Problem - 01

- You have x no. of 5 rupee coins and y no. of 1 rupee coins. You want to purchase an item for amount z. The shopkeeper wants you to provide exact change. You want to pay using minimum number of coins. How many 5 rupee coins and 1 rupee coins will you use? If exact change is not possible then display -1.

Available Rs. 1 coins	Available Rs. 5 notes	Amount to be made	Rs. 1 coins needed	Rs. 5 notes needed
2	4	21	1	4
11	2	11	1	2
3	3	19	-1	

In [24]:

```
def coin_change_problem(total_amount, coins_of_five, coins_of_one):
    x_five = 0 # initialize the no of five needed for the total amount
    x_one = 0  # Same with no of one needed

    if( total_amount > (5 * coins_of_five + 1 * coins_of_one)):  # check what is the maximu
        print(-1)
    else:
        x_five = int(total_amount / 5)
        x_one = total_amount - (5 * x_five)
        print("No. of Five needed :", x_five)
        print("No. of One needed  :", x_one)

coin_change_problem(41, 10, 6)
```

```
No. of Five needed : 8
No. of One needed  : 1
```

In [ ]:

```

```

Problem - 02

- FoodCorner home delivers vegetarian and non-vegetarian combos to its customer based on order.
- A vegetarian combo costs Rs.120 per plate and a non-vegetarian combo costs Rs.150 per plate. Their non-veg combo is really famous that they get more orders for their non-vegetarian combo than the vegetarian combo.
- Apart from the cost per plate of food, customers are also charged for home delivery based on the distance in kms from the restaurant to the delivery point. The delivery charges are as mentioned below:

Distance in kms	Delivery charge in Rs. per km
For first 3kms	0
For next 3kms	3
For the remaining	6

- Given the type of food, quantity (no. of plates) and the distance in kms from the restaurant to the delivery point, write a python program to calculate the final bill amount to be paid by a customer.
- The below information must be used to check the validity of the data provided by the customer:
- Type of food must be ‘V’ for vegetarian and ‘N’ for non-vegetarian.
- Distance in kms must be greater than 0.
- Quantity ordered should be minimum 1.
- If any of the input is invalid, the bill amount should be considered as -1.

In [36]:

```
def calculate_bill_amount(food_type, quantity_ordered, distance_in_kms):
    bill_amount = 0

    if((food_type == "V" or  food_type == "N") and (quantity_ordered > 0 and distance_in_km
        if(food_type == "V"):
            bill_amount = 120 * quantity_ordered
        elif(food_type == "N"):
            bill_amount = 150 * quantity_ordered
        else:
            bill_amount = -1

        if(0 < distance_in_kms <= 3):
            bill_amount += 0
        elif(3 < distance_in_kms <= 6):
            bill_amount += 3 * (distance_in_kms - 3)
        else:
            bill_amount += (9 + 6 * (distance_in_kms - 6))
    else:

        bill_amount = -1

    return bill_amount

bill_amount = calculate_bill_amount("N", 3, 6)
print(bill_amount)
```

```
459
```

Problem - 03

- Write a python program which finds the maximum number from num1 to num2 (num2 inclusive) based on the following rules.
  - Always num1 should be less than num2
  - Consider each number from num1 to num2 (num2 inclusive). Populate the number into a list, if the below conditions are satisfied
    - Sum of the digits of the number is a multiple of 3
    - Number has only two digits
    - Number is a multiple of 5

In [51]:

```
def find_max(num1, num2):

    list_x1 = []
    max_num = -1
    list_x = list(range(num1, num2 + 1))

    if num1 >= num2:
        max_num = -1
    else:
        for i in list_x:
            temp_1 = i
            temp_2 = 0
            while temp_1 != 0:
                temp_2 += temp_1 % 10
                temp_1 = temp_1 // 10

            if temp_2 % 3 == 0 and i % 5 == 0 and len(str(i)) == 2:   # checking if the num
                    list_x1.append(i)

        if(len(list_x1) != 0):   # checking if list is empty or not
            list_x1.sort()     # to find the largest element in the list
            max_num = list_x1[-1]   # printing the last element
        else:
            max_num = -1
    return max_num
max_num = find_max(20, 45)
print(max_num)
```

```
45
```

Problem - 04

- A teacher is conducting a camp for a group of five children. Based on their performance and behavior during the camp, the teacher rewards them with chocolates.
- Write a Python function to

    1. Find the total number of chocolates received by all the children put together. Assume that each child is identified by an id and it is stored in a tuple and the number of chocolates given to each child is stored in a list.
    2. The teacher also rewards a child with few extra chocolates for his/her best conduct during the camp.
        - If the number of extra chocolates is less than 1, an error message "Extra chocolates is less than 1", should be displayed.
        - If the given child Id is invalid, an error message "Child id is invalid" should be displayed. Otherwise, the extra chocolates provided for the child must be added to his/her existing number of chocolates and display the list containing the total number of chocolates received by each child.

In [67]:

```
child_id = (1, 2, 3, 4, 5)
chocolates_received = [10, 5, 7, 4, 6]

def calculate_total_chocolates():
    sum = 0
    for choco in chocolates_received:
        sum = sum + choco

    return sum

def reward_child(child_id_rewarded, extra_chocolates):

    if extra_chocolates < 1:
        print("Extra chocolates is less than 1")
    elif child_id_rewarded not in child_id:
        print("Child id is invalid")
    else:
        length = len(child_id)

        for num in range(length - 1):
            if child_id_rewarded == child_id[num]:
                chocolates_received[num] = chocolates_received[num] + extra_chocolates
        print(chocolates_received)


print(calculate_total_chocolates())

reward_child(1, 4)
```

```
32
[14, 5, 7, 4, 6]
```

problem - 05

- Write python function, **sms_encoding()** which accepts a sentence and converts it into an abbreviated sentence to be sent as SMS and returns the abbreviated sentence.
- Rules are as follows:
  - Spaces are to be retained as is
  - Each word should be encoded separately
    - If a word has only vowels then retain the word as is
    - If a word has a consonant (at least 1) then retain only those consonants
- Note: Assume that the sentence would begin with a word and there will be only a single space between the words.

Sample Input	ExpectedOutput
I love Python	I lv Pythn
MSD says I love cricket and tennis too	MSD sys I lv crckt nd tnns t
I will not repeat mistakes	I wll nt rpt mstks

In [63]:

```
def encrypt_sentence(sentence):
    word = sentence.split()      # splitting the word by word
    vowels = "aeiouAEIOU"
    st = ""
    for i in word:
        if(len(i) == 1):
            st = st + i
        else:
            for j in i:
                if j not in set(vowels):
                    st = st + j
        st = st + " "
    return st[0: -1]
sentence = "To the Love of my life"
encrypt_sentence(sentence)
```

Out[63]:

```
'T th Lv f my lf'
```

In [ ]: