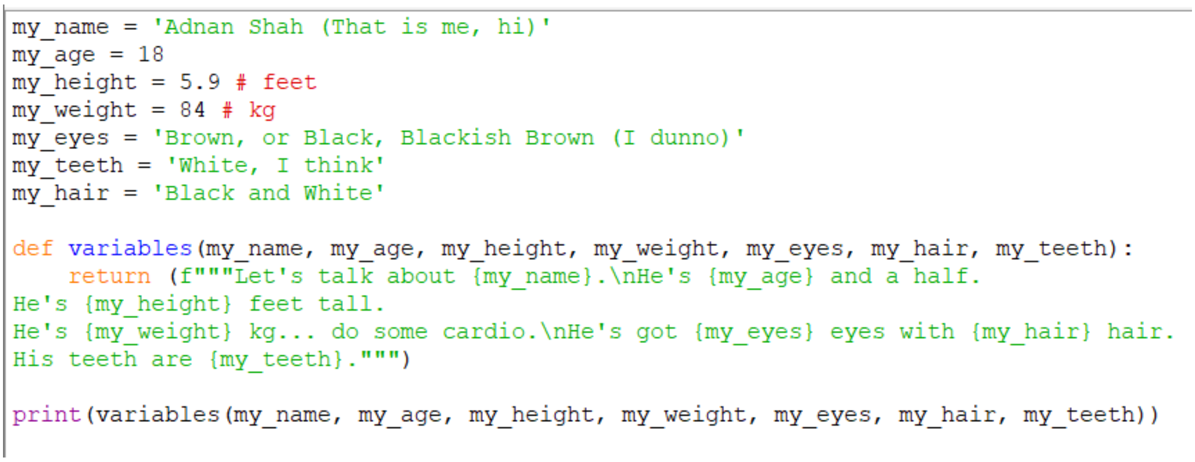
Screenshots

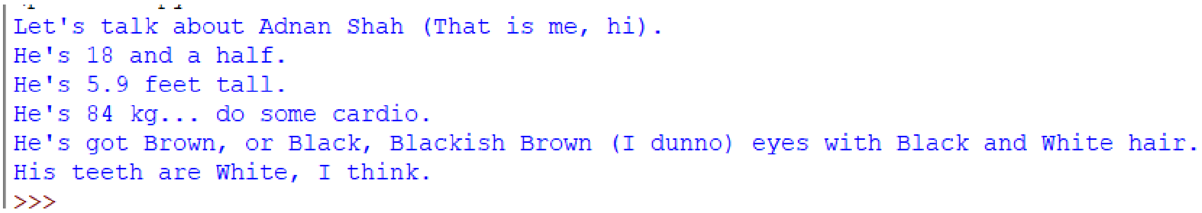
Problem 1

Code:



```
my_name = 'Adnan Shah (That is me, hi)'
my_age = 18
my_height = 5.9 # feet
my_weight = 84 # kg
my_eyes = 'Brown, or Black, Blackish Brown (I dunno)'
my_teeth = 'White, I think'
my_hair = 'Black and White'

def variables(my_name, my_age, my_height, my_weight, my_eyes, my_hair, my_teeth):
    return (f"""Let's talk about {my_name}.\nHe's {my_age} and a half.
He's {my_height} feet tall.
He's {my_weight} kg... do some cardio.\nHe's got {my_eyes} eyes with {my_hair} hair.
His teeth are {my_teeth}.""")

print(variables(my_name, my_age, my_height, my_weight, my_eyes, my_hair, my_teeth))
```

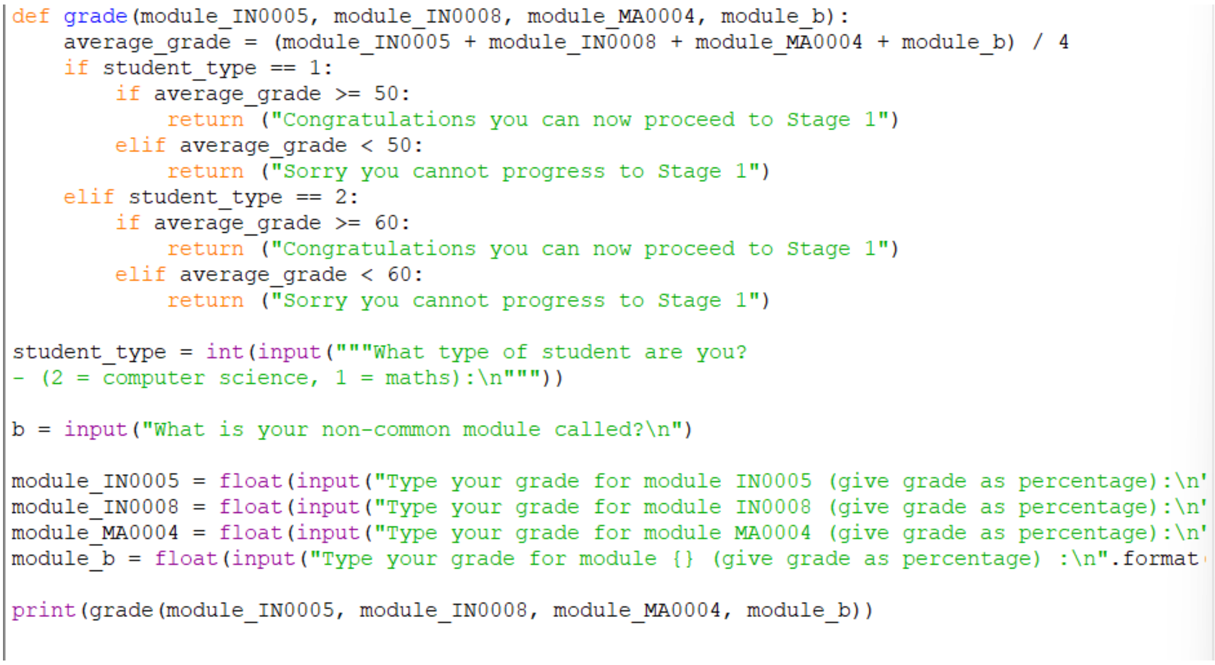
Output:



```
Let's talk about Adnan Shah (That is me, hi).
He's 18 and a half.
He's 5.9 feet tall.
He's 84 kg... do some cardio.
He's got Brown, or Black, Blackish Brown (I dunno) eyes with Black and White hair.
His teeth are White, I think.
>>>
```

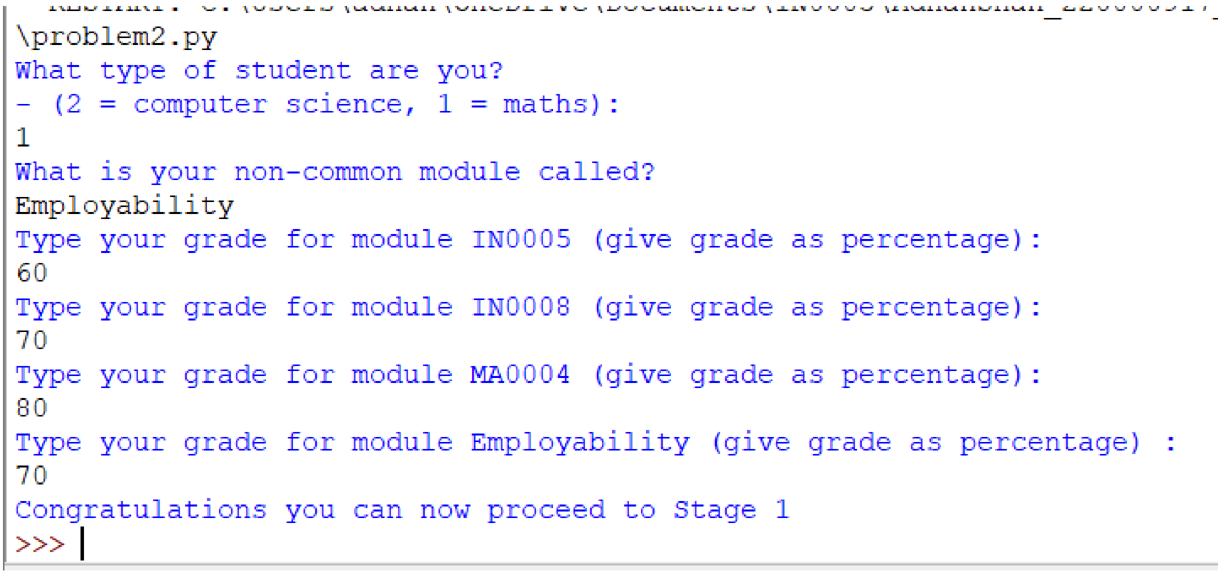
Problem 2

Code:



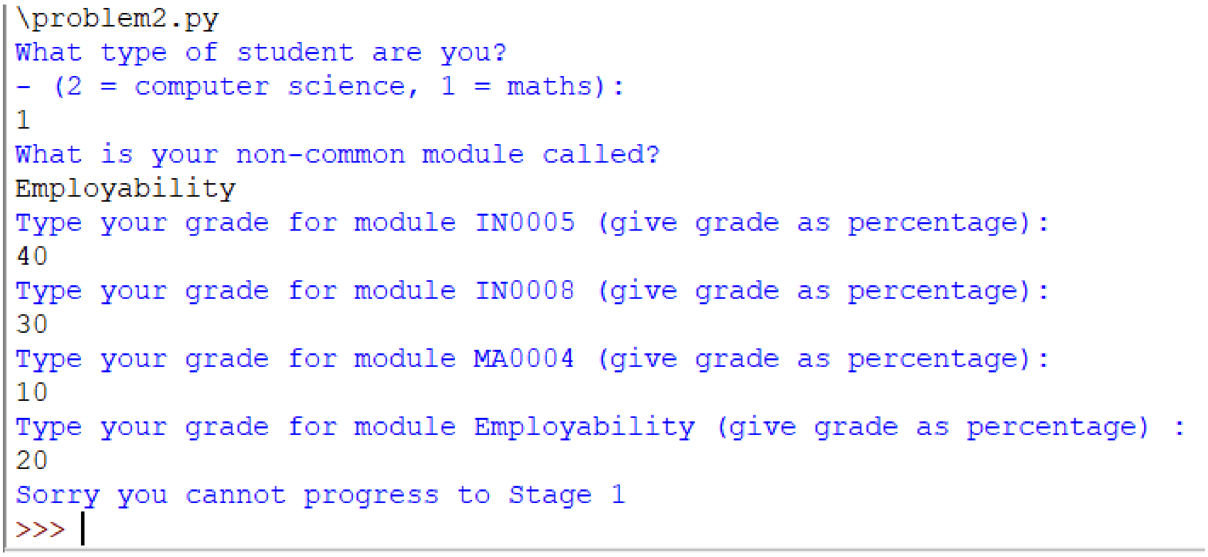
```
def grade(module_IN0005, module_IN0008, module_MA0004, module_b):
    average_grade = (module_IN0005 + module_IN0008 + module_MA0004 + module_b) / 4
    if student_type == 1:
        if average_grade >= 50:
            return ("Congratulations you can now proceed to Stage 1")
        elif average_grade < 50:
            return ("Sorry you cannot progress to Stage 1")
    elif student_type == 2:
        if average_grade >= 60:
            return ("Congratulations you can now proceed to Stage 1")
        elif average_grade < 60:
            return ("Sorry you cannot progress to Stage 1")

student_type = int(input("""What type of student are you?
- (2 = computer science, 1 = maths):\n"""))

b = input("What is your non-common module called?\n")

module_IN0005 = float(input("Type your grade for module IN0005 (give grade as percentage):\n'
module_IN0008 = float(input("Type your grade for module IN0008 (give grade as percentage):\n'
module_MA0004 = float(input("Type your grade for module MA0004 (give grade as percentage):\n'
module_b = float(input("Type your grade for module {} (give grade as percentage) :\n".format

print(grade(module_IN0005, module_IN0008, module_MA0004, module_b))
```

Output 1 – Maths – Proceed to the next stage:



