Question 6:

Recommended time: 5 min

Print characters from a string that are present at an even index number

Write a program to accept a string from the user and display characters that are present at an even index number.

For example, `str = "pynative"` so you should display ‘p’, ‘n’, ‘t’, ‘v’.

Expected Output:

```
Orginal String is  pynative
Printing only even index chars
p
n
t
v
```

Question 7:

Recommended time: 5 min

Remove first `n` characters from a string

Write a program to remove characters from a string starting from zero up to `n` and return a new string.

For example:

- `remove_chars("pynative", 4)` so output must be `tive`. Here we need to remove first four characters from a string.
- `remove_chars("pynative", 2)` so output must be `native`. Here we need to remove first two characters from a string.

Note: `n` must be less than the length of the string.

Question 8:

Recommended time: 5 min

Check if the first and last number of a list is the same

Write a function to return `True` if the first and last number of a given list is same. If numbers are different then return `False`.

Given:

```
numbers_x = [10, 20, 30, 40, 10]
numbers_y = [75, 65, 35, 75, 30]
```

Expected Output:

```
Given list: [10, 20, 30, 40, 10]
result is True

numbers_y = [75, 65, 35, 75, 30]
result is False
```

Question 9:

Recommended time: 7 min

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
```

Question 10:

Recommended time: 10 min

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
```

Question 11:

Recommended time: 5 min

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
```