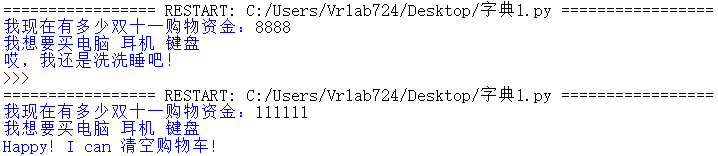
**Homeworks**

**Homework Tasks:**

1. Can I use my pocket money to clear my shopping cart during Double Eleven? Knowing the items you want to buy and the unit price, as well as the available shopping funds in hand, determine whether you can clear the shopping cart.

Tip: Use a dictionary to create a shopping list, wants = {"computer":9999, "headphones":3500, "keyboard":2000}.

Input and output examples:



1. Guess the number game

Your program needs to generate a random number between 1 and 100, and then ask the user: "Please guess a number between 1 and 100." The user has unlimited opportunities to guess this number until they guess correctly. Each time the user guesses, the program needs to tell the user whether the number they guessed is too high or too low until the user guesses correctly. When the user guesses the number correctly, the program should output how many times it took to guess the number correctly.

Require:

1. Use the random module to generate random numbers between 1 and 100.
2. Use the while to let the user guess numbers repeatedly.
3. Use an if statement to determine the relationship between the number guessed by the user and the random number.
4. After the user guesses the number correctly, output how many times it took to guess the number correctly.
5. text adventure game

Your program needs to create a text adventure game in which the player goes on an adventure in a virtual world. Players can choose to move "forward", "left", or "right" to explore different locations. Each location has different events or items, and players need to make choices based on the events. For example, in one location, players may encounter monsters and can choose to "fight" or "escape"; in another location, players may find treasures and can choose to "pick up" or "leave."

Require:

1. Create a virtual world of different places, events, and options.
2. Players start the game at a specific location in the virtual world.
3. Use a while loop to keep the game going until the player chooses to quit.
4. Use if statements to display different events or outcomes based on player choices.
5. In each location, different options are provided to the player, and the player can choose one of them to decide the next action.
6. The game should have a victory condition, such as finding the treasure or defeating the final boss.
7. Enter an integer less than 1000 and factor it

Output: For example 99 = 3\*3\*11

Analysis: To decompose the correct input n into prime factors, you should first find a minimum prime number k, and then complete the following steps:

(1) If this prime number is exactly equal to n, it means that the process of decomposing prime factors has ended, just print it out.

(2) If n!=k, but n is divisible by k, the value of k should be printed out, and the quotient of n divided by k is used as the new positive integer n, and the first step is repeated.

(3) If n is not divisible by k, use k+1 as the value of k and repeat the first step.

**Submission Content:**

1. Submit .txt or .py files containing the source code (one file for each task).
2. Submit screenshots of the program running (one screenshot for each task) or a video demo for each task.

**Grading Criteria:**

1. Functionality Completeness (50%): Whether all tasks are completed as per the requirements.
2. Code Quality (30%): Readability, modularity, appropriate use of comments, independent completion, etc.
3. Screenshots of Execution (10%): Clear and valid screenshots demonstrating the proper functioning of the program.
4. Other Criteria (10%): Timely submission, adherence to guidelines, etc.

**Note: Please ensure that the submitted code and screenshots are clear and visible. Grading will be based on the completeness and quality of the assignment.**