1. List of vowel and consonants

*def* vowel\_counter(*str*):

vowel\_count=0

consonant\_count=0

*str*=*str*.lower()

for i in range(len(*str*)):

if *str*[i]=='a' or *str*[i]=='e' or *str*[i]=='i' or *str*[i]=='o' or *str*[i]=='u':

vowel\_count+=1

else:

consonant\_count+=1

return vowel\_count,consonant\_count

str=input().split()

list\_of\_not\_allowed=[]

list\_of\_allowed=[]

print(str)

for i in range(len(str)):

if len(str[i]) < 5:

list\_of\_not\_allowed.append(str[i])

else:

list\_of\_allowed.append(str[i])

vowel=[]

non\_vowel=[]

for i in range(len(list\_of\_allowed)):

vow,con = vowel\_counter(list\_of\_allowed[i])

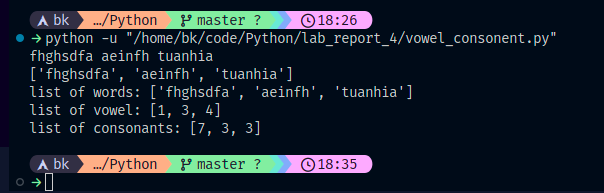
vowel.append(vow)

non\_vowel.append(con)

print(*f*"list of words: {list\_of\_allowed}")

print(*f*"list of vowel: {vowel}")

print(*f*"list of consonants: {non\_vowel}")



2. Verb identifier

verb\_list=["am","going",'enjoying']

reverse\_list=[]

str=input().split()

if len(str)<7:

print("condition doesn't match")

else:

for i in range(len(str)):

if str[i].lower() in verb\_list:

reverse\_list.append(str[i][::-1])

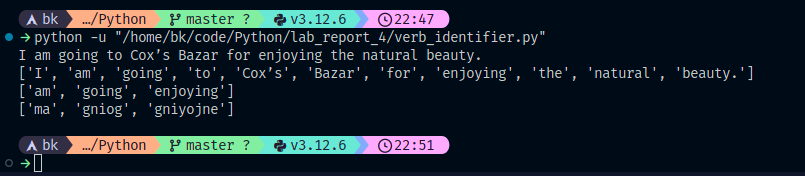
else:

continue

print(str)

print(verb\_list)

print(reverse\_list)



3. Positive Negative word

negative\_words=["bad","not","no","dangerous","terrible"]

positive\_words=["good","happy","amazing","congratulations","thanks"]

*def* sentiment\_analysis(*negative*,*positive*,*comment*):

for item in *negative*:

if item in *comment*:

return "negative"

for item in *positive*:

if item in *comment*:

return 'positive'

return "neutral"

comm=input("enter you comment: ").lower()

sent=sentiment\_analysis(negative\_words,positive\_words,comm)

print(comm)

if sent=='negative':

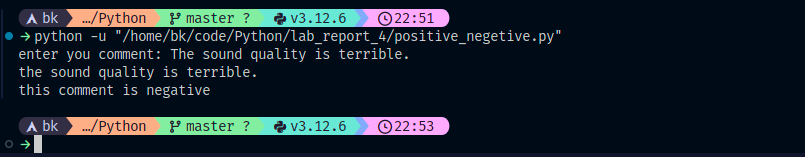
print("this comment is negative")

elif sent=='positive':

print("this comment is positive")

else:

print("this comment is neutral")



4. Number sum

input = list(map(int,input("enter numbers: ").split(',')))

list2=[]

for i in range(len(input)):

if (len(str(input[i])))<3:

print(*f*"{input[i]} not possible")

else:

sum=0

val=input[i]

while val>0:

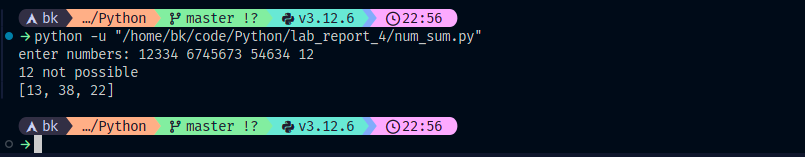
curr=val%10

sum=sum+curr

val=val//10

list2.append(sum)

print(list2)



5. Movie

*def* movie\_recommendation(*x*,*y*):

new\_list=[]

for i in *y*:

if i>6:

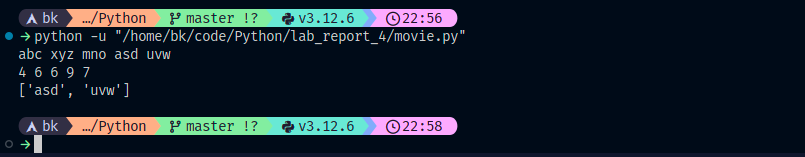
new\_list.append(*x*[*y*.index(i)])

print(new\_list)

movie\_list=list(map(str,input().strip().split()))

rating\_list=list(map(int,input().strip().split()))

movie\_recommendation(movie\_list,rating\_list)



6. Password

*def* pass\_maker(*string*):

if len(*string*) < 10:

print("expected password of 10 length")

return

ascii\_str=str(ord(*string*[0]))

upper\_str=*string*[-3:].upper()

lower\_str=*string*[:4].lower()

special\_str="@"

still\_now= ascii\_str+upper\_str+lower\_str

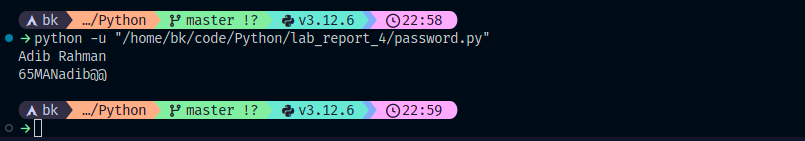
special\_str=special\_str\*((len(*string*))-len(still\_now))

return still\_now+special\_str

string=input()

new\_pass = pass\_maker(string)

print(new\_pass)



7. Book discount

book\_names = ['A', 'B', 'C', 'D']

book\_prices = [250, 150, 300, 450]

new\_dict={}

for i in range(len(book\_prices)):

if book\_prices[i]>=500:

discount\_price = book\_prices[i] - (book\_prices[i] \* 0.20)

elif 300 <= book\_prices[i] and book\_prices[i] < 500:

discount\_price = book\_prices[i] - (book\_prices[i] \* 0.15)

elif 100 <= book\_prices[i] and book\_prices[i] < 300:

discount\_price = book\_prices[i] - (book\_prices[i] \* 0.10)

else:

discount\_price = book\_prices[i] - (book\_prices[i] \* 0.05)

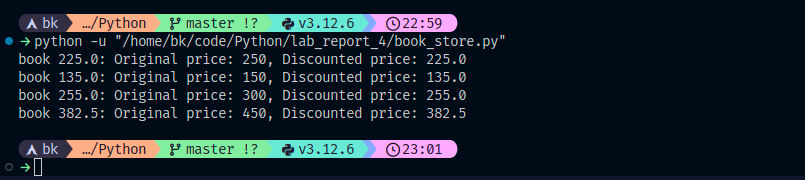
new\_dict[book\_names[i]]=discount\_price

i=0

for keys in new\_dict.keys():

print(*f*"book {new\_dict[keys]}: Original price: {book\_prices[i]}, Discounted price: {new\_dict[keys]}")

i+=1



8. Palindrome word

*def* palindrome(*str*):

for i in range(len(*str*)):

if *str*[i]!=*str*[len(*str*)-1-i]:

return False

return True

palindrome\_list=[]

list\_str=input().split()

for i in range(len(list\_str)):

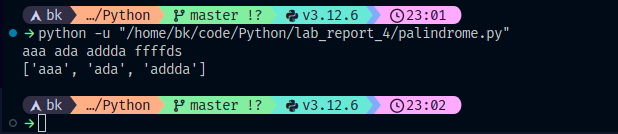
if palindrome(list\_str[i]):

palindrome\_list.append(list\_str[i])

else:

continue

print(palindrome\_list)



9. Purchase

book\_name = ['A', 'B', 'C', 'D', 'E']

book\_price = [200, 100, 300, 500, 250]

book\_list=input().split()

price=0

for i in range(len(book\_list)):

if book\_list[i] in book\_name:

index=book\_name.index(book\_list[i])

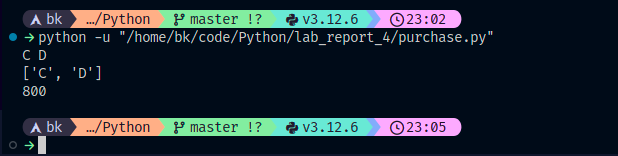
price=price+book\_price[index]

else:

print("book not found")

print(book\_list)

print(price)



10. Calculate balance

balance=int(input())

transactions=list(map(int,input().split()))