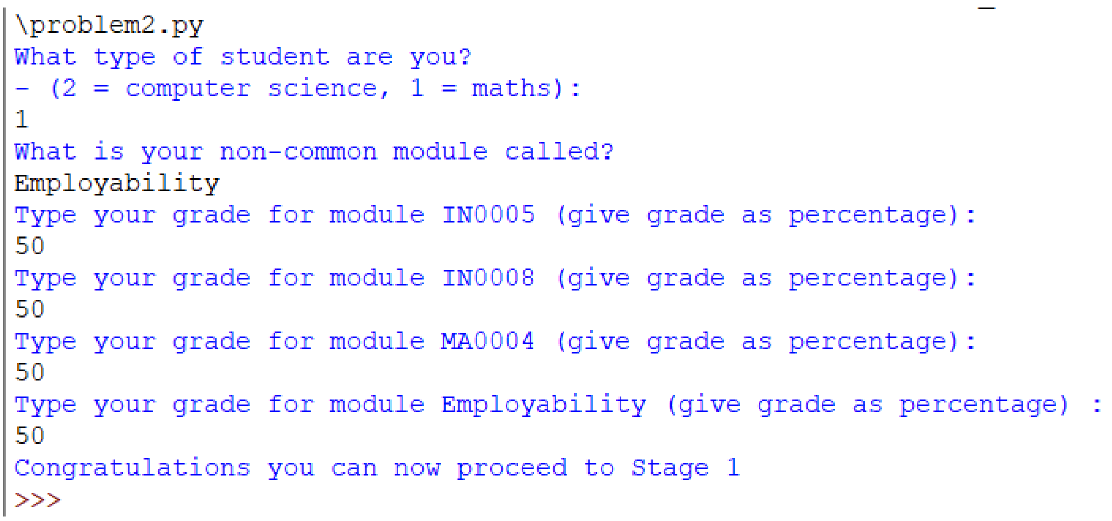
```
\problem2.py
What type of student are you?
- (2 = computer science, 1 = maths):
1
What is your non-common module called?
Employability
Type your grade for module IN0005 (give grade as percentage):
60
Type your grade for module IN0008 (give grade as percentage):
70
Type your grade for module MA0004 (give grade as percentage):
80
Type your grade for module Employability (give grade as percentage) :
70
Congratulations you can now proceed to Stage 1
>>> 
```

Output 2 – Maths – Cannot Proceed:



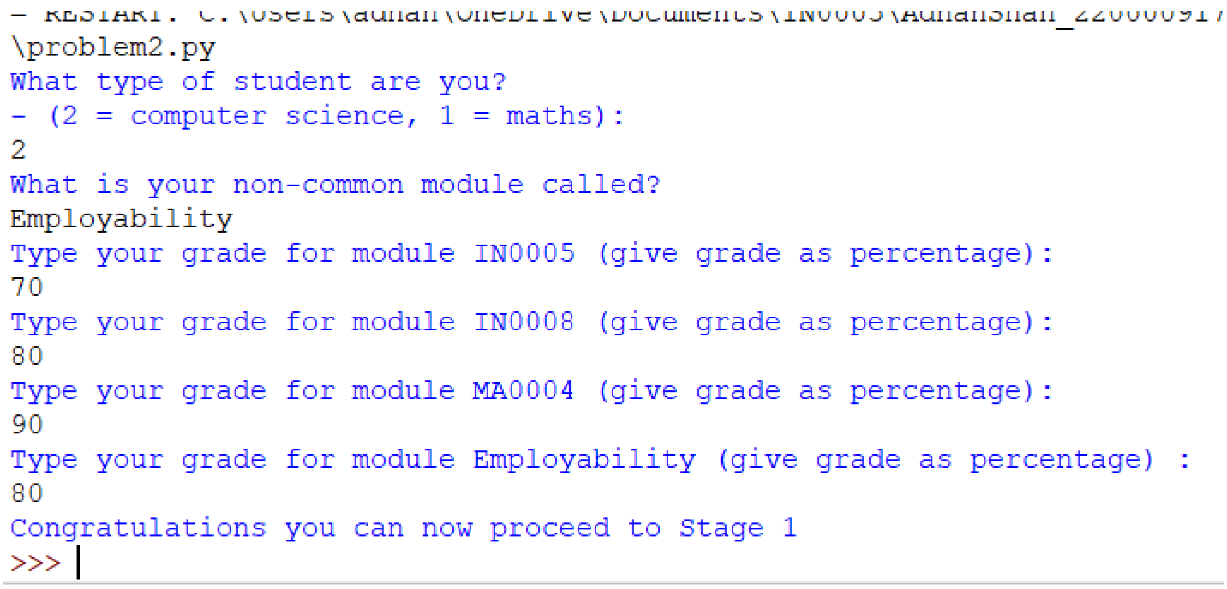
```
\problem2.py
What type of student are you?
- (2 = computer science, 1 = maths):
1
What is your non-common module called?
Employability
Type your grade for module IN0005 (give grade as percentage):
40
Type your grade for module IN0008 (give grade as percentage):
30
Type your grade for module MA0004 (give grade as percentage):
10
Type your grade for module Employability (give grade as percentage) :
20
Sorry you cannot progress to Stage 1
>>> 
```

Output 3 – Maths – Exactly 50%:



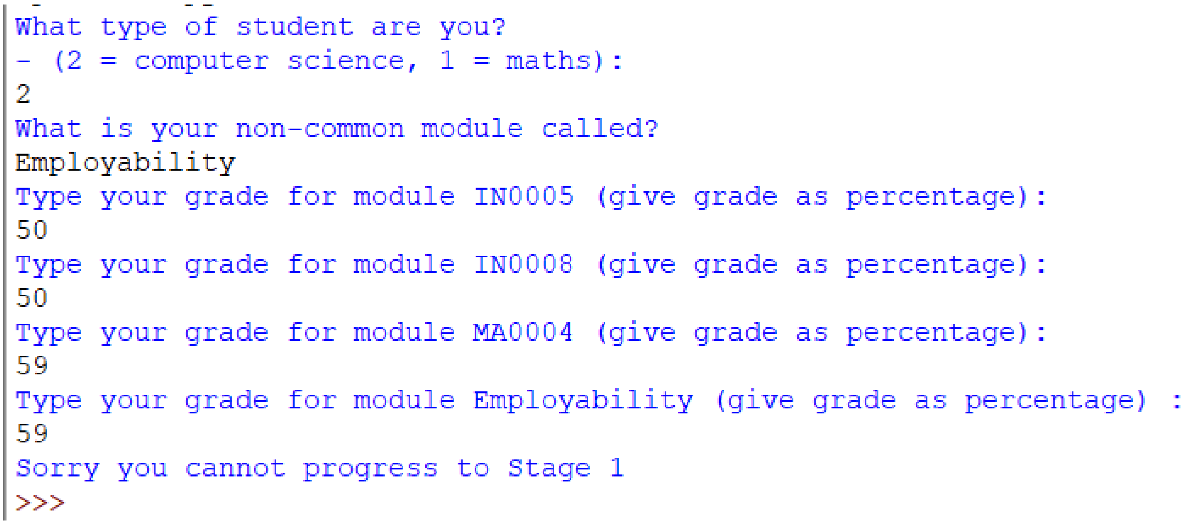
```
\problem2.py
What type of student are you?
- (2 = computer science, 1 = maths):
1
What is your non-common module called?
Employability
Type your grade for module IN0005 (give grade as percentage):
50
Type your grade for module IN0008 (give grade as percentage):
50
Type your grade for module MA0004 (give grade as percentage):
50
Type your grade for module Employability (give grade as percentage) :
50
Congratulations you can now proceed to Stage 1
>>>
```

Output 1 – Computer Science - Proceed to the next stage:



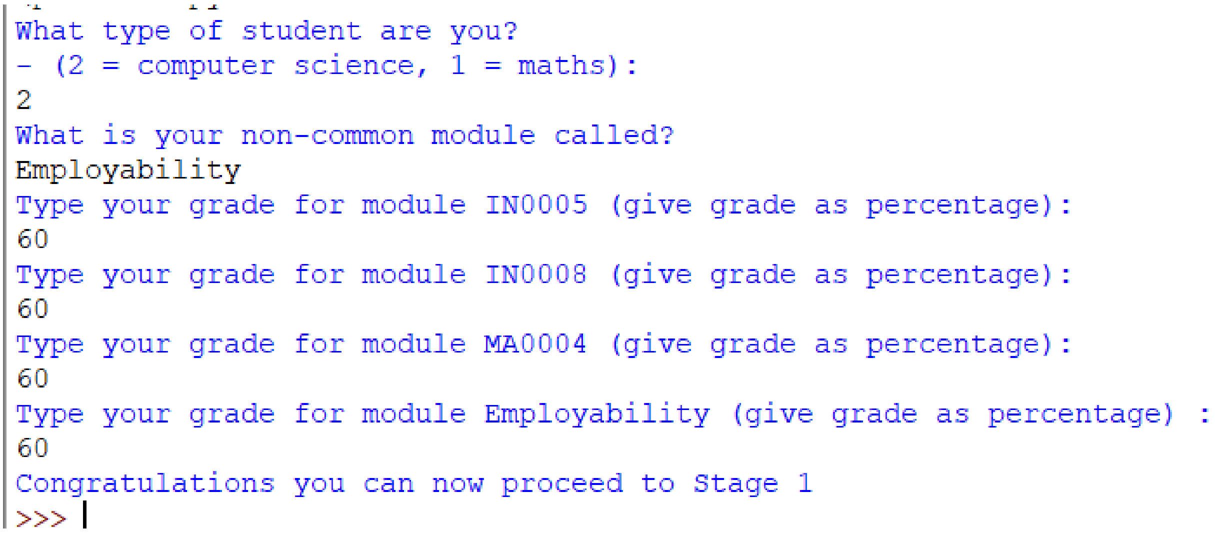
```
= RESTART: C:\Users\adnan\OneDrive\Documents\IN0005\AdnanShah_220000917
\problem2.py
What type of student are you?
- (2 = computer science, 1 = maths):
2
What is your non-common module called?
Employability
Type your grade for module IN0005 (give grade as percentage):
70
Type your grade for module IN0008 (give grade as percentage):
80
Type your grade for module MA0004 (give grade as percentage):
90
Type your grade for module Employability (give grade as percentage) :
80
Congratulations you can now proceed to Stage 1
>>> 
```

Output 2 – Computer Science - Cannot Proceed:



```
What type of student are you?
- (2 = computer science, 1 = maths):
2
What is your non-common module called?
Employability
Type your grade for module IN0005 (give grade as percentage):
50
Type your grade for module IN0008 (give grade as percentage):
50
Type your grade for module MA0004 (give grade as percentage):
59
Type your grade for module Employability (give grade as percentage) :
59
Sorry you cannot progress to Stage 1
>>>
```

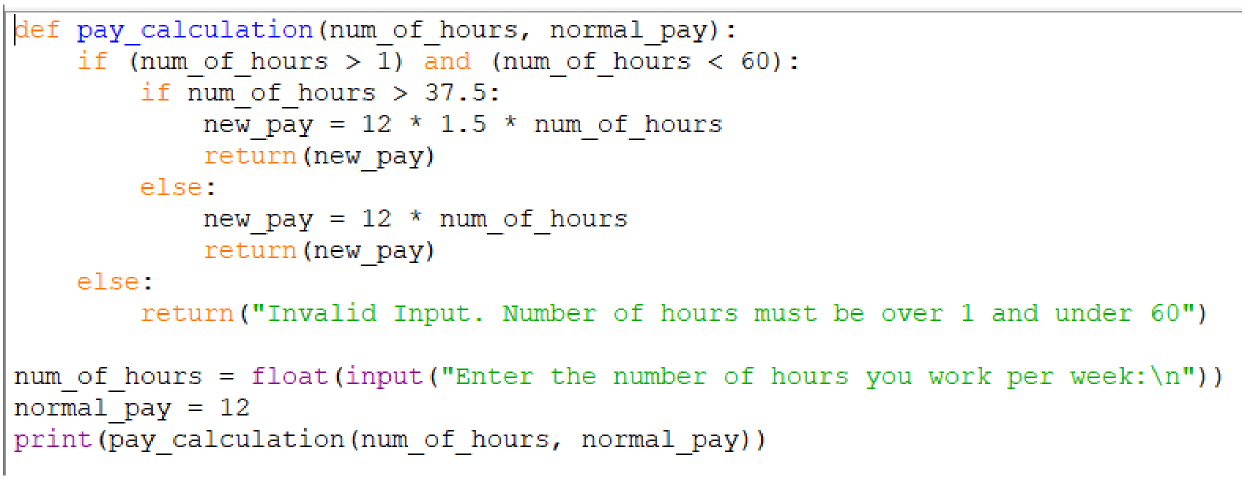
Output 3 – Computer Science – Exactly 60%:



```
What type of student are you?
- (2 = computer science, 1 = maths):
2
What is your non-common module called?
Employability
Type your grade for module IN0005 (give grade as percentage):
60
Type your grade for module IN0008 (give grade as percentage):
60
Type your grade for module MA0004 (give grade as percentage):
60
Type your grade for module Employability (give grade as percentage) :
60
Congratulations you can now proceed to Stage 1
>>> 
```

