In [3]:

```
##Answer 1:

def rev_sentence(sentence):
    words = sentence.split(' ')
    reverse_sentence = ' '.join(reversed(words))
    return reverse_sentence
if __name__ == "__main__":
    input = 'perfect man a makes Practice'
    print (rev_sentence(input))
```

```
Practice makes a man perfect
```

In [5]:

```
##Answer 2:

test_str = "Bicycle"
print ("The original string is : " + test_str)
new_str = ""
for i in range(len(test_str)):
    if i != 2:
        new_str = new_str + test_str[i]
print ("The string after removal of i'th character : " + new_str)
```

```
The original string is : Bicycle
The string after removal of i'th character : Biycle
```

In [7]:

```
##Answer 3:

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

s2 = "A friend in need is a friend indeed"
s1 ="friend"
check(s2, s1)
```

```
YES
```

In [9]:

```
##Answer 4:

test_str = "A friend in need is a friend indeed"
print("The original string is : " + str(test_str))
res = {key: test_str.count(key) for key in test_str.split()}
print("The words frequency : " + str(res))
```

```
The original string is : A friend in need is a friend indeed
The words frequency : {'A': 1, 'friend': 2, 'in': 2, 'need': 1, 'is': 1, 'a': 1, 'indee
d': 1}
```

In [11]:

```
##Answer 5:

test_str = "A friend in need_is_a friend indeed"
print("The original string is : " + test_str)
```

```
res = test_str.replace("_", " ").title().replace(" ", "")
print("The String after changing case : " + str(res))
```

```
The original string is : A friend in need_is_a friend indeed
The String after changing case : AFriendInNeedIsAFriendIndeed
```

In [12]:

```
##Answer 6.1:

str = "Hello World"
print(len(str))
```

```
11
```

In [13]:

```
##Answer 6.2:

def findLen(str):
    counter = 0
    for i in str:
        counter += 1
    return counter
str = "Hello World"
print(findLen(str))
```

```
11
```

In [16]:

```
##Answer 6.3:

def findLen(str):
    counter = 0
    while str[counter:]:
        counter += 1
    return counter
str = "Hello World"
print(findLen(str))
```

```
11
```

In [17]:

```
##Answer 6.4:

def findLen(str):
    if not str:
        return 0
    else:
        some_random_str = 'py'
        return ((some_random_str).join(str)).count(some_random_str) + 1

str = "Hello World"
print(findLen(str))
```

```
11
```

In [18]:

```
##Answer 7:

def printWords(s):
    s = s.split(' ')
    for word in s:
        if len(word)%2==0:
            print(word)
s = "I am an Indian"
printWords(s)
```

```
am
an
Indian
```

In [19]:

```
##Answer 8:

def check(string):
    if len(set(string.lower()).intersection("aeiou")) >= 5:
        return ('accepted')
    else:
        return ("not accepted")
if __name__ == "__main__":
    string = "EUTOPIA"
    print(check(string))
```

```
accepted
```

In [21]:

```
##Answer 9:

def count(str1, str2):
    c, j = 0, 0
    for i in str1:
        if str2.find(i)>= 0 and j == str1.find(i):
            c += 1
        j += 1
    print ('No. of matching characters are : ', c)
def main():
    str1 ='Hello World'
    str2 ='Hello India'
    count(str1, str2)
if __name__=="__main__":
    main()
```

```
No. of matching characters are :  6
```

In [23]:

```
##Answer 10:



def removeDuplicate(str):
    s=set(str)
    s="".join(s)
    print("Without Order:",s)
    t=""
    for i in str:
        if(i in t):
            pass
        else:
            t=t+i
        print("With Order:",t)

str="A friend in need is a friend indeed"
removeDuplicate(str)
```

```
Without Order: andrif Aes
```

```
With Order: A
With Order: A
With Order: A f
With Order: A fr
With Order: A fri
With Order: A frie
With Order: A frien
With Order: A friend
With Order: A friend
With Order: A friend
With Order: A friend
With Order: A friend
With Order: A friend
With Order: A friend
With Order: A friend
With Order: A friend
With Order: A friend
With Order: A friend
With Order: A friends
With Order: A friends
With Order: A friendsa
With Order: A friendsa
With Order: A friendsa
With Order: A friendsa
With Order: A friendsa
With Order: A friendsa
With Order: A friendsa
With Order: A friendsa
With Order: A friendsa
With Order: A friendsa
With Order: A friendsa
With Order: A friendsa
With Order: A friendsa
With Order: A friendsa
With Order: A friendsa
```