine Program

Requirements:

An automatic ham sandwich maker uses a set of programmed ingredients to make sandwiches. First, users select the size of the sandwich, then the machine determines how much resources are available before making it. The user receives a message if there are insufficient resources. Otherwise, the user must pay for sandwich by coins. The machine will indicate if insufficient coins are inserted or if there is an exchange amount when coins are inserted. Also, on the screen, users can view the remaining resources report and turn off the machine.

1- Prompt user "What would you like? (small/ medium/ large/ off/ report)"

- **small, medium, large:** Check the resources and decide what to do next.
- **off:** Turn off the machine,
- **report:** A report should be generated that shows the current resource values.
- After the sandwich is dispensed, the prompt should display again to serve the next customer once the order has been completed.

2- Check that the resources are sufficient.

- Program should check whether there are enough resources to make the sandwich when the user chooses it. For example, if a small size ham sandwich requires 4 slices of bread but there are only 2 slices left, the machine should stop the process and return the following message: "Sorry there is not enough bread."
- Resources is a dictionary with these initial values:

```
resources = {  
    "bread": 12,  ## slice  
    "ham": 18,   ## slice  
    "cheese": 24 ## ounces  
}
```

3- Process the inserted coins:

- The machine accepts large dollar (\$1), half dollar (\$0.5), quarter (\$0.25) and nickel (\$0.05) coins.
- If there are sufficient resources to make the drink selected, then the program should prompt the user to insert coins.
- Using the coins inserted, calculate their monetary value.

4- Has the transaction been successful?

- Make sure the user has inserted enough money to purchase the sandwich they have selected. IF there is not enough money, machine returns "Sorry, that's not enough money. Money refunded."
- The machine should offer change if the user has inserted too much money by returning the following message: "Here is \$x in change"

5- Make sandwich:

- The ingredients for the sandwich should be deducted from machine resources if the transaction is successful.

- Once all resources have been deducted, returns the following message: "<small> sandwich is ready. Bon appetit!". (You should include the size of ordered sandwich in this message)
- Here are recipes for three types of ham sandwiches:

```

recipes = {
  "small": {
    "ingredients": {
      "bread": 2,  ## slice
      "ham": 4,   ## slice
      "cheese": 4, ## ounces
    },
    "cost": 1.75,
  },
  "medium": {
    "ingredients": {
      "bread": 4,  ## slice
      "ham": 6,   ## slice
      "cheese": 8, ## ounces
    },
    "cost": 3.25,
  },
  "large": {
    "ingredients": {
      "bread": 6,  ## slice
      "ham": 8,   ## slice
      "cheese": 12, ## ounces
    },
    "cost": 5.5,
  }
}

```

6- Show report:

- Resources should be listed along with their current values in the report. It is therefore necessary for the report to show the values that have been deducted after making a sandwich. For example

Sample interactive terminal input/output:

```

What would you like? (small/ medium/ large/ off/ report): small
Please insert coins.
how many large dollars?: 1
how many half dollars?: 2
how many quarters?: 0
how many nickels?: 0
Here is $0.25 in change.
small sandwich is ready. Bon appetit!
What would you like? (small/ medium/ large/ off/ report): report
Bread: 10 slice(s)

```

Ham: 14 slice(s)
Cheese: 20 pound(s)
What would you like? (small/ medium/ large/ off/ report): large
Please insert coins.
how many large dollars?: 3
how many half dollars?: 2
how many quarters?: 1
how many nickels?: 0
Sorry that's not enough money. Money refunded.
What would you like? (small/ medium/ large/ off/ report): medium
Please insert coins.
how many large dollars?: 2
how many half dollars?: 2
how many quarters?: 1
how many nickels?: 0
Here is \$0.0 in change.
medium sandwich is ready. Bon appetit!
What would you like? (small/ medium/ large/ off/ report): report
Bread: 6 slice(s)
Ham: 8 slice(s)
Cheese: 12 pound(s)
What would you like? (small/ medium/ large/ off/ report): large
Please insert coins.
how many large dollars?: 4
how many half dollars?: 2
how many quarters?: 3
how many nickels?: 0
Here is \$0.25 in change.
large sandwich is ready. Bon appetit!
What would you like? (small/ medium/ large/ off/ report): report
Bread: 0 slice(s)
Ham: 0 slice(s)
Cheese: 0 pound(s)
What would you like? (small/ medium/ large/ off/ report): small
Sorry there is not enough bread.
What would you like? (small/ medium/ large/ off/ report): off

Methods:

`__init__:`

- Attributes:
 - machine_resources : list of resources in sandwich machine
- functionality:
 - Assigns input to the self variable

`check_resources:`

- Attributes:
 - ingredients: can be found from recipes dictionary based on user order.
- functionality:
 - Returns True when order can be made, False if ingredients are insufficient.

`process_coins:`

- functionality:
 - Returns the total calculated from coins inserted.

`transaction_result:`

- Attributes:
 - coins: the output of process_coins() function.
 - cost: can be found from recipes dictionary based on user order.
- functionality:
 - Returns when the payment is accepted, or False if money is insufficient.

`make_sandwich:`

- Attributes:
 - sandwich_size: Based on user input.
 - order_ingredients: can be found from recipes dictionary based on user order.
- functionality:
 - Deduct the required ingredients from the resources (void function, no return).

Code Skeleton and project setup:

- 1- Download the assignments repository, which contains the skeleton code. (You can find the link on Canvas)
- 2- Create a new with virtual environment
- 3- Complete the code based on project requirements.