Exercise 1A: Create a string made of the first, middle and last character

Write a program to create a new string made of an input string's first, middle, and last character.

```
'''
Given:
str1 = "James"

Expected Output:
Jms
'''
str1=input("Enter string: ")
mid=len(str1)//2
print(str1[0]+str1[mid]+str1[-1])
```

```
Enter string: James
Jms
```

Exercise 1B: Create a string made of the middle three characters

Write a program to create a new string made of the middle three characters of an input string.

```
'''
Given:
Case 1
str1 = "JhonDipPeta"

Output
Dip

Case 2
str2 = "JaSonAy"

Output
Son
'''
str1=input("Enter string: ")
mid=len(str1)//2
print(str1[mid-1]+str1[mid]+str1[mid+1])
```

```
Enter string: JaSonAy
Son
```

Exercise 2: Append new string in the middle of a given string

Given two strings, s1 and s2. Write a program to create a new string s3 by appending s2 in the middle of s1.

```
'''
Given:
s1 = "Ault"
s2 = "Kelly"

```

```
Expected Output:
AuKellylt
'''
s1=input("Enter a string: ")
s2=input("Enter a new string: ")
mid=len(s1)//2
print(s1[:mid]+s2+s1[mid:])
```

```
Enter a string: Ault
Enter a new string: Kelly
AuKellylt
```

Exercise 3: Create a new string made of the first, middle, and last characters of each input string

Given two strings, s1 and s2, write a program to return a new string made of s1 and s2's first, middle, and last characters.

```
'''
Given:
s1 = "America"
s2 = "Japan"

Expected Output:
AJrpan
'''
s1=input("Enter a string: ")
s2=input("Enter a new string: ")
mid1=len(s1)//2
mid2=len(s2)//2
print(s1[0]+s2[0]+s1[mid+1]+s2[mid]+s1[-1]+s2[-1])
```

```
Enter a string: America
Enter a new string: Japan
AJrpan
```

Exercise 4: Arrange string characters such that lowercase letters should come first

Given string contains a combination of the lower and upper case letters. Write a program to arrange the characters of a string so that all lowercase letters should come first.

```
'''
Given:
str1 = PyNaTive

Expected Output:
yaivePNT
'''
s1=input("Enter a string: ")
s2=""
for char in s1:
  if char.islower():
    s2=s2+char
for char in s1:
  if char.isupper():
```

```
    s2=s2+char
print(s2)
```

```
Enter a string: PyNaTive
yaivePNT
```

Exercise 5: Count all letters, digits, and special symbols from a given string

```
'''
Given:
str1 = "P@#yn26at^&i5ve"

Expected Outcome:
Total counts of chars, digits, and symbols
Chars = 8
Digits = 3
Symbol = 4
'''
s1=input("Enter a string: ")
char, digi, symb=0, 0, 0
for ch in s1:
  if ch.isalpha():
    char+=1
  elif ch.isdigit():
    digi+=1
  else:
    symb+=1
print("Total counts of chars, digits, and symbols\nChars =", char, "\nDigits =", digi, "\nSymbols =", symb)
```

```
Enter a string: P@#yn26at^&i5ve
Total counts of chars, digits, and symbols
Chars = 8
Digits = 3
Symbols = 4
```

Exercise 6: Create a mixed String using the following rules

Given two strings, s1 and s2. Write a program to create a new string s3 made of the first char of s1, then the last char of s2, Next, the second char of s1 and second last char of s2, and so on. Any leftover chars go at the end of the result.

```
'''
Given:
s1 = "Abc"
s2 = "Xyz"

Expected Output:
AzbycX
'''
s1=input("Enter first string: ")
s2=input("Enter second string: ")
s3=""
i, j = 0, len(s2)-1
while i<len(s1) or j>0:
```

```
  s3=s3+s1[i]
  i+=1
  s3=s3+s2[j]
  j-=1
print(s3)
```

```
Enter first string: Abc
Enter second string: Xyz
AzbycX
```

Exercise 7: String characters balance Test

Write a program to check if two strings are balanced. For example, strings s1 and s2 are balanced if all the characters in the s1 are present in s2. The character's position doesn't matter.

```
'''
Given:
Case 1:
s1 = "Yn"
s2 = "PYnative"

Expected Output:
True

Case 2:
s1 = "Ynf"
s2 = "PYnative"

Expected Output:
False
'''
s1=input("Enter first string: ")
s2=input("Enter second string: ")
flag=True
for ch in s1:
  if ch not in s2:
    flag=False
print(flag)
```

```
Enter first string: Ynf
Enter second string: PYnative
False
```

Exercise 8: Find all occurrences of a substring in a given string by ignoring the case

Write a program to find all occurrences of "USA" in a given string ignoring the case.

```
'''
Given:
str1 = "Welcome to USA. usa awesome, isn't it?"

Expected Outcome:
The USA count is: 2
'''
```

```
s1="Welcome to USA. usa awesome, isn't it?"
wlst=s1.lower().split()
cnt=0
for item in wlst:
  if "usa" in item:
    cnt+=1
print("The USA count is:", cnt)
```

```
The USA count is: 2
```

Exercise 9: Calculate the sum and average of the digits present in a string

Given a string s1, write a program to return the sum and average of the digits that appear in the string, ignoring all other characters.

```
'''
Given:
str1 = "PYnative29@#8496"

Expected Outcome:
Sum is: 38 Average is 6.333333333333333
'''
str1 = "PYnative29@#8496"
sum,cnt=0,0
for ch in str1:
  if ch.isdigit():
    sum=sum+int(ch)
    cnt+=1
print("Sum is:",sum, "Average is", sum/cnt)
```

```
Sum is: 38 Average is 6.333333333333333
```

Exercise 10: Write a program to count occurrences of all characters within a string

```
'''
Given:
str1 = "Apple"

Expected Outcome:
{'A': 1, 'p': 2, 'l': 1, 'e': 1}
'''
str1 = "Apple"
chlst=[]
cntd={}
for ch in str1:
  chlst.append(ch)
set1=set(chlst)
for se in sorted(set1):
  cntd.update({se:chlst.count(se)})
print(cntd)
```

```
{'A': 1, 'e': 1, 'l': 1, 'p': 2}
```

Exercise 11: Reverse a given string

```
'''
Given:
str1 = "PYnative"

Expected Output:
evitanYP
'''
str1 = "PYnative"
print(str1[::-1])
```

```
evitanYP
```

Exercise 12: Find the last position of a given substring

Write a program to find the last position of a substring “Emma” in a given string.

```
'''
Given:
str1 = "Emma is a data scientist who knows Python. Emma works at google."

Expected Output:
Last occurrence of Emma starts at index 43
'''
str1 = "Emma is a data scientist who knows Python. Emma works at google."
str2="Emma"
print(str1.rfind("Emma"))
```

```
43
```

Exercise 13: Split a string on hyphens Write a program to split a given string on hyphens and display each substring.

```
'''
Given:
str1 = Emma-is-a-data-scientist

Expected Output:
Displaying each substring
Emma
is
a
data
scientist
'''
str1 = "Emma-is-a-data-scientist"
wlst=str1.split("-")
for word in wlst:
  print(word)
```

```
Emma
is
a
```

```
data
scientist
```

Exercise 14: Remove empty strings from a list of strings

```
'''
Given:
str_list = ["Emma", "Jon", "", "Kelly", None, "Eric", ""]

Expected Output:
Original list of sting
['Emma', 'Jon', '', 'Kelly', None, 'Eric', '']

After removing empty strings
['Emma', 'Jon', 'Kelly', 'Eric']
'''
str_list = ["Emma", "Jon", "", "Kelly", None, "Eric", ""]
new_lst=[]
for item in str_list:
  if item!=None:
    if len(item)>0:
      new_lst.append(item)
print(new_lst)
```

```
['Emma', 'Jon', 'Kelly', 'Eric']
```

Exercise 15: Remove special symbols / punctuation from a string

```
'''
Given:
str1 = "/*Jon is @developer & musician"

Expected Output:
"Jon is developer musician"
'''
str1 = "/*Jon is @developer & musician"
str2=""
wlst=str1.split()
for ele in wlst:
  for ch in ele:
    if ch.isalnum()==False:
      str2=str2+""
    else:
      str2=str2+ch
  str2=str2+" "
print(str2)
```

```
Jon is developer  musician
```

Exercise 16: Removal all characters from a string except integers

```
'''
Given:
str1 = 'I am 25 years and 10 months old'

Expected Output:
2510
'''
str1 = 'I am 25 years and 10 months old'
str2=""
for ch in str1:
  if ch.isdigit():
    str2=str2+ch
print(str2)
```

```
2510
```

Exercise 17: Find words with both alphabets and numbers

Write a program to find words with both alphabets and numbers from an input string.

```
'''
Given:
str1 = "Emma25 is Data scientist50 and AI Expert"

Expected Output:
Emma25
scientist50
'''
str1 = "Emma25 is Data scientist50 and AI Expert"
lst=str1.split()
for word in lst:
  alpchk, numchk=False, False
  for ch in word:
    if ch.isalnum():
      alpchk=True
      break
  for ch in word:
    if ch.isdigit():
      numchk=True
      break
  if alpchk and numchk:
    print(word)
```

```
Emma25
scientist50
```

Exercise 18: Replace each special symbol with # in the following string

```
'''
Given:
str1 = '/*Jon is @developer & musician!!'

Expected Output:
```

```
##Jon is #developer # musician##
'''
str1 = '/*Jon is @developer & musician!!'
str2=""
wlst=str1.split()
for ele in wlst:
  for ch in ele:
    if ch.isalnum()==False:
      str2=str2+"#"
    else:
      str2=str2+ch
  str2=str2+" "
print(str2)
```

```
##Jon is #developer # musician##
```

```

```