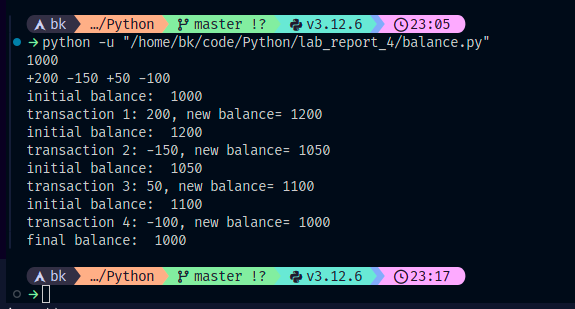
for i in range(len(transactions)):

print("initial balance: ",balance)

balance=balance+transactions[i]

print(*f*"transaction {i+1}: {str(transactions[i])}, new balance= {balance}")

print(*f*"final balance: ",balance)



11. Student registration

*def* student\_registration\_system(*student\_list*, *registration\_status*):

sum=0

for i in range(len(*registration\_status*)):

if *registration\_status*[i]=='Yes':

sum+=1

print(sum)

print(len(*student\_list*)-(sum))

updated\_list=[]

for i in range(len(*registration\_status*)):

if *registration\_status*[i]=="Yes":

updated\_list.append(*student\_list*[i])

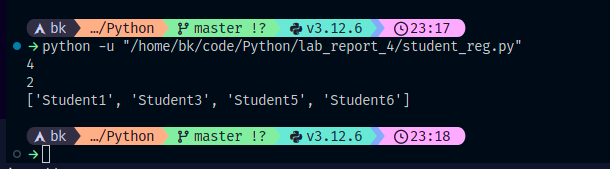
return updated\_list

student\_list = ['Student1', 'Student2', 'Student3', 'Student4', 'Student5', 'Student6']

registration\_status = ['Yes', 'No', 'Yes', 'No', 'Yes', 'Yes']

updated\_list = student\_registration\_system(student\_list, registration\_status)

print(updated\_list)



12. Generate Password

*def* pass\_maker(*string*):

if len(*string*) < 10:

print("expected password of 10 length")

return

ascii\_str=str(ord(*string*[0]))

upper\_str=*string*[-3:].upper()

lower\_str=*string*[:4].lower()

special\_str="@"

still\_now= ascii\_str+upper\_str+lower\_str

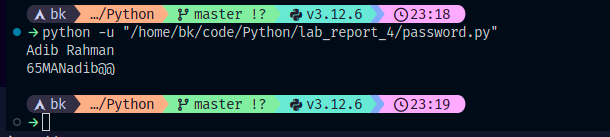
special\_str=special\_str\*((len(*string*))-len(still\_now))

return still\_now+special\_str

string=input()

new\_pass = pass\_maker(string)

print(new\_pass)



13. Temperature

celsius\_temps = [0, 10, 20, 30, 40, 50]

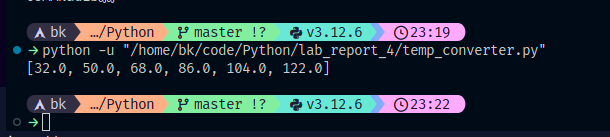
*def* temp\_convert(*celsius*):

fahrenheit = (*celsius* \* 9 / 5) + 32

return fahrenheit

new\_temps=list(map(temp\_convert,celsius\_temps))

print(new\_temps)



14. Subject score

subject\_scores = {

'Math' : [90, 85, 88, 92, 95] ,

'Physics' : [75, 80, 85, 90, 95] ,

'Chemistry': [85, 90, 92, 88, 82]

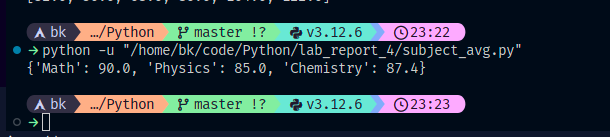
}

new\_dict={}

for key in subject\_scores.keys():

new\_dict[key] = sum(subject\_scores[key])/len(subject\_scores[key])

print(new\_dict)



15. Product average

product\_prices = {

'Apples': [1.5, 1.7, 1.6, 1.8],

'Bananas': [0.5, 0.6, 0.55, 0.65],

'Oranges': [2.0, 2.1, 2.05, 2.2]

}

new\_dict={}

for key in product\_prices.keys():

new\_dict[key] = sum(product\_prices[key])/len(product\_prices[key])

print(new\_dict)

