

1. Given a string of odd length greater than 7, return a new string made of the middle three characters of a given String

Given:

```
str1 = "RakeshzipPetabb"
```

Output

zip

```
str2 = "JazzbonAyxx"
```

Output

bon

2. Given two strings, s1 and s2, create a new string by appending s2 in the middle of s1

Given:

```
s1 = "Ault"
```

```
s2 = "Kelly"
```

Expected Output:

AuKellylt

3. two strings, s1, and s2 return a new string made of the first, middle, and last characters each input string

Given:

```
s1 = "America"
```

```
s2 = "Japan"
```

Expected Output:

AJrpan

solution:

```
def mix_string(s1, s2):
```

```
    first_char = s1[:1] + s2[:1]
```

```
    middle_char = s1[int(len(s1) / 2):int(len(s1) / 2) + 1] + s2[int(len(s2) / 2):int(len(s2) / 2) + 1]
```

```
    last_char = s1[len(s1) - 1] + s2[len(s2) - 1]
```

```
    res = first_char + middle_char + last_char
```

```
    print("Mix String is ", res)
```

```
s1 = "America"
s2 = "Japan"
mix_string(s1, s2)
```

4. Given an input string with the combination of the lower and upper case arrange characters in such a way that all lowercase letters should come first.

solution:

```
str1 = "PytHONStudy"
lower = []
upper = []
for char in str1:
    if char.islower():
        lower.append(char)
    else:
        upper.append(char)
sorted_string = ''.join(lower + upper)
print("arranging characters giving precedence to lowercase letters:")
print(sorted_string)
```

5. create a third-string 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
```

6. Find all occurrences of "USA" from right to left in a given string ignoring the case. also display the position

Given:

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