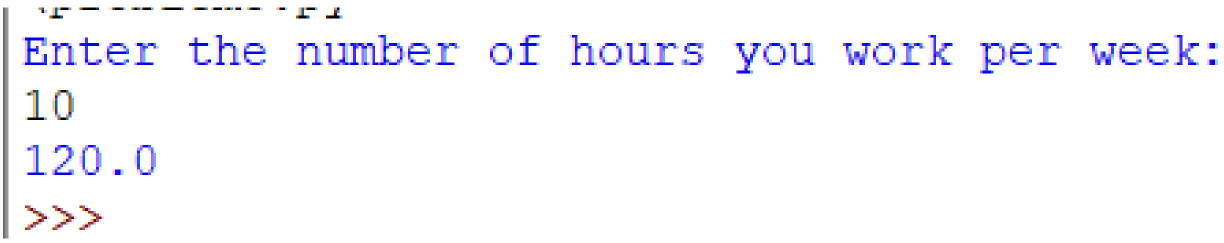
Problem 3:

Code:



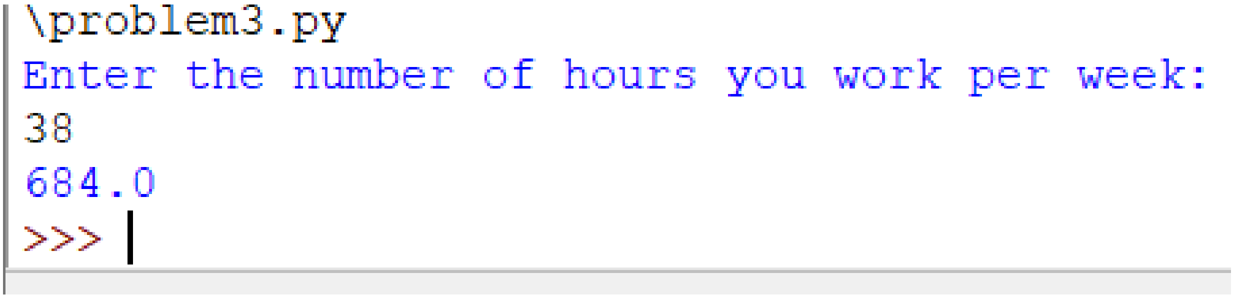
```
def pay_calculation(num_of_hours, normal_pay):
    if (num_of_hours > 1) and (num_of_hours < 60):
        if num_of_hours > 37.5:
            new_pay = 12 * 1.5 * num_of_hours
            return(new_pay)
        else:
            new_pay = 12 * num_of_hours
            return(new_pay)
    else:
        return("Invalid Input. Number of hours must be over 1 and under 60")

num_of_hours = float(input("Enter the number of hours you work per week:\n"))
normal_pay = 12
print(pay_calculation(num_of_hours, normal_pay))
```

Output 1 – number of hours between 1 and 60 (non-inclusive), below 37.5:



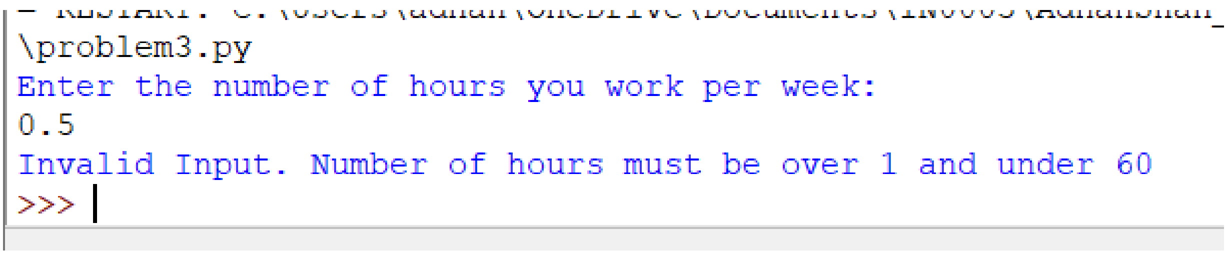
```
Enter the number of hours you work per week:
10
120.0
>>>
```

Output 2 - number of hours between 1 and 60 (non-inclusive), above 37.5:

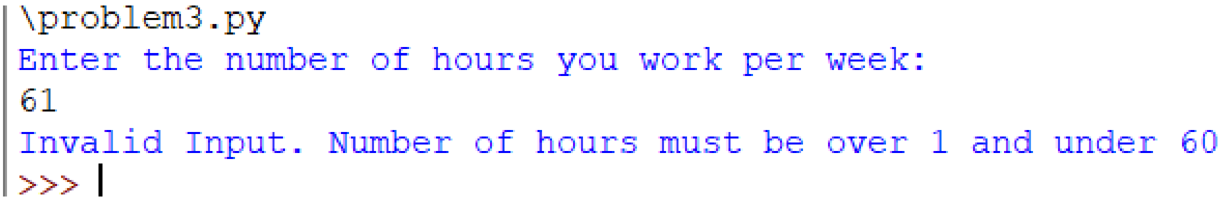


```
\problem3.py
Enter the number of hours you work per week:
38
684.0
>>>
```

Output 3 – number of hours below 1 and above 60:



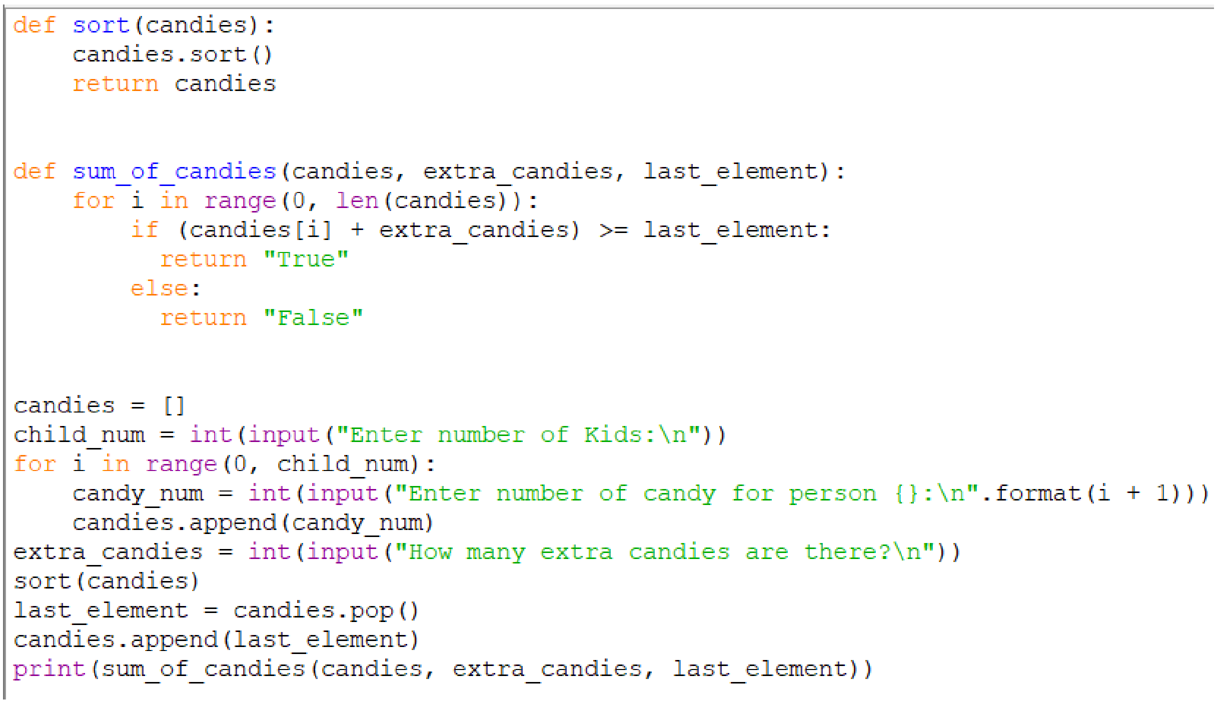
```
\problem3.py
Enter the number of hours you work per week:
0.5
Invalid Input. Number of hours must be over 1 and under 60
>>>
```



```
\problem3.py
Enter the number of hours you work per week:
61
Invalid Input. Number of hours must be over 1 and under 60
>>>
```

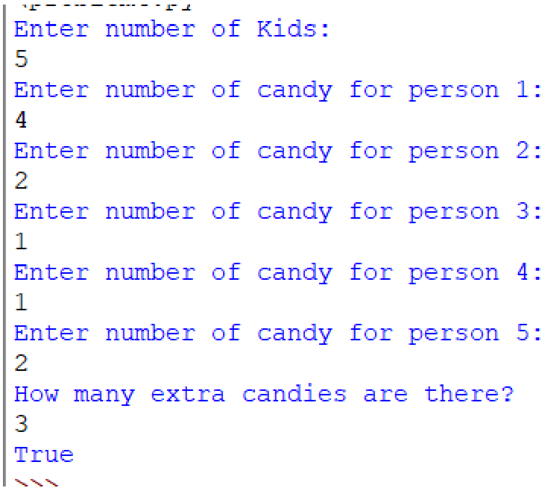
Problem 4

Code:

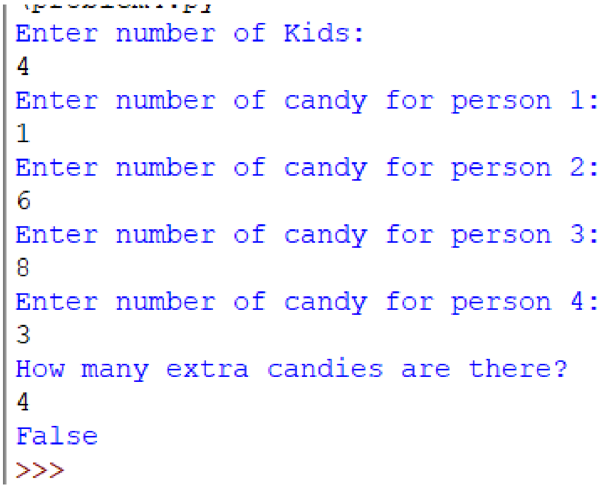


```
def sort(candies):
    candies.sort()
    return candies


def sum_of_candies(candies, extra_candies, last_element):
    for i in range(0, len(candies)):
        if (candies[i] + extra_candies) >= last_element:
          return "True"
        else:
          return "False"


candies = []
child_num = int(input("Enter number of Kids:\n"))
for i in range(0, child_num):
    candy_num = int(input("Enter number of candy for person {}:\n".format(i + 1)))
    candies.append(candy_num)
extra_candies = int(input("How many extra candies are there?\n"))
sort(candies)
last_element = candies.pop()
candies.append(last_element)
print(sum_of_candies(candies, extra_candies, last_element))
```

Output:



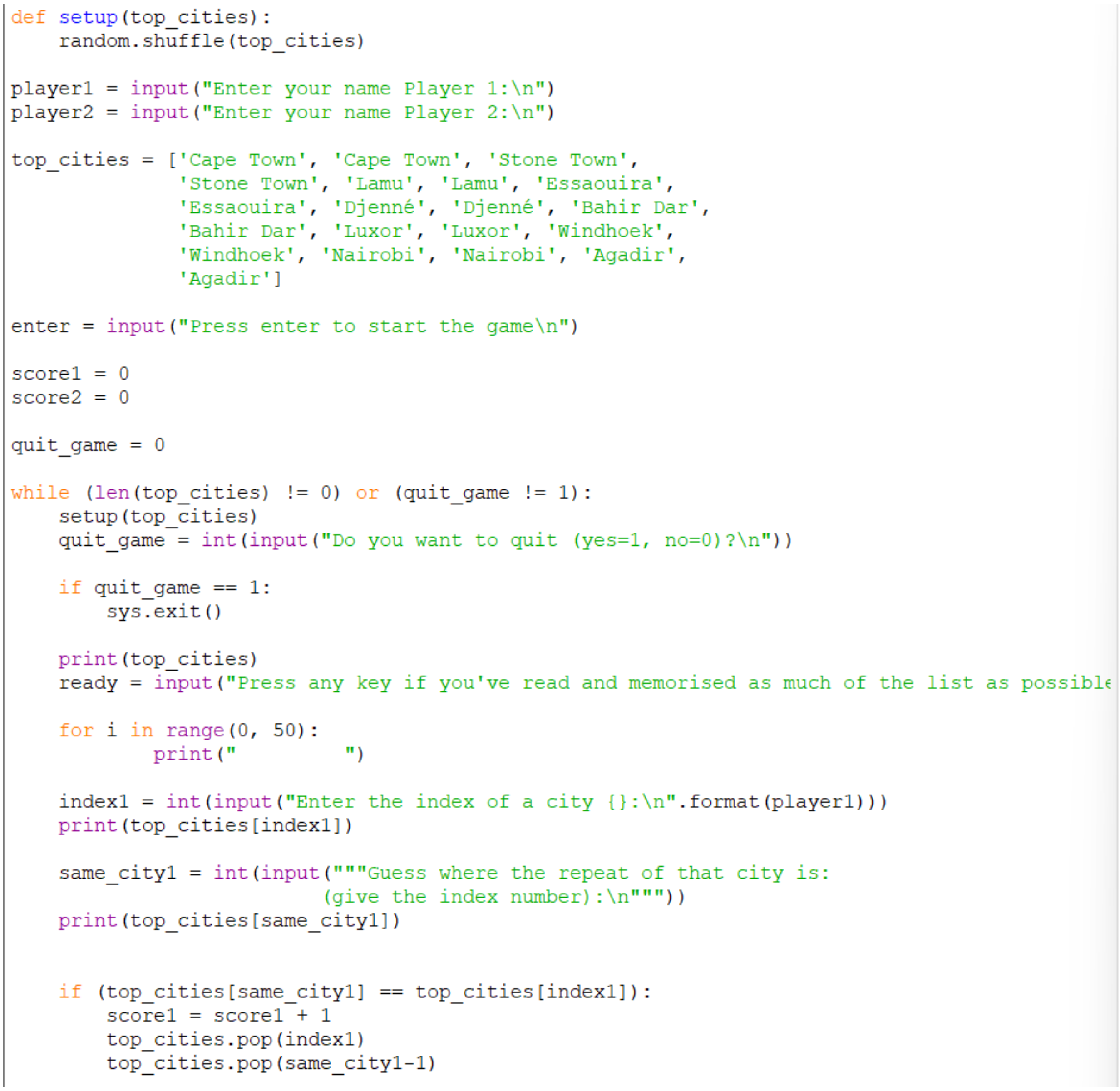
```
Enter number of Kids:
5
Enter number of candy for person 1:
4
Enter number of candy for person 2:
2
Enter number of candy for person 3:
1
Enter number of candy for person 4:
1
Enter number of candy for person 5:
2
How many extra candies are there?
3
True
>>>
```



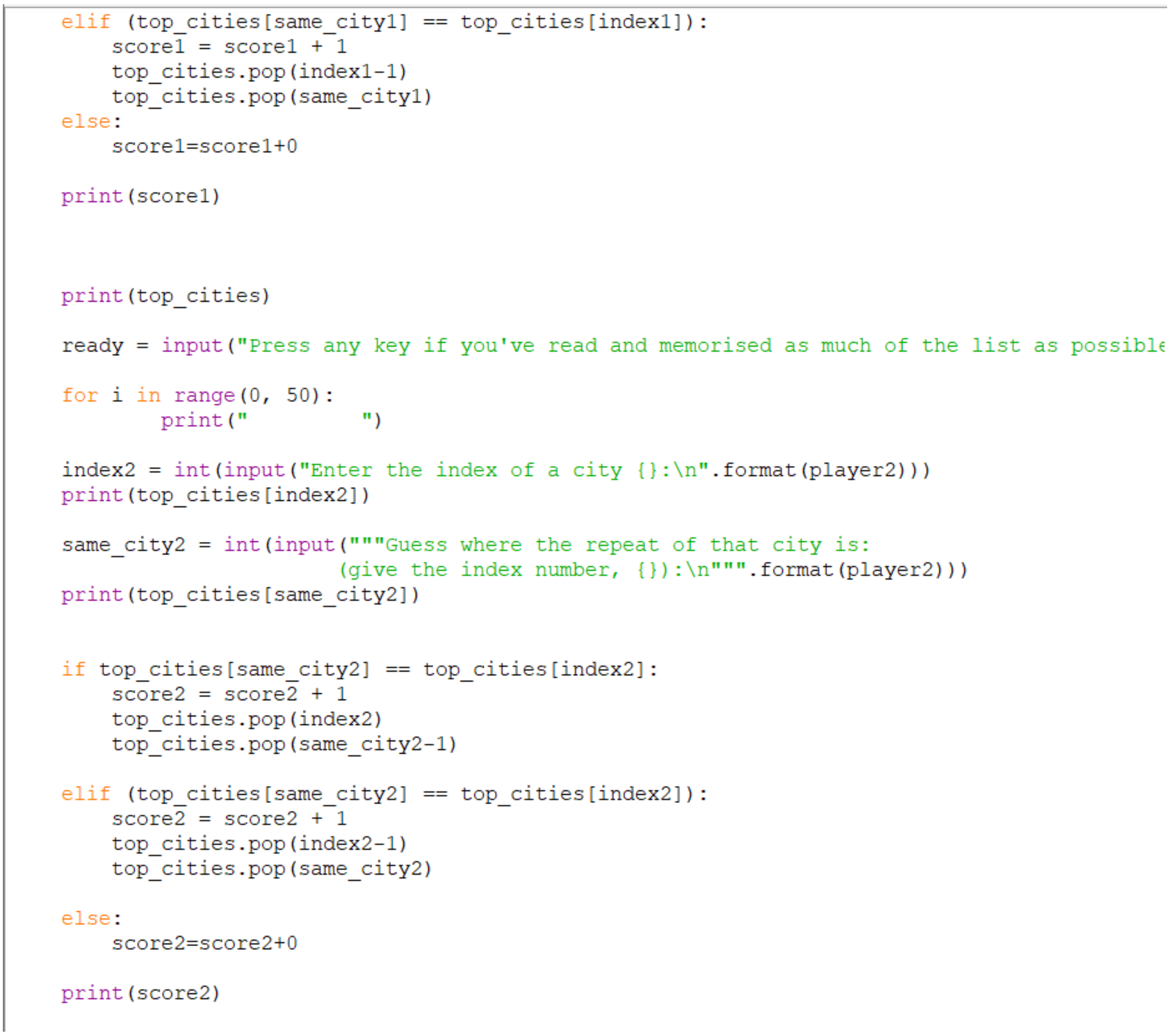
```
Enter number of Kids:
4
Enter number of candy for person 1:
1
Enter number of candy for person 2:
6
Enter number of candy for person 3:
8
Enter number of candy for person 4:
3
How many extra candies are there?
4
False
>>>
```

Problem 5:

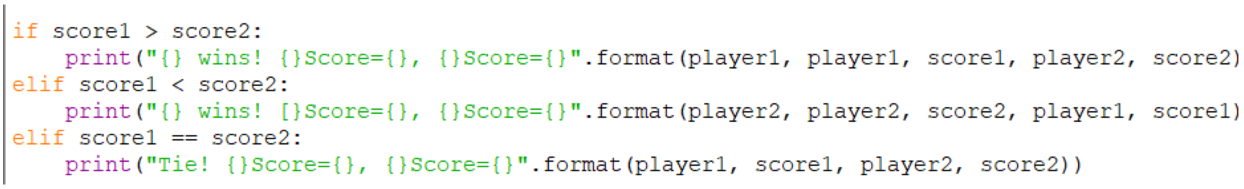
Code:



```
def setup(top_cities):
    random.shuffle(top_cities)

player1 = input("Enter your name Player 1:\n")
player2 = input("Enter your name Player 2:\n")

top_cities = ['Cape Town', 'Cape Town', 'Stone Town',
              'Stone Town', 'Lamu', 'Lamu', 'Essaouira',
              'Essaouira', 'Djenné', 'Djenné', 'Bahir Dar',
              'Bahir Dar', 'Luxor', 'Luxor', 'Windhoek',
              'Windhoek', 'Nairobi', 'Nairobi', 'Agadir',
              'Agadir']

enter = input("Press enter to start the game\n")

score1 = 0
score2 = 0

quit_game = 0

while (len(top_cities) != 0) or (quit_game != 1):
    setup(top_cities)
    quit_game = int(input("Do you want to quit (yes=1, no=0)?\n"))

    if quit_game == 1:
        sys.exit()

    print(top_cities)
    ready = input("Press any key if you've read and memorised as much of the list as possible

    for i in range(0, 50):
            print("         ")

    index1 = int(input("Enter the index of a city {}:\n".format(player1)))
    print(top_cities[index1])

    same_city1 = int(input("""Guess where the repeat of that city is:
                        (give the index number):\n"""))
    print(top_cities[same_city1])


    if (top_cities[same_city1] == top_cities[index1]):
        score1 = score1 + 1
        top_cities.pop(index1)
        top_cities.pop(same_city1-1)
```

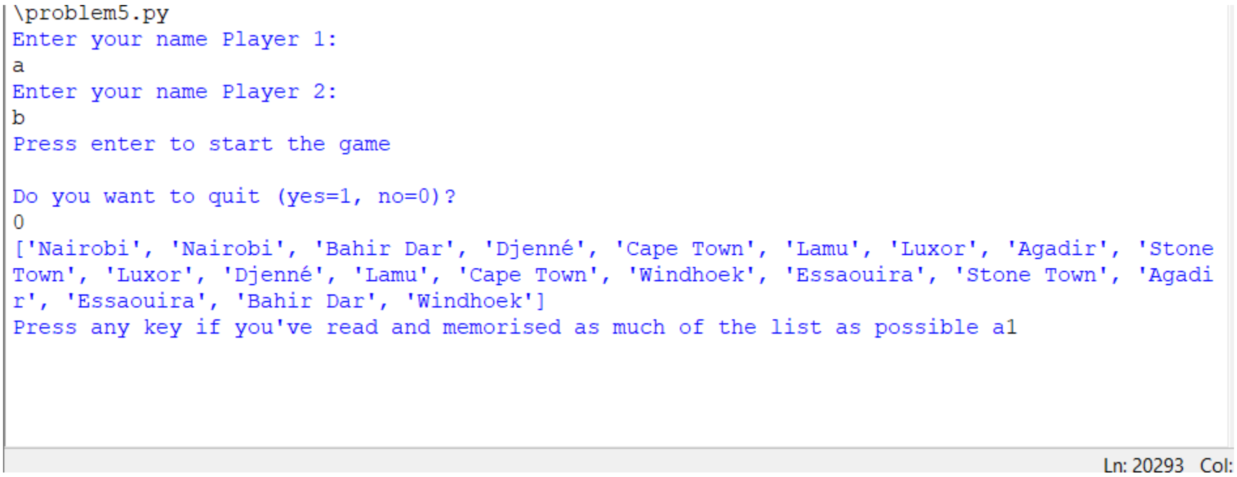


```
    elif (top_cities[same_city1] == top_cities[index1]):
        score1 = score1 + 1
        top_cities.pop(index1-1)
        top_cities.pop(same_city1)
    else:
        score1=score1+0

    print(score1)



    print(top_cities)

    ready = input("Press any key if you've read and memorised as much of the list as possible

    for i in range(0, 50):
            print("         ")

    index2 = int(input("Enter the index of a city {}:\n".format(player2)))
    print(top_cities[index2])

    same_city2 = int(input("""Guess where the repeat of that city is:
                          (give the index number, {}):\n""".format(player2)))
    print(top_cities[same_city2])


    if top_cities[same_city2] == top_cities[index2]:
        score2 = score2 + 1
        top_cities.pop(index2)
        top_cities.pop(same_city2-1)

    elif (top_cities[same_city2] == top_cities[index2]):
        score2 = score2 + 1
        top_cities.pop(index2-1)
        top_cities.pop(same_city2)

    else:
        score2=score2+0

    print(score2)
```

