Experiment 6 :

Aim :- Write a Program to check if a substring is present in a given string.

Theory:-

Python string contains another string or a substring in Python. Given two strings, check if a substring is there in the given string or not.

Does Python have a string containing the substring method
Yes, Checking a substring is one of the most used tasks in python. Python uses many methods to check a string containing a substring like, find(), index(), count(), etc. The most efficient and fast method is by using an "in" operator which is used as a comparison operator. different approaches for check if a substring is present in a given string :

Using the if... in
Checking using the split() method
Using find() method
Using "count()" method
Using the index() method
Using __contains__" magic class.
Using regular expressions

Method 1: Check substring using the if... in.

```
# Take input from users
MyString1 = "A geek in need is a geek indeed"

if "need" in MyString1:
   print("Yes! it is present in the string")
else:
   print("No! it is not present")
```

Output
Yes! it is present in the string

Output
yes

Method 3: Check substring using the find() method
We can iteratively check for every word, but Python provides us an inbuilt function find() which checks if a substring is present in the string, which is done in one line. find() function returns -1 if it is not found, else it returns the first occurrence, so using this function this problem can be solved.

```
# function to check if small string is
# there in big string


def check(string, sub_str):
    if (string.find(sub_str) == -1):
        print("NO")
    else:
        print("YES")


# driver code
string = "ncer talegaon pune"
sub_str = "pune"
check(string, sub_str)
```

Output
YES

Method 4: Check substring using "count()" method
You can also count the number of occurrences of a specific substring in a string, then you can use the Python count() method. If the substring is not found then "yes " will print otherwise "no will be printed".

```
def check(s2, s1):
    if (s2.count(s1) > 0):
        print("YES")
    else:
        print("NO")


s2 = "Allow the User to input the Width and Height"
s1 = "new"
check(s2, s1)
```

Output
NO

Method 5: Check substring using the index() method
The .index() method returns the starting index of the substring passed as a parameter. Here "substring" is present at index 16.

```
any_string = "Allow the User to input the Width and Height "
start = 0
end = 1000
print(any_string.index('substring', start, end))
```

Output:
16

Metho 6: Using slicing
This implementation uses a loop to iterate through every possible starting index of the substring in the string, and then uses slicing to compare the current substring to the substring argument.

If the current substring matches the substring argument, then the function returns True. If the substring is not found after checking all possible starting indices, then the function returns False.

```
def is_substring(string, substring):
    for i in range(len(string) - len(substring) + 1):
        if string[i:i+len(substring)] == substring:
            return True
    return False
string = "Allow the User to input the Width and Height"
substring = " User "
print(is_substring(string,substring))
```

Output
True