

Project Report

Spring 2025

Name: Rakshith

Project: Making Change

Note: As you write each section, try to be as clear and detailed as possible. Your goal is to communicate your thought process and work clearly. Don't worry if you faced challenges or made mistakes; discussing these is a valuable part of learning and shows your problem-solving skills! Remember, there's no single 'right' way to do these tasks, so be creative and honest in your responses.

Problem Statement (2-3 Paragraphs):

The goal was to build a change calculator that used the fewest number of coins. The program asks for a number from 0 to 99 and it gets a way to return the fewest number of coins back to the user. After printing, it asks the user if they want to do another computation and only continues if they enter y.

The inputs are the number of cents and the output is the original amount and count of number of all the coins. The instructions says to assume valid input, so no try and except conditions are used. I added the .lower() function so it takes both y and Y inputs to rerun the function. I would add checks to limit the range later.

Design (1-3 Paragraphs):

I used integer division and modulo. The order goes from the biggest to smallest amount of coins' worth. It divides to get the coin count, then uses modulo to carry a remainder and continues to the next. I think it was pretty effective because it's short and simple to read. If I changed it, I would add the coin division into a function. I could also add it so it adds dollars and other numbers without changing the main flow.

Testing (1-2 Paragraphs + screenshots of 3 test cases):

I tested by running the program from the terminal and entering differentiating values. I used normal values like 37 and 99 like the example given and also added stuff like 0 for no coins. I tested both y and Y to continue the loop and n to stop it. Results matched in all cases.

```
How much money do you want to get change for? (0-99 cents): 37
For an amount of 37 cents, we will need:
```

```
1 Quarters
1 Dimes
0 Nickels
2 Pennies
```

```
Go again? (y/n) Y
```

```
How much money do you want to get change for? (0-99 cents): 99
For an amount of 99 cents, we will need:
```

```
3 Quarters
2 Dimes
0 Nickels
4 Pennies
```

```
Go again? (y/n) n
```

```
How much money do you want to get change for? (0-99 cents): 0
For an amount of 0 cents, we will need:
```

```
0 Quarters
0 Dimes
0 Nickels
0 Pennies
```

```
Go again? (y/n) y
```

```
How much money do you want to get change for? (0-99 cents): 99
For an amount of 99 cents, we will need:
```

```
3 Quarters
2 Dimes
0 Nickels
4 Pennies
```

```
How much money do you want to get change for? (0-99 cents): 50
For an amount of 50 cents, we will need:
```

```
2 Quarters
0 Dimes
0 Nickels
0 Pennies
```

```
Go again? (y/n) n
```

Conclusion (1 paragraph)

The project met the given goals. I learned to use floor division and modulo together to find the easiest way to calculate coin change. The program is simple, short, and easy to understand. I will add a way to reuse the math by putting it all into a function and also add a way to include other types of currency.