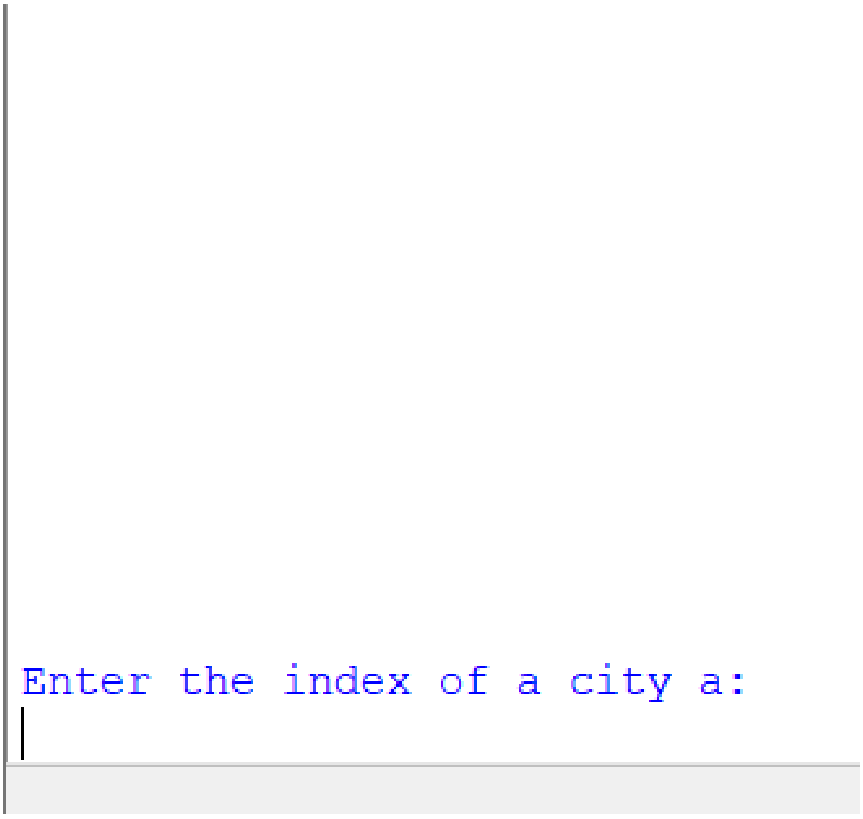


```
if score1 > score2:
    print("{} wins! {}Score={}, {}Score={}".format(player1, player1, score1, player2, score2)
elif score1 < score2:
    print("{} wins! {}Score={}, {}Score={}".format(player2, player2, score2, player1, score1)
elif score1 == score2:
    print("Tie! {}Score={}, {}Score={}".format(player1, score1, player2, score2))
```

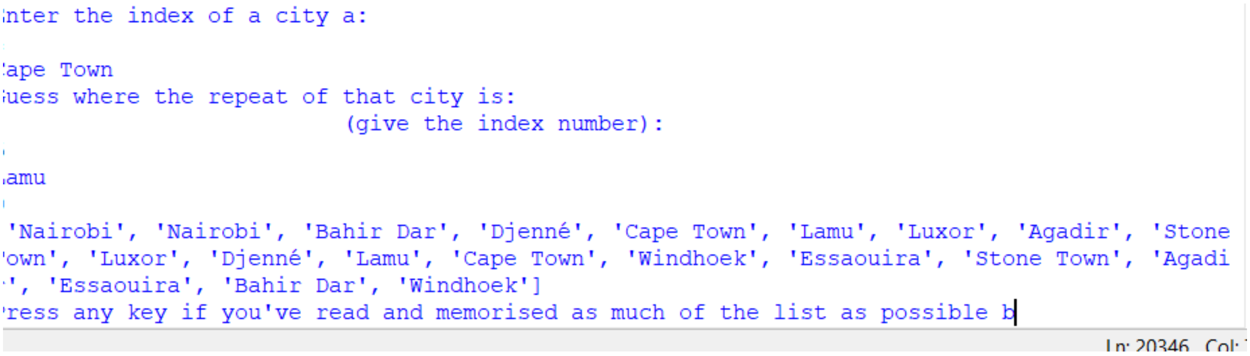
Output:



```
\problem5.py
Enter your name Player 1:
a
Enter your name Player 2:
b
Press enter to start the game

Do you want to quit (yes=1, no=0)?
0
['Nairobi', 'Nairobi', 'Bahir Dar', 'Djenné', 'Cape Town', 'Lamu', 'Luxor', 'Agadir', 'Stone
Town', 'Luxor', 'Djenné', 'Lamu', 'Cape Town', 'Windhoek', 'Essaouira', 'Stone Town', 'Agadi
r', 'Essaouira', 'Bahir Dar', 'Windhoek']
Press any key if you've read and memorised as much of the list as possible a1
```

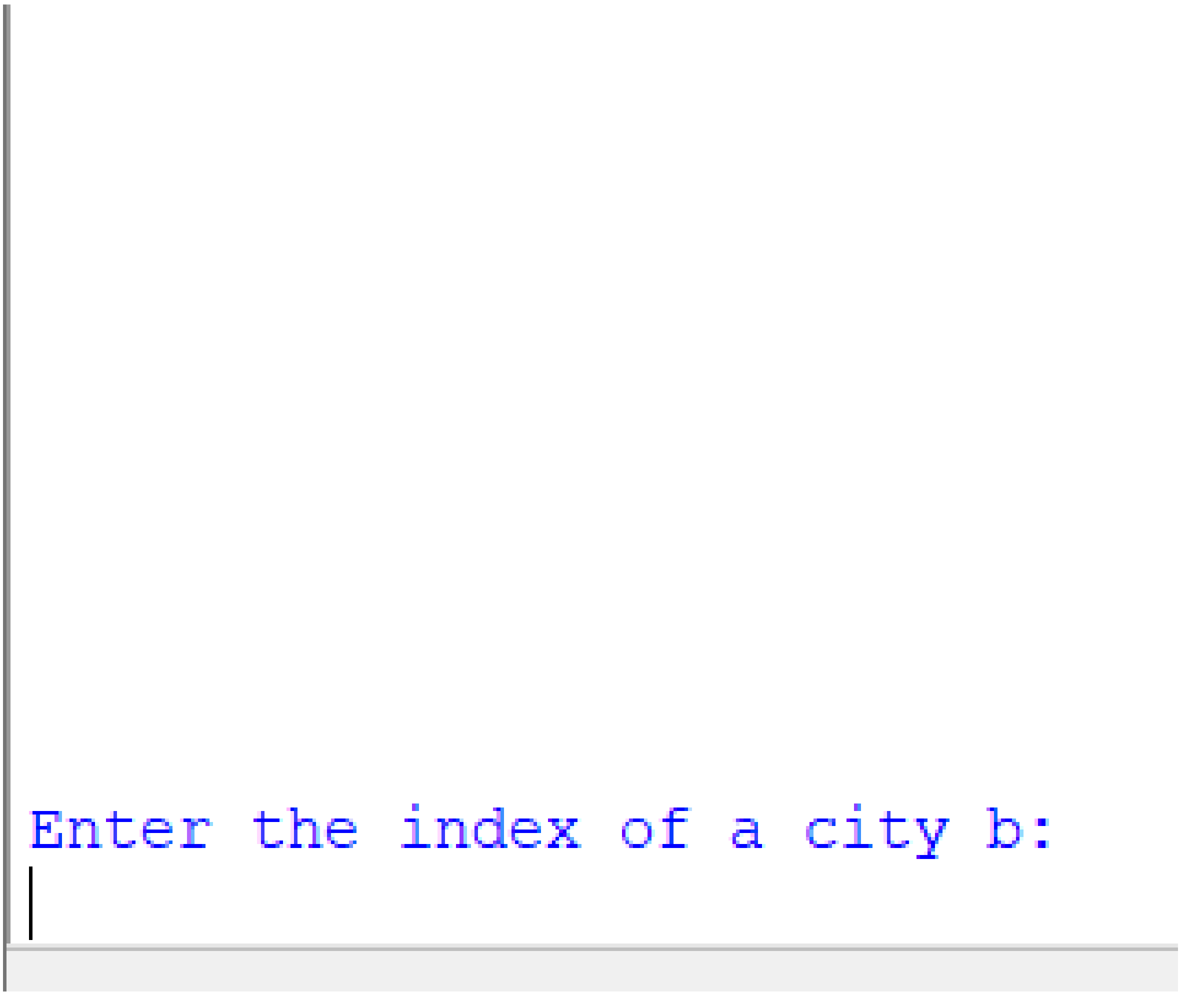


```
Enter the index of a city a:
```

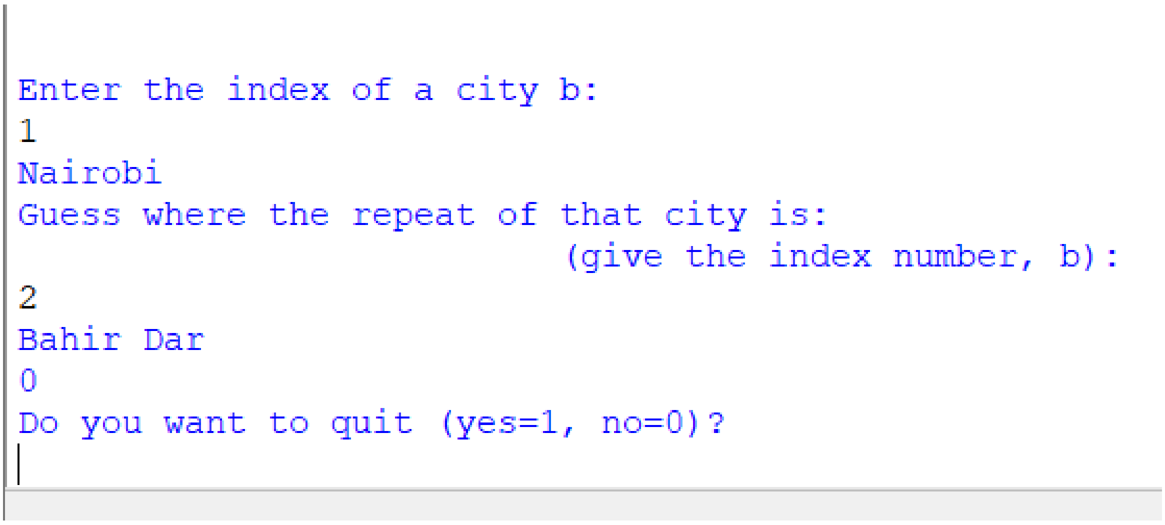


```
inter the index of a city a:

ape Town
uess where the repeat of that city is:
                          (give the index number):

amu

'Nairobi', 'Nairobi', 'Bahir Dar', 'Djenné', 'Cape Town', 'Lamu', 'Luxor', 'Agadir', 'Stone
'own', 'Luxor', 'Djenné', 'Lamu', 'Cape Town', 'Windhoek', 'Essaouira', 'Stone Town', 'Agadi
', 'Essaouira', 'Bahir Dar', 'Windhoek']
'ress any key if you've read and memorised as much of the list as possible b
```

