Question 1)

There is a group of tourists who would like to visit a city where the official currency is `CengCoin`. The group would like to visit the museums in the city. However, they have a limited amount of time and money, and they would like to make good use of their resources. For this purpose, they will only visit the museums with entry prices less than or equal to 25 `CengCoin` and greater than or equal to 10 `CengCoin`. You will implement a program that will calculate the total amount of money the group will pay during their visit. Note that, if the entry price of a museum is within the given limit, the entire group will visit that museum.

First of all, you will read an input as a list with `n` elements where the first `n-1` elements are integers that define the per person entry prices for each museum. This means that there are in total `n-1` museums in the city. The last element of the list is again an integer indicating the total number of tourists in the group. Finally, you will print the total amount of money the group will pay. Notice that, the list size `n` is not fixed.

Input Format:

$[EntryPrice_1, EntryPrice_2, ..., EntryPrice_{n-1}, GroupSize]$

Regulations & Hints

- You can read a list from input directly with `eval(input())`.
- You should **NOT** print anything other than the result. Otherwise, your code will be graded as **ZERO**. Please verify that your outputs confirm with the samples given below: (Neither the string "Input:" nor "Output:" will be printed. You will print only a single integer which is the total amount of money.)

SAMPLE I/O:

```
Input:
[5, 3, 11, 9, 29, 3]
Output:
33

Input:
[7, 20, 25, 2]
Output:
90
```

To help you understand the question better, you can follow the execution steps of the second example input/output combination which you can find below:

- Read the input list. Notice that for this case `n` is 4.
- First 3 integers (i.e., `n-1` integers) are museum ticket prices and the last element is the number of people in the group.
- Find the total ticket price to be paid per person where each ticket has to be between 10 and 25 `CengCoin`. (In this case, 20+25=45)
- Multiply it with the number of people in the group and print the final result. (45×2=90)

Solution

```
entry_prices = eval(input())
total_ticket_prices = 0
total_money = 0

for i in range(len(entry_prices)-1):
      if entry_prices[i] <= 25 and entry_prices[i] >= 10:
            total_ticket_prices += entry_prices[i]

total_money = total_ticket_prices * entry_prices[-1]
print(total_money)
```

Question 2) (Already given to students, no need to solve in the lab)

VENDING MACHINE (VM)

You are to implement a program that simulates a vending machine (VM). Your program will read three pieces of data in the following order:

1. A dictionary, representing the current stock of the VM. The keys of the dictionary are item names (as strings) and the prices as integers. You can assume the stock is limitless for the items in the VM. E.g.:
   {"chocolate": 2, "spring_water": 1, "energy_drink": 7}
2. A list of strings, representing the items requested by the customer. Repeating an item's name means requesting a second from that item, e.g.:
   ["energy_drink", "chips", "chocolate", "chocolate"]
   If a requested item is not in the VM, your VM should ignore it.
3. An integer, representing the amount of money inserted by the customer.

After these three inputs, your VM should calculate the total price of the items requested and should print one of the following:

- "Change:X" if the money inserted is *higher* than the total price of the requested items. The integer X is the amount of money that the customer needs to get back.
- "Insert:X" if the money inserted is *less* than the total price of the requested items. The integer X is the amount of extra money that the customer needs to insert.
- "Done", if the inserted money and the total price of the requested items are equal to each other.

Hints:

Hint 1: You can use *eval(input())* to read the provided dictionary and list directly.

Hint 2: If needed, you can use *<dictionary>.keys()* to get the keys of the dictionary.

Example I/O:

==========

Input 1:

{"chocolate": 2, "spring_water": 1, "energy_drink": 7}

["energy_drink", "chocolate"]

20

Output 1:

Change:11

==========

Input 2:

{"chocolate": 2, "spring_water": 1, "energy_drink": 7, "coke":5}

["coke", "coke", "coke", "chips", "candy", "chocolate"]

15

Output 2:

Insert:2

==========

Input 3:

{"chocolate": 2, "spring_water": 1, "energy_drink": 7, "coke":5, "hand_sanitizer": 5}

["hand_sanitizer", "mask"]

5

Output 3:

Done

Solution

```
vending_machine = eval(input())
requests = eval(input())
inserted_money = int(input())

valid_requests = [req for req in requests if req in vending_machine.keys()]

total_price = 0
for item in valid_requests:
    price = vending_machine[item]
    total_price += price

balance = inserted_money - total_price

if balance < 0:
    print("Insert:" + str(-1 * balance))

elif balance > 0:
    print("Change:" + str(balance))

else:
    print("Done")
```