task-1

```
In [4]: mylist = ['hobby', 'hobby', 'hobby', 'hobby', 'hobby', 'hobby', 'hobby', 'hobby
        mylist
```

```
Out[4]: ['hobby',
         'hobby',
         'hobby',
         'hobby',
         'hobby',
         'hobby',
         'hobby',
         'hobby',
         'hobby',
         'hobby']
```

task-2

```
In [16]: matriculation_marks = [95, 92, 87, 93, 88]
         intermediate_marks = [85, 90, 92, 88, 91]

         marks = [matriculation_marks, intermediate_marks]
         marks
```

```
Out[16]: [[95, 92, 87, 93, 88], [85, 90, 92, 88, 91]]
```

task-3

```
In [14]: mylist = ['hobby0', 'hobby1', 'hobby2', 'hobby3', 'hobby4', 'hobby5', 'hobby6',
         mylist[3]
```

```
Out[14]: 'hobby3'
```

task-4

In [22]:

```
matriculation_marks = [95, 92, 87, 93, 88]
intermediate_marks = [85, 90, 92, 88, 91]

marks = [matriculation_marks, intermediate_marks]


print("Intermediate marks:")
for index, mark in enumerate(marks[1]):
    print("Subject", index+1, ":", mark)


all_marks = matriculation_marks + intermediate_marks
highest_marks = max(all_marks)
print("\nHighest marks in your educational career:", highest_marks)


lowest_marks = min(all_marks)
print("Lowest marks in your educational career:", lowest_marks)
```

```
Intermediate marks:
Subject 1 : 85
Subject 2 : 90
Subject 3 : 92
Subject 4 : 88
Subject 5 : 91

Highest marks in your educational career: 95
Lowest marks in your educational career: 85
```

task-5

In [31]:

```
def is_prime(num):
    if num < 2:
        return False
    for i in range(2, int(num ** 0.5) + 1):
        if num % i == 0:
            return False
    return True

my_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 0]
even_numbers = []
odd_numbers = []
prime_numbers = []

for num in my_list:
    if num % 2 == 0:
        even_numbers.append(num)
    else:
        odd_numbers.append(num)

    if is_prime(num):
        prime_numbers.append(num)

print("Even numbers:", even_numbers)
print("Odd numbers:", odd_numbers)
print("Prime numbers:", prime_numbers)
```

```
Even numbers: [2, 4, 6, 8, 0]
Odd numbers: [1, 3, 5, 7, 9]
Prime numbers: [2, 3, 5, 7]
```

task-6

In [32]:

```
siblings = {
    "older_sister": "Emma",
    "older_brother": "Jacob",
    "younger_sister": "Olivia",
    "younger_brother": "Noah"
}
siblings
```

Out[32]:

```
{'older_sister': 'Emma',
 'older_brother': 'Jacob',
 'younger_sister': 'Olivia',
 'younger_brother': 'Noah'}
```

task-7

In [41]:

```
scoreboard = {
"jamal": {
1: {0, "WD", "No", "free hit+6", "catch", 4, "bowled", 6, 0},
2: {4,4,4,"wicket","wicket","wicket"},
3: {0,0,0,0,1,0},
4:{"out","NO+FREEHIT",0,0,6,6,"catch"}
},
"hamza": {
1: {0, "WD", "No", "free hit+6", 0, 4, "bowled", 6, 0},
2: {4,4,4,"wicket","wicket","wicket"},
3: {0,0,0,0,1,0},
4:{"out","NO+FREEHIT",0,"catch",6,6,"out"}

}
}

print('wickets taken by jamal: ')
j_w = sum(1 for wickets in scoreboard["jamal"].values if "wicket" in wickets)
j_w
```

```
wickets taken by jamal: 

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[41], line 18
      1 scoreboard = {
      2 "jamal": {
      3 1: {0, "WD", "No", "free hit+6", "catch", 4, "bowled", 6, 0},
   (...)
     14 }
     15 }
     17 print('wickets taken by jamal: ')
---> 18 j_w = sum(1 for wickets in scoreboard["jamal"].values if "wicket" in 
wickets)
     19 j_w

TypeError: 'builtin_function_or_method' object is not iterable
```

In [42]:

```
jamal_wickets = sum(1 for over in scoreboard["jamal"].values() if "wicket" in c
print("Wickets taken by Jamal:", jamal_wickets)
```

```
Wickets taken by Jamal: 1
```

In [43]:
```
scoreboard = {
    "jamal": {
        1: {0, "WD", "No", "free hit+6", "catch", 4, "bowled", 6, 0},
        2: {4, 4, 4, "wicket", "wicket", "wicket"},
        3: {0, 0, 0, 0, 1, 0},
        4: {"out", "NO+FREEHIT", 0, 0, 6, 6, "catch"}
    },
    "hamza": {
        1: {0, "WD", "No", "free hit+6", 0, 4, "bowled", 6, 0},
        2: {4, 4, 4, "wicket", "wicket", "wicket"},
        3: {0, 0, 0, 0, 1, 0},
        4: {"out", "NO+FREEHIT", 0, "catch", 6, 6, "out"}
    }
}

# How many wickets taken by Jamal
jamal_wickets = sum(1 for over in scoreboard["jamal"].values() if "wicket" in o
print("Wickets taken by Jamal:", jamal_wickets)

# How many wickets taken by Hamza
hamza_wickets = sum(1 for over in scoreboard["hamza"].values() if "wicket" in o
print("Wickets taken by Hamza:", hamza_wickets)

# Overall total wickets taken in the match
total_wickets = jamal_wickets + hamza_wickets
print("Total wickets taken in the match:", total_wickets)

# Overall "WD" and "NO" and total score of the match
wd_count = sum(1 for over in scoreboard.values() for ball in over.values() if b
no_count = sum(1 for over in scoreboard.values() for ball in over.values() if b
total_score = sum(sum(1 for ball in over if isinstance(ball, int)) for over in 
print("Total 'WD' count:", wd_count)
print("Total 'NO' count:", no_count)
print("Total score of the match:", total_score)

# Score given by Jamal
jamal_score = sum(sum(ball for ball in over if isinstance(ball, int)) for over 
print("Score given by Jamal:", jamal_score)

# Score given by Hamza
hamza_score = sum(sum(ball for ball in over if isinstance(ball, int)) for over 
print("Score given by Hamza:", hamza_score)

# Score given in the 2nd over of Hamza and Jamal
jamal_2nd_over_score = sum(ball for ball in scoreboard["jamal"][2] if isinstanc
hamza_2nd_over_score = sum(ball for ball in scoreboard["hamza"][2] if isinstanc
print("Score given in the 2nd over of Jamal:", jamal_2nd_over_score)
print("Score given in the 2nd over of Hamza:", hamza_2nd_over_score)
```

```
Wickets taken by Jamal: 1
Wickets taken by Hamza: 1
Total wickets taken in the match: 2
Total 'WD