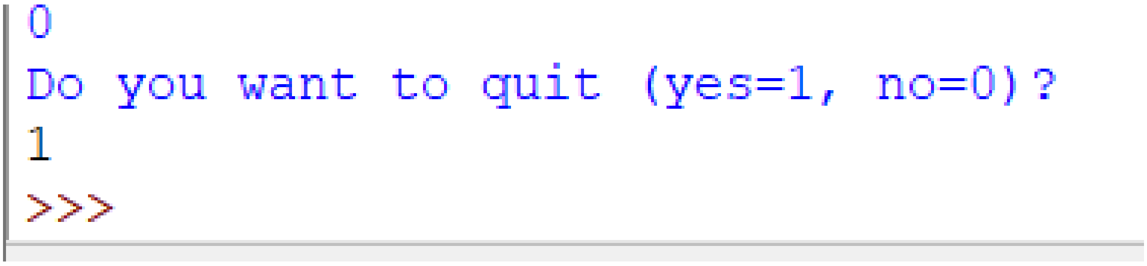
```
Enter the index of a city a:
```



```
Enter the index of a city b:
```



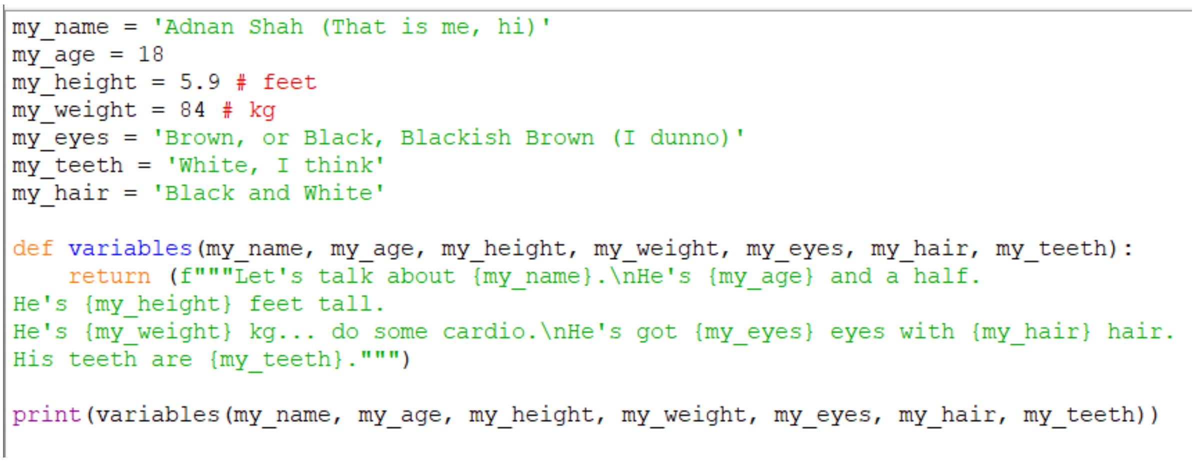
```
Enter the index of a city b:
1
Nairobi
Guess where the repeat of that city is:
                          (give the index number, b):
2
Bahir Dar
0
Do you want to quit (yes=1, no=0)?
```



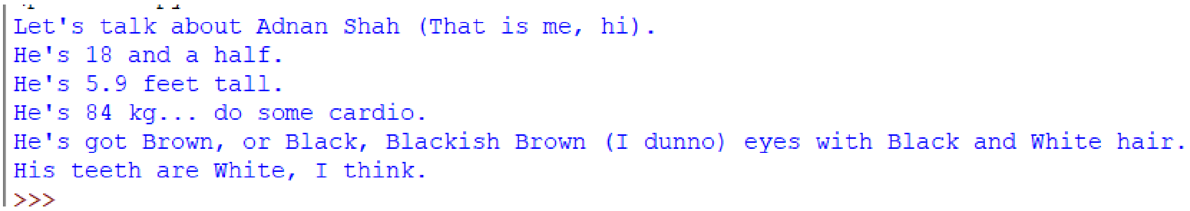
```
0
Do you want to quit (yes=1, no=0)?
1
>>>
```

Problem 6 - Functions:

1) Problem 1:

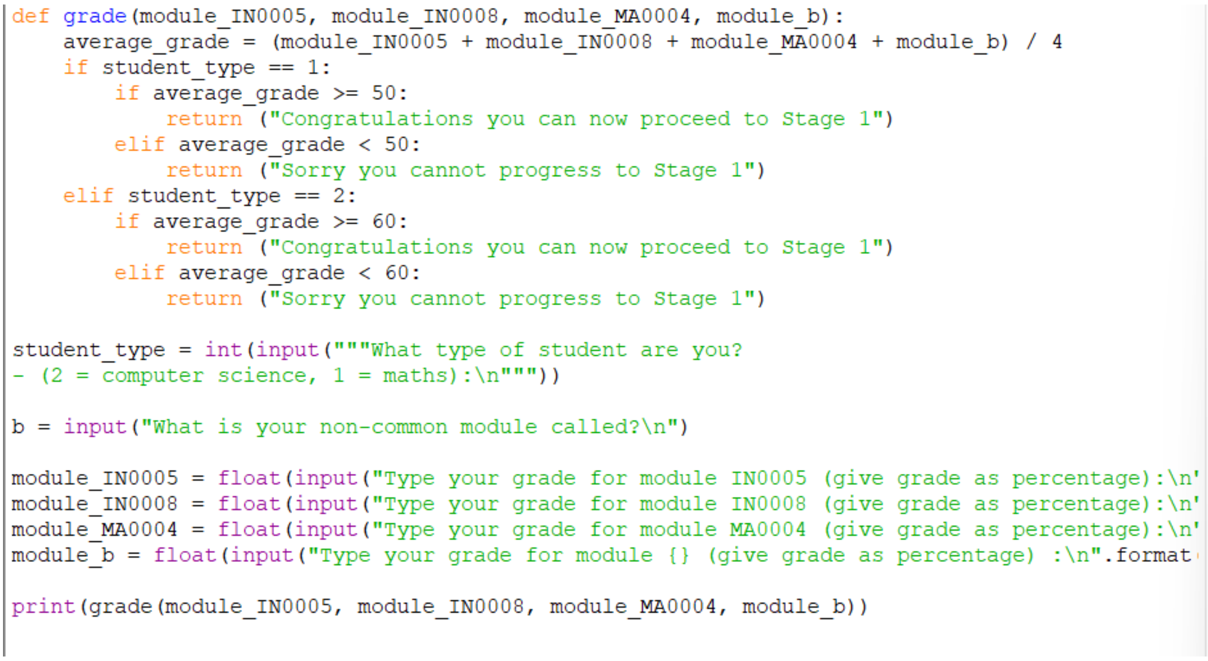


```
my_name = 'Adnan Shah (That is me, hi)'
my_age = 18
my_height = 5.9 # feet
my_weight = 84 # kg
my_eyes = 'Brown, or Black, Blackish Brown (I dunno)'
my_teeth = 'White, I think'
my_hair = 'Black and White'

def variables(my_name, my_age, my_height, my_weight, my_eyes, my_hair, my_teeth):
    return (f"""Let's talk about {my_name}.\nHe's {my_age} and a half.
He's {my_height} feet tall.
He's {my_weight} kg... do some cardio.\nHe's got {my_eyes} eyes with {my_hair} hair.
His teeth are {my_teeth}.""")

print(variables(my_name, my_age, my_height, my_weight, my_eyes, my_hair, my_teeth))
```

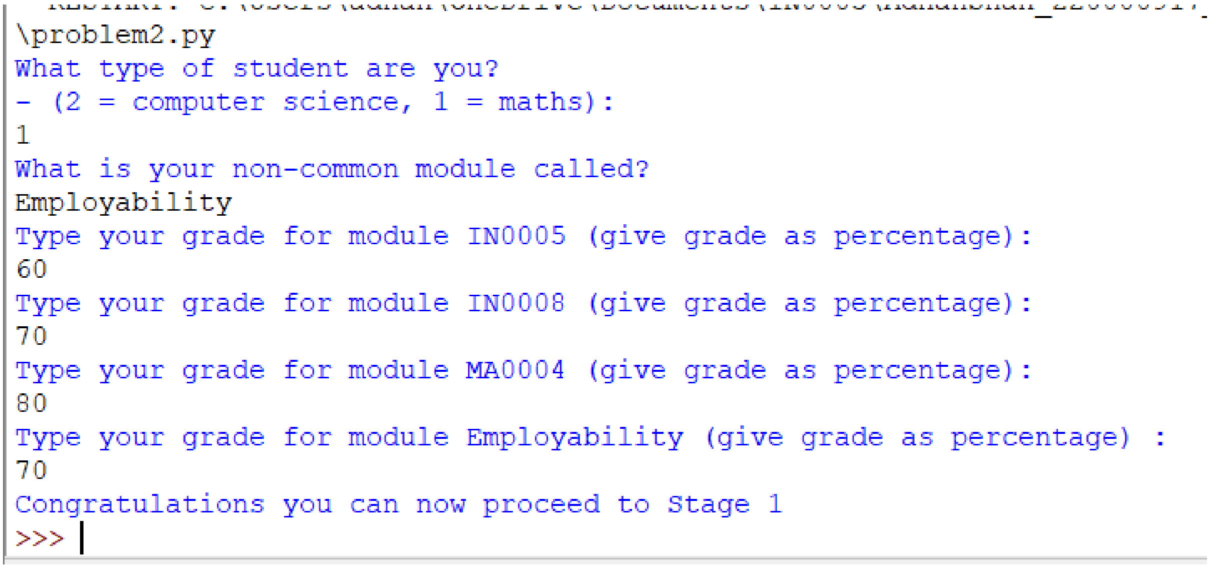


```
Let's talk about Adnan Shah (That is me, hi).
He's 18 and a half.
He's 5.9 feet tall.
He's 84 kg... do some cardio.
He's got Brown, or Black, Blackish Brown (I dunno) eyes with Black and White hair.
His teeth are White, I think.
>>>
```

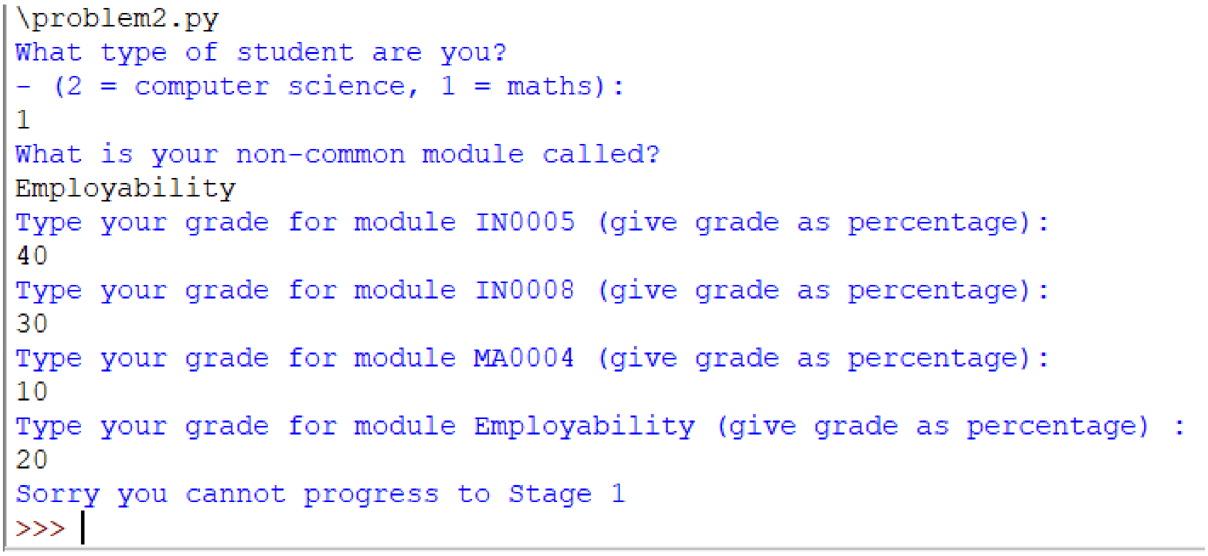
2) Problem 2:



