Q1) Write a Python program to get a list, sorted in increasing order by the last element in each tuple from a given list of non-empty tuples. Sample List : [(2, 5), (1, 2), (4, 4), (2, 3), (2, 1)]

In [1]:
```
l = [(2, 5), (1, 2), (4, 4), (2, 3), (2, 1)]
print("The sorrted List : ", sorted(l, key = lambda x : x[-1]))
```

```
The sorrted List :  [(2, 1), (1, 2), (2, 3), (4, 4), (2, 5)]
```

Q2) Write a Python program to get a string from a given string where all occurrences of its first char have been changed to '$', except the first char itself. Sample String : 'restart'

In [2]:
```
def replacechar(s):
    c = s[0]
    s = s.replace(c, '$')
    s = c + s[1:]
    return s

str = "restart"

print(replacechar(str))
```

```
resta$t
```

Q3) Write a Python program to find the first appearance of the substring 'not' and 'poor' from a given string, if 'bad' follows the 'poor', replace the whole 'not'...'poor' substring with 'good'. Return the resulting string.

In [3]:
```
def not_poor(text):
    index_not = text.find('not')
    index_poor = text.find('poor')

    if index_poor > index_not and index_not > 0 and index_poor > 0:
        text = text.replace(text[index_not:(index_poor+4)], 'good')
        return text
    else:
        return text

print(not_poor('The lyrics is not that poor!'))
print(not_poor('The lyrics is poor!'))
```

```
The lyrics is good!
The lyrics is poor!
```

Q4) Write a python program to sort a dictionary by value.

In [4]:
```
my_dict = {'a': 10, 'b': 2, 'c': 25, 'd': 7}
print("The sorrted Dict : ", sorted(my_dict.items(), key = lambda x : x[-1]))
```

```
The sorrted Dict :  [('b', 2), ('d', 7), ('a', 10), ('c', 25)]
```

Q5) Write a python program to add key to a dictionary.

In [5]:
```
my_dict = {0: 10, 1: 20}

key = input("Enter the key: ")
value = input("Enter the value: ")
```

```
if key.isdigit():
    key = int(key)

if value.isdigit():
    value = int(value)

my_dict[key] = value

print("Updated dictionary:", my_dict)
```

```
Updated dictionary: {0: 10, 1: 20, 3: 42}
```

In [ ]:

In [ ]: