# Assignment No.2

```
class SetADT:
    def __init__(self):
        self.elements = set()

    def add(self, element):
        self.elements.add(element)

    def remove(self, element):
        if element in self.elements:
            self.elements.remove(element)
        else:
            raise ValueError(f"Element {element} not found in the set.")

    def contains(self, element):
        return element in self.elements

    def size(self):
        return len(self.elements)

    def iterator(self):
        return iter(self.elements)

    def intersection(self, other_set):
        return SetADT.from_elements(self.elements.intersection(other_set.elements))

    def union(self, other_set):
        return SetADT.from_elements(self.elements.union(other_set.elements))
    def difference(self, other_set):
        return SetADT.from_elements(self.elements.difference(other_set.elements))

    def is_subset(self, other_set):
        return self.elements.issubset(other_set.elements)

    @classmethod
    def from_elements(cls, elements):
        new_set = cls()
        new_set.elements = elements
        return new_set

    def __repr__(self):
        return f"SetADT({self.elements})"


set1 = SetADT()
set1.add(10)
set1.add(20)
set1.add(30)

set2 = SetADT()
set2.add(30)
set2.add(40)
set2.add(50)

# Add element
print(set1.contains(20))

# Remove element
set1.remove(20)
```

```
print(set1.contains(20))

# size
print(set1.size())

# Iterate
for element in set1.iterator():
    print(element)

# Intersection
intersection_set = set1.intersection(set2)
print(intersection_set)

# Union
union_set = set1.union(set2)
print(union_set)

# Difference
difference_set = set1.difference(set2)
print(difference_set)

# Check subset
print(set1.is_subset(set2))
```

Output-

```
True
False
2
10
30
```

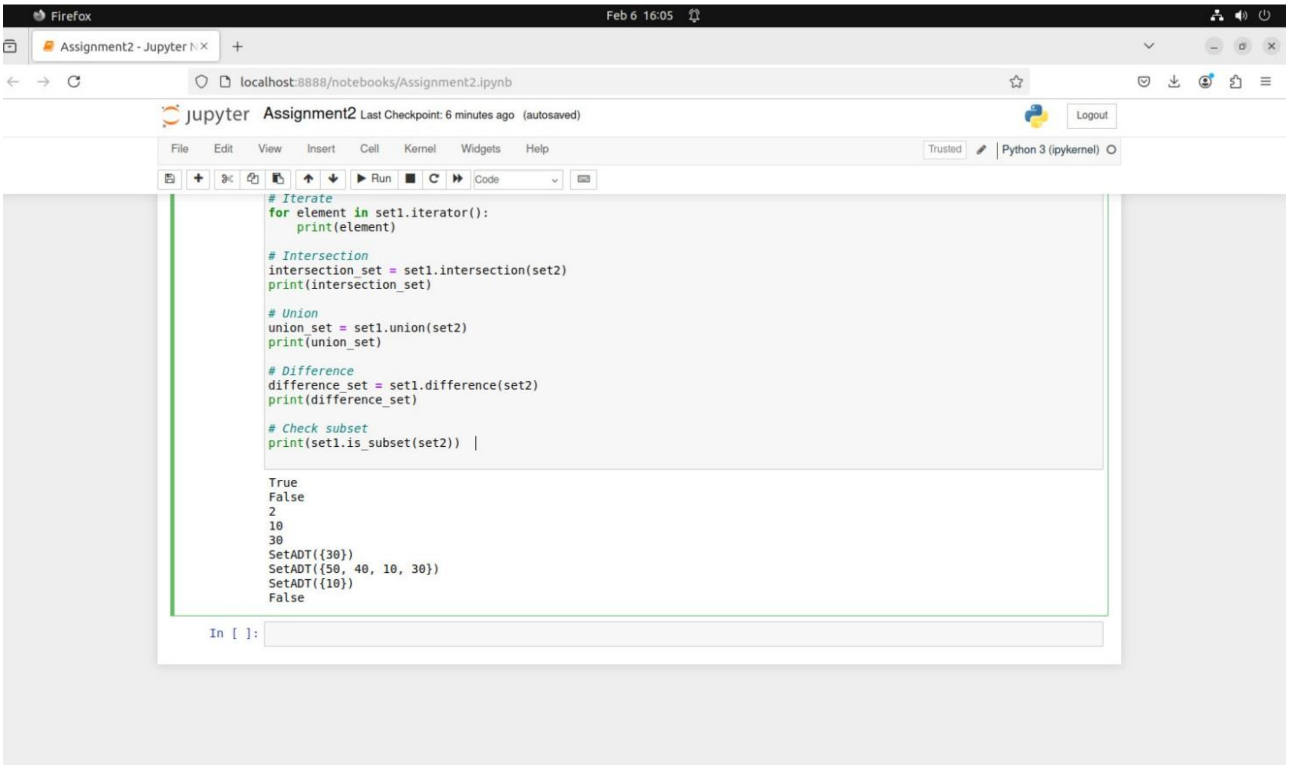
SetADT({30})

SetADT({50, 40, 10, 30})

SetADT({10})

False

## Output :



```
# Iterate
for element in set1.iterator():
    print(element)

# Intersection
intersection_set = set1.intersection(set2)
print(intersection_set)

# Union
union_set = set1.union(set2)
print(union_set)

# Difference
difference_set = set1.difference(set2)
print(difference_set)

# Check subset
print(set1.is_subset(set2))
```

```
True
False
2
10
30
SetADT({30})
SetADT({50, 40, 10, 30})
SetADT({10})
False
```

```
In [ ]:
```