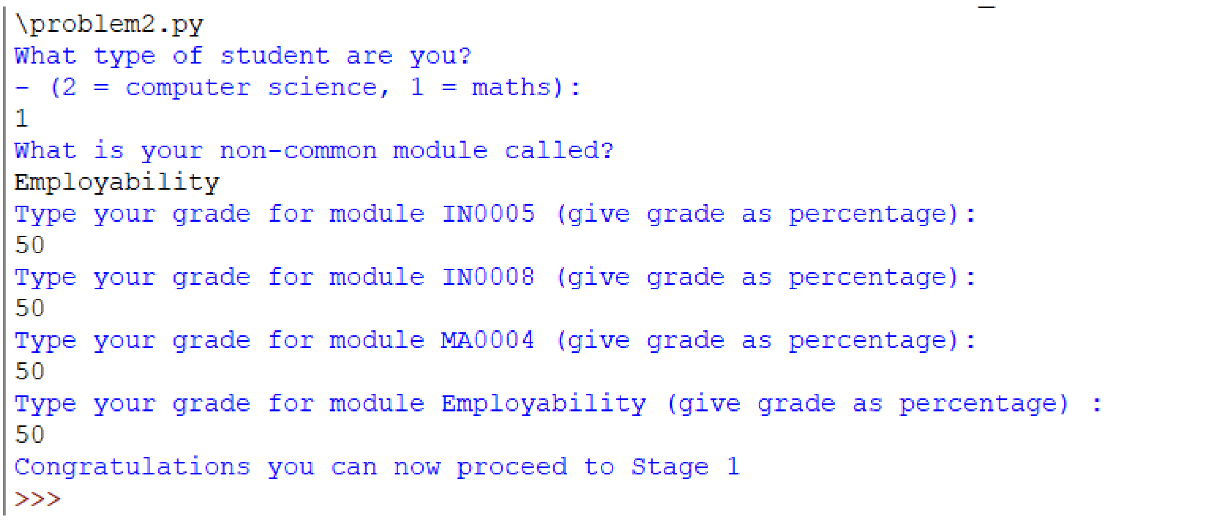
```
def grade(module_IN0005, module_IN0008, module_MA0004, module_b):
    average_grade = (module_IN0005 + module_IN0008 + module_MA0004 + module_b) / 4
    if student_type == 1:
        if average_grade >= 50:
            return ("Congratulations you can now proceed to Stage 1")
        elif average_grade < 50:
            return ("Sorry you cannot progress to Stage 1")
    elif student_type == 2:
        if average_grade >= 60:
            return ("Congratulations you can now proceed to Stage 1")
        elif average_grade < 60:
            return ("Sorry you cannot progress to Stage 1")

student_type = int(input("""What type of student are you?
- (2 = computer science, 1 = maths):\n"""))

b = input("What is your non-common module called?\n")

module_IN0005 = float(input("Type your grade for module IN0005 (give grade as percentage):\n'
module_IN0008 = float(input("Type your grade for module IN0008 (give grade as percentage):\n'
module_MA0004 = float(input("Type your grade for module MA0004 (give grade as percentage):\n'
module_b = float(input("Type your grade for module {} (give grade as percentage) :\n".format

print(grade(module_IN0005, module_IN0008, module_MA0004, module_b))
```



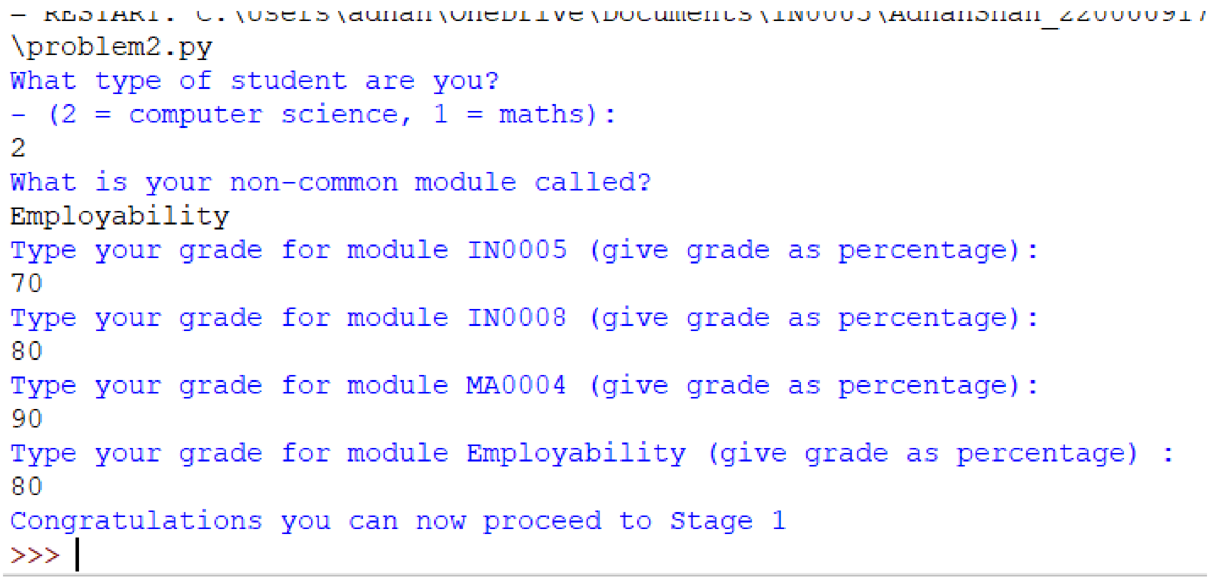
```
\problem2.py
What type of student are you?
- (2 = computer science, 1 = maths):
1
What is your non-common module called?
Employability
Type your grade for module IN0005 (give grade as percentage):
60
Type your grade for module IN0008 (give grade as percentage):
70
Type your grade for module MA0004 (give grade as percentage):
80
Type your grade for module Employability (give grade as percentage) :
70
Congratulations you can now proceed to Stage 1
>>> 
```



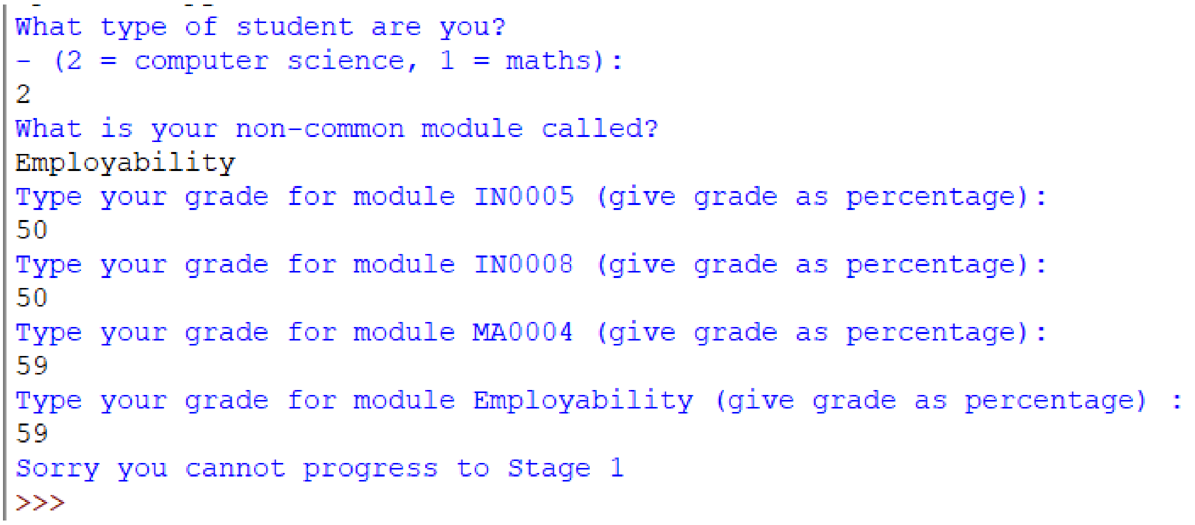
```
\problem2.py
What type of student are you?
- (2 = computer science, 1 = maths):
1
What is your non-common module called?
Employability
Type your grade for module IN0005 (give grade as percentage):
40
Type your grade for module IN0008 (give grade as percentage):
30
Type your grade for module MA0004 (give grade as percentage):
10
Type your grade for module Employability (give grade as percentage) :
20
Sorry you cannot progress to Stage 1
>>> 
```



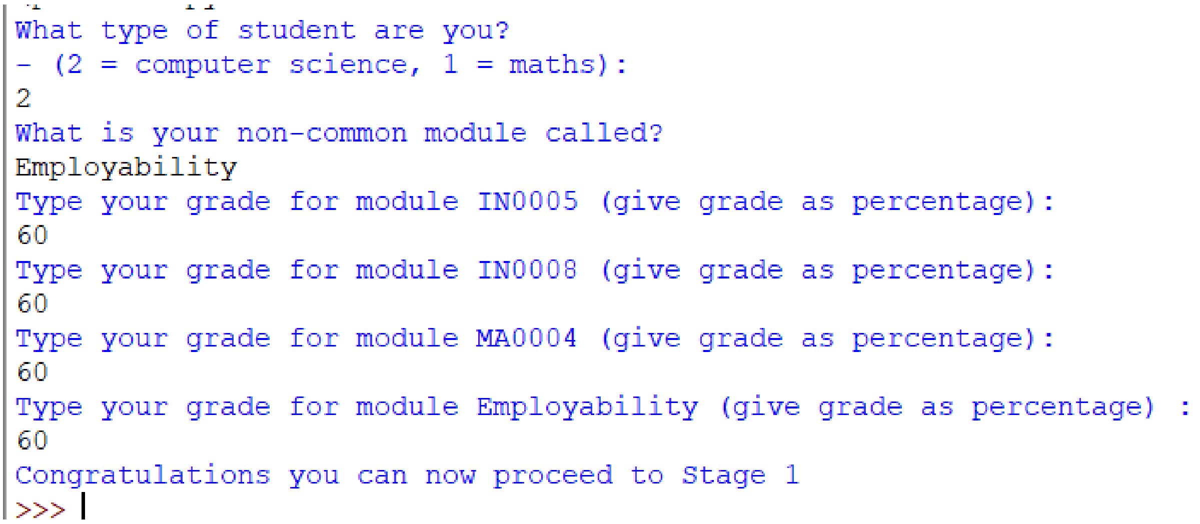
```
\problem2.py
What type of student are you?
- (2 = computer science, 1 = maths):
1
What is your non-common module called?
Employability
Type your grade for module IN0005 (give grade as percentage):
50
Type your grade for module IN0008 (give grade as percentage):
50
Type your grade for module MA0004 (give grade as percentage):
50
Type your grade for module Employability (give grade as percentage) :
50
Congratulations you can now proceed to Stage 1
>>>
```



```
= RESTART: C:\Users\adnan\OneDrive\Documents\IN0005\AdnanShah_220000917
\problem2.py
What type of student are you?
- (2 = computer science, 1 = maths):
2
What is your non-common module called?
Employability
Type your grade for module IN0005 (give grade as percentage):
70
Type your grade for module IN0008 (give grade as percentage):
80
Type your grade for module MA0004 (give grade as percentage):
90
Type your grade for module Employability (give grade as percentage) :
80
Congratulations you can now proceed to Stage 1
>>> 
```

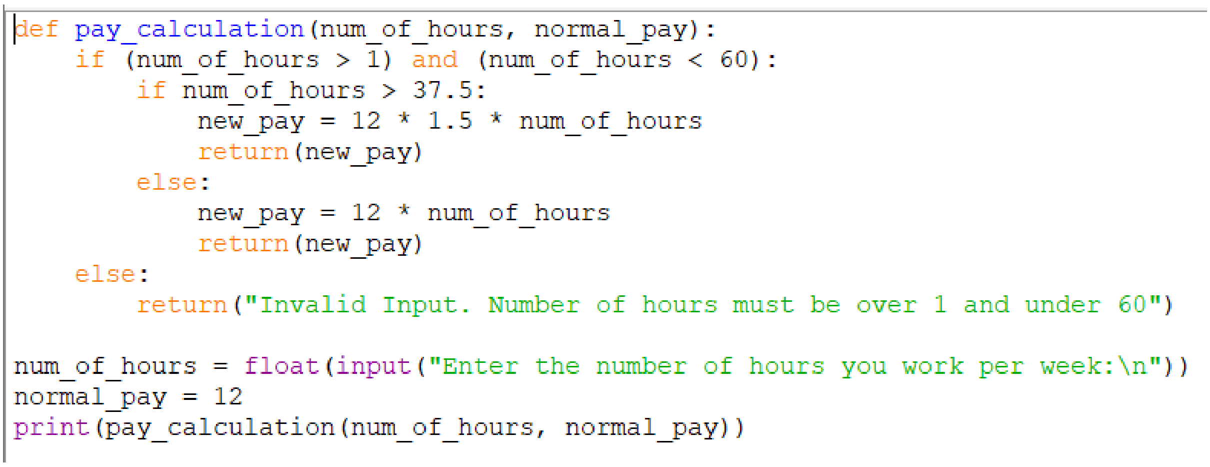


```
What type of student are you?
- (2 = computer science, 1 = maths):
2
What is your non-common module called?
Employability
Type your grade for module IN0005 (give grade as percentage):
50
Type your grade for module IN0008 (give grade as percentage):
50
Type your grade for module MA0004 (give grade as percentage):
59
Type your grade for module Employability (give grade as percentage) :
59
Sorry you cannot progress to Stage 1
>>>
```

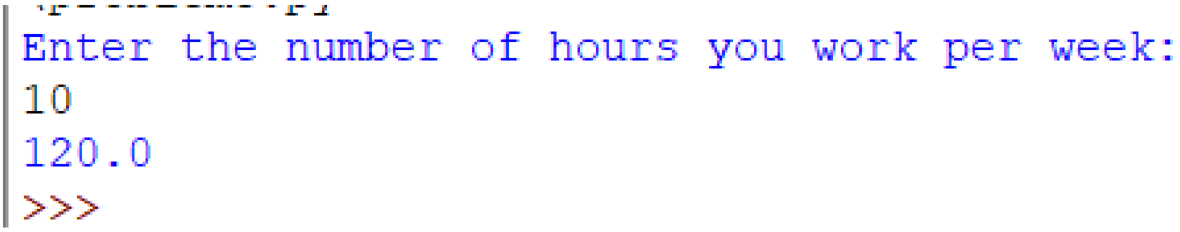


```
What type of student are you?
- (2 = computer science, 1 = maths):
2
What is your non-common module called?
Employability
Type your grade for module IN0005 (give grade as percentage):
60
Type your grade for module IN0008 (give grade as percentage):
60
Type your grade for module MA0004 (give grade as percentage):
60
Type your grade for module Employability (give grade as percentage) :
60
Congratulations you can now proceed to Stage 1
>>> 
```

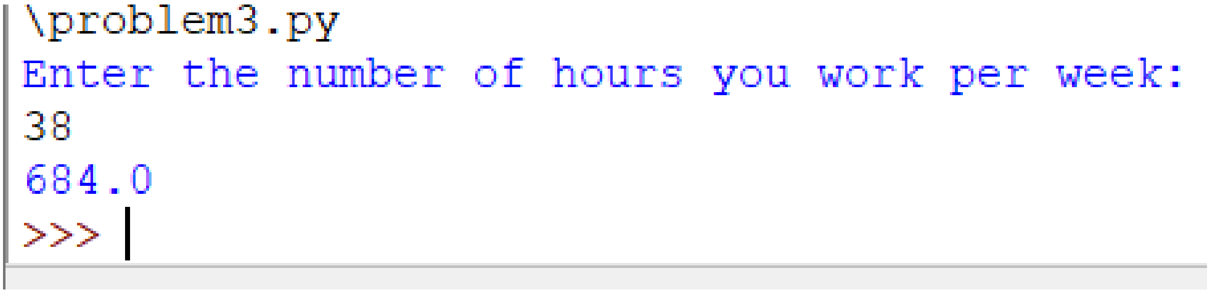
3) Problem 3:



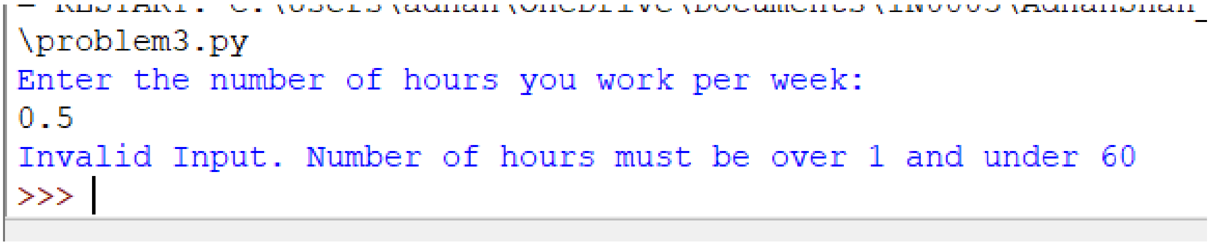
```
def pay_calculation(num_of_hours, normal_pay):
    if (num_of_hours > 1) and (num_of_hours < 60):
        if num_of_hours > 37.5:
            new_pay = 12 * 1.5 * num_of_hours
            return(new_pay)
        else:
            new_pay = 12 * num_of_hours
            return(new_pay)
    else:
        return("Invalid Input. Number of hours must be over 1 and under 60")

num_of_hours = float(input("Enter the number of hours you work per week:\n"))
normal_pay = 12
print(pay_calculation(num_of_hours, normal_pay))
```



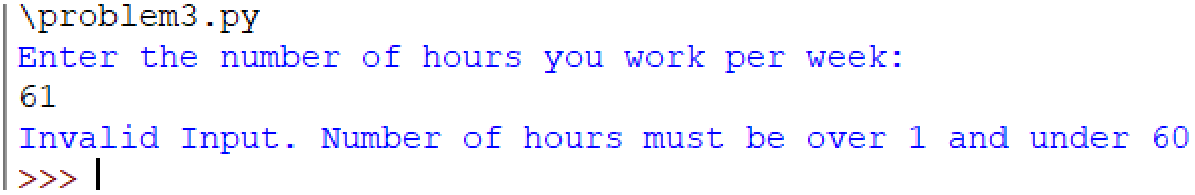
```
Enter the number of hours you work per week:
10
120.0
>>>
```



```
\problem3.py
Enter the number of hours you work per week:
38
684.0
>>> 
```

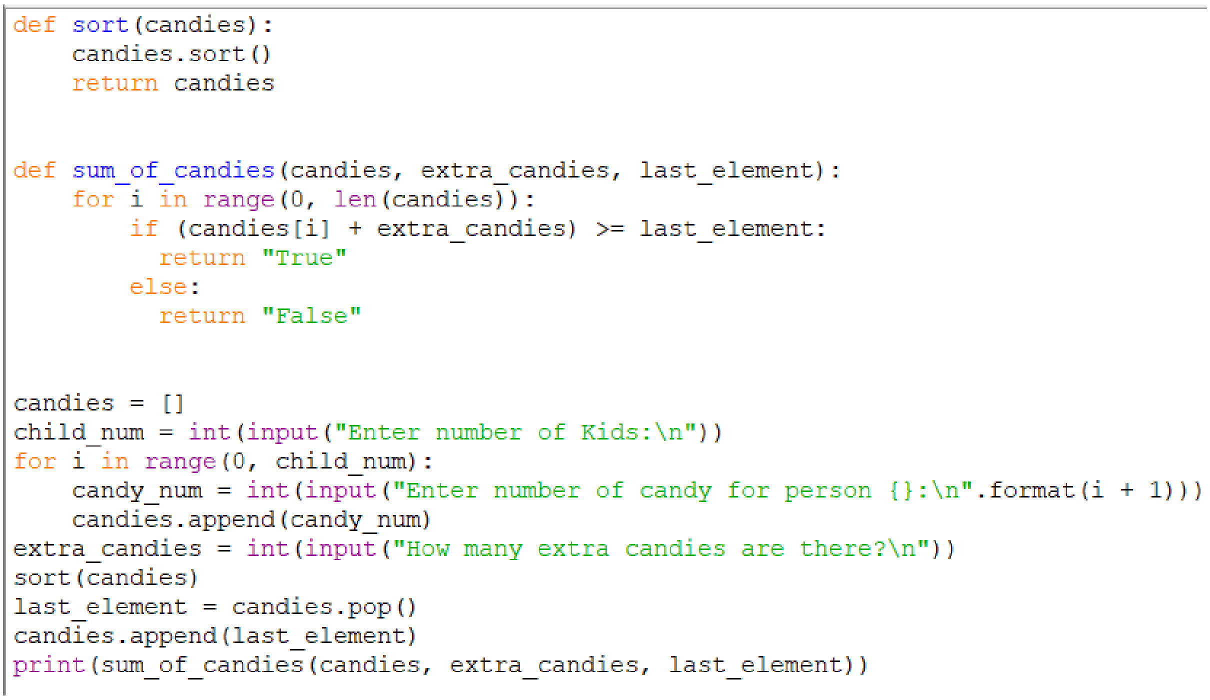


```
\problem3.py
Enter the number of hours you work per week:
0.5
Invalid Input. Number of hours must be over 1 and under 60
>>> 
```

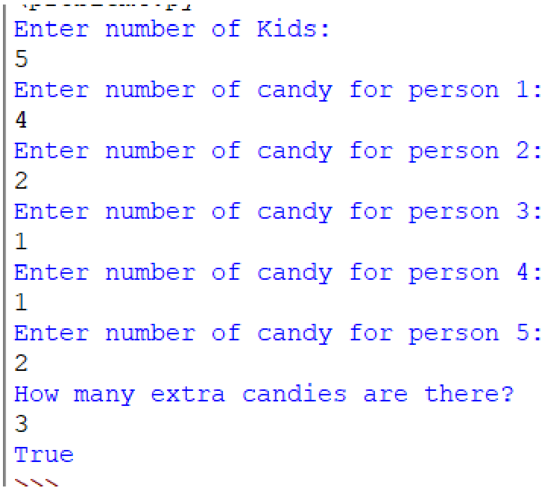


```
\problem3.py
Enter the number of hours you work per week:
61
Invalid Input. Number of hours must be over 1 and under 60
>>> 
```

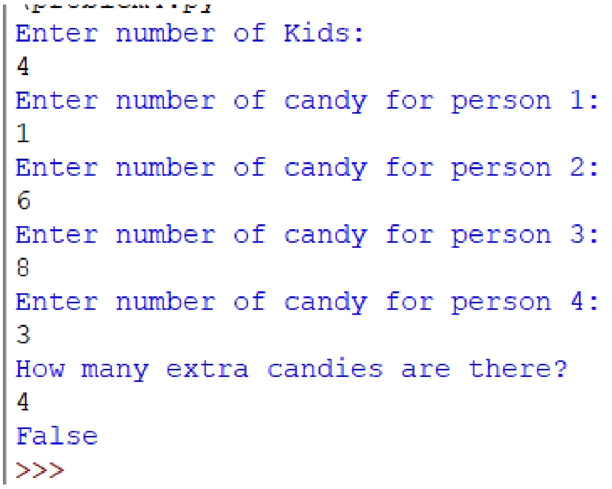
4) Problem 4:



```
def sort(candies):
    candies.sort()
    return candies


def sum_of_candies(candies, extra_candies, last_element):
    for i in range(0, len(candies)):
        if (candies[i] + extra_candies) >= last_element:
          return "True"
        else:
          return "False"


candies = []
child_num = int(input("Enter number of Kids:\n"))
for i in range(0, child_num):
    candy_num = int(input("Enter number of candy for person {}:\n".format(i + 1)))
    candies.append(candy_num)
extra_candies = int(input("How many extra candies are there?\n"))
sort(candies)
last_element = candies.pop()
candies.append(last_element)
print(sum_of_candies(candies, extra_candies, last_element))
```



```
Enter number of Kids:
5
Enter number of candy for person 1:
4
Enter number of candy for person 2:
2
Enter number of candy for person 3:
1
Enter number of candy for person 4:
1
Enter number of candy for person 5:
2
How many extra candies are there?
3
True
```



```
Enter number of Kids:
4
Enter number of candy for person 1:
1
Enter number of candy for person 2:
6
Enter number of candy for person 3:
8
Enter number of candy for person 4:
3
How many extra candies are there?
4
False
>>>
```