```
# Assignmenr 4

import re
pattern = '[A-Za-z:*$,-]+'
string = 'Fare:$18,369 - $120,379*'
result = re.sub(pattern,'',string) # replacing pattern in place of
nothing('') in string
#storing fare in var
var1=""
var2=""
start=0
while result[start]!=" ":
    var1+=result[start];
    start+=1
while start<=len(result)-1:
    var2+=result[start];
    start+=1
# string to int conversion
fare1=int(var1)
fare2=int(var2)
# compare
if (fare1>0 and fare2>0):
    print("fare1 and fare2 is greater than zero")
```

```
18369   120379
fare1 and fare2 is greater than zero
```

```
import re

def function(string):
    pattern = '[A-Za-z:*$,-]+'
    result = re.sub(pattern,'',string)  # replacing pattern with
nothing('') in the string
    var1=""
    var2=""
    start=0
    while result[start]!=" ":
        var1+=result[start];
        start+=1
    while start<=len(result)-1:
        var2+=result[start];
        start+=1
    # string to int conversion
    fare1=int(var1)
    fare2=int(var2)
    # compare
    if (fare1>0 and fare2>0):
        return "fare1 and fare2 is greater than zero"

    else:
        return "Both are smaller than zero"
```

```
string=input("Enter the String :")
function(string)
```

```
Enter the String :Fare:$18,369 - $120,379*
```

```
'fare1 and fare2 is greater than zero'
```

2nd approach of assignment

```
text = 'Fare: $18,369 - $120,379*'
text_range=text[6:]
print(text_range)
```

```
$18,369 - $120,379*
```

```
special_chars='$,*'
new_text_range=''
for value in text_range:
    if value in special_chars:
        continue
    else:
        new_text_range +=value
print(new_text_range)
```

```
18369 - 120379
```

```
value1=int(new_text_range.split('-')[0])  # splitting the i/p on the
basis of '-'
```

```
value2=int(new_text_range.split('-')[1])
```

```
if value1>0 and value2>0:
    print("both the value are greater than 0")
else:
    print("not greater")
```

```
both the value are greater than 0
```

```
# user input
#input() function to take i/p from user
#int() function to convert string to int
```

```
name=input('what is your name')
print(f"you have entered your name as {name}")
```

```
what is your nameDeepinder
you have entered your name as Deepinder
```

```
name=input('what is your name')
year=input('which year is it ')
print(type(year))
```

```
print(f"you have entered your name as {name}")
print(f"2 years down the line it will be {int(year)+2}")
```

```
what is your nameAnkit Nayan
which year is it 3
<class 'str'>
you have entered your name as Ankit Nayan
2 years down the line it will be 5
```

```
# task
no_of_people=input('How many people are in their dinner group: ')
if int(no_of_people)>8:
    print("They will have to wait for table")
else:
    print("Table is ready")
```

```
How many people are in their dinner group: 11
They will have to wait for table
```

```
number=input("Enter the number: ")
if int(number)%10==0:
    print(f"{number} is multiple of 10")
else:
    print("number is not multiple of 10")
```

```
enter the number: 100
100 is multiple of 10
```

```
name=input("enter your name: ")
current_year=input("enter the current year: ")
if int(current_year)%2==0:
    print(f"you entered your name as: {name} year {current_year} is an
even year")
else:
    print(f"{current_year} is not an even year")
```

```
enter your name: Ankit Nayan
enter the current year: 2020
you entered your name as: Ankit Nayan year 2020 is an even year
```

while loop

#loops keep working untill certain condition is not satisfied

```
# virtual parrot
prompt= "Please enter something"
active=True
while active:
    message=input(prompt)
```

```
    if message=='quit':
        active=False
    else:
        print(message)
```

```
Please enter somethinggood morning
good morning
Please enter somethinggood morning
good morning
Please enter somethinggood afternoon
good afternoon
Please enter somethingquit
```

```
# wrong uses of while loop
i=1
while i<5:
    print("good morning")
    i+=1
```

```
good morning
good morning
good morning
good morning
```

```
# break and continue
prompt= "Please enter something"
active=True
while active:
    message=input(prompt)
    if message=='quit':
        # active=False
        break
    else:
        print(message)
```

```
Please enter somethinghi
hi
Please enter somethinghi
hi
Please enter somethingquit
```

```
#Task 2
toppings="enter the series of toppings: "
active=True
while active:
    message=input(toppings)
    if message =='quit':   # or 'Quit' or 'QUIT'
        break
        print(f"{message} toppings will be added on pizza")
    else message == 'cherry':  #or 'basil'
        print("i don't have these toppings: ")
```

```
  File "<ipython-input-53-8bcc7945bbc8>", line 9
    else message == 'cherry':  #or 'basil'
         ^
SyntaxError: invalid syntax
```

```
prompt="please enter your toppings. Type 'quit' when done"
topping_not_availaible_in_store=['basil leaves','cherry']
toppings=[]
active =True

while active:
    message=input(prompt)
    if message.lower()=='quit':
        break
    else:
        toppings.append(message)
for topping in toppings:
    if topping in topping_not_availaible_in_store:
        print(f"We are sorry! {topping.title()} is not available
today")
    else:
        print(f"Topping {topping} will be added on your pizza")
```

```
please enter your toppings. Type 'quit' when donejalapeno
please enter your toppings. Type 'quit' when donecheese
please enter your toppings. Type 'quit' when donechicken
please enter your toppings. Type 'quit' when donequit
Topping jalapeno will be added on your pizza
Topping cheese will be added on your pizza
Topping chicken will be added on your pizza
```

