1. >>> help(str.replace)
Help on method_descriptor:
replace(...)  
S.replace(old, new[, count]) -> str  

Return a copy of S with all occurrences of substring old replaced by new. If the optional argument count is given, only the first count occurrences are replaced.

What is the output of this code?

```python
s = 'Mississippi'
t = len(s.replace('ss', 'a'))
print(t)
```

A 11  
B ss  
C 9  
D Miaiaaippi

Recall that len returns the length of its argument, so t must be an integer. Here all copies of “ss” in “Mississippi” are replaced with “a” to create “Miaiaippi”, which is of length 9.

2. >>> help(str.replace)
Help on method_descriptor:
replace(...)  
S.replace(old, new[, count]) -> str  

Return a copy of S with all occurrences of substring old replaced by new. If the optional argument count is given, only the first count occurrences are replaced.

What is the output of this code?

```python
s = 'Mississippi'
t = len(s.replace('ss', 'a'))
print(t)
```

A 11  
B ss  
C 9  
D Miaiaaippi

3. >>> help(str.center)
Help on method_descriptor:
center(...)  
S.center(width[, fillchar]) -> str  

Return S centered in a string of length width. Padding is done using the specified fill character (default is a space)

What is the string produced by the following:

`'cave'.center(8, 'x')`

A `xxcavexx`
4. >>> help(str.center)
Help on method_descriptor:
center(...)  
S.center(width[, fillchar]) -> str
Return S centered in a string of length width. Padding is done using the specified fill character (default is a space)

What is the string produced by the following:
‘cave’.center(8, ‘x’)

A  ‘xxcavexx’
B  ‘cave ’
C  ‘xxxxcavexxx’
D  ‘xxxxxxxxcavexxxxxxx’

This question can be hard to answer without testing, here width is the total width, so the answer is 8 as adding "xx" before and after "cave" is required to get 8 characters total.