Functions

```
def function_name():
function_name()
```

```
def greet_user():   #defining the function
    print("Hello How are you!")

greet_user()  #calling the function
```

```
Hello How are you!
```

```
# arguments and parameters of a function
def function_name(parameter):
function_name(argument)
```

```
def greet_user(name):   #name is parameter
    print(f"Hello How are you! {name}")
```

```
greet_user("ankit") #ankit is argument
```

```
Hello How are you! ankit
```

```
def greet_user(name,tempreature):   #name is parameter
    print(f"Hello How are you! {name}")
    print(f"it is {tempreature} degree today")
greet_user("ankit",36)
```

```
Hello How are you! ankit
it is 36 degree today
```

```
# Task 1
def favorite_book(title):   #title is parameter
    print(f"one of my favorite book is {title}")
favorite_book("alice in the wonderland")  # alice in the wonderland is
argument
```

```
one of my favorite book is alice in the wonderland
```

```
# functions with deafult parameter
def describe_pet(pet_name,color,animal_type='dog'):
    print(f"I have a {animal_type}")
    print(f"It's name is {pet_name}")
    print(f"It is {color} in color")
    print()
describe_pet("buzz","white")
describe_pet("kittie","black","cat")
```

```
I have a dog
It's name is buzz
It is white in color

I have a cat
It's name is kittie
It is black in color
```

```
# keyword argument
def describe_pet(pet_name,color,animal_type='dog'):
    print(f"I have a {animal_type}")
    print(f"It's name is {pet_name}")
    print(f"It is {color} in color")
    print()
describe_pet(color="white",pet_name="buzz")  # key word argument #name
value pair which you pass with the function
describe_pet("kittie","black","cat")  # positional argument

# using key word argument
def make_shirt(size,text):
    print(f"the size of shirt {size} and {text} should be printed on
```

```
the shirt as message")
make_shirt("6","jai shree ram")
```

```
the size of shirt 6 and jai shree ram should be printed on the shirt
as message
```

```
# using positional argument
def make_shirt(size,text):
    print(f"the size of shirt {size} and {text} should be printed on
the shirt as message")
make_shirt(text="jai shree ram",size="6")
```

```
the size of shirt 6 and jai shree ram should be printed on the shirt
as message
```

```
#doubt
```

```
# Return value
def get_full_name(first_name,last_name):
    full_name=f"{first_name} {last_name}"
    return full_name
get_full_name('James' ,'Bond')
```

```
'James Bond'
```

```
# Return value
def get_full_name(first_name,last_name):
    full_name=f"{first_name} {last_name}"
    return full_name
full_name=get_full_name('James' ,'Bond')
print(full_name)
```

```
James Bond
```

```
# making an argument optional
def get_full_name(first_name,last_name,middle_name=''):
    if middle_name:
        full_name=f"{first_name} {last_name} {last_name}"
    else:
        full_name=f"{first_name} {last_name}"
    return full_name

full_name=get_full_name('James' ,'Bond')
print(full_name)
```

```
James Bond
```

```
#Return dictionary
def create_person(first_name,last_name):
    person={
        'first':first_name,
        'last':last_name,
```

```
    }
    return person
actor=create_person('james','bond')
print(f"first name={actor['first']}")
print(f"last name={actor['last']}")

first name=james
last name=bond

def create_person(first_name,last_name):
    person={
        'first':first_name,
        'last':last_name,
    }
    return person

first_name="ask the user first name "
last_name="ask the user last name"
active=True
while active:
    first=input(first_name)
    last=input(last_name)
    if first == 'quit':
        break
    else:
        print(first,last)
        x=create_person(first,last)
        print(x)


ask the user first name ankit
ask the user last namenayan
ankit nayan
{'first': 'ankit', 'last': 'nayan'}
ask the user first name quit
ask the user last namequit

def create_person(first_name,last_name):
    person={
        'first':first_name,
        'last':last_name,
    }
    return person
active =True
while active:
    print("Please enter first name and last name")
    print("type 'quit' to exit")
    first_name=input("firstname=")
    if first_name.lower()=='q':
        break
    if last_name.lower()=='q':
```

```
        break
actor=create_person(first_name,last_name)

# passing a list as parameter in function
def greet_people(names):
    for name in names:
        print(f"hello {name},welcome to the party")
people=['tim','tom','john','brad']
greet_people(people)
```

```
hello tim,welcome to the party
hello tom,welcome to the party
hello john,welcome to the party
hello brad,welcome to the party
```

```
# passing arbitrary number of argument
def function_name(*parameters)

def make_pizza(size,*toppings):  # you can take as many no of argument
    print(f"Making {size} = inches pizza with following toppings")
    for topping in toppings:
        print(f"{topping.title()}")
make_pizza(10, 'extra cheese')
make_pizza(12, 'extra cheese', 'capsicum', 'jalapeno')
```

```
Making 10 = inches pizza with following toppings
Extra Cheese
Making 12 = inches pizza with following toppings
Extra Cheese
Capsicum
Jalapeno
```

```
# positiona and arbitrary argument
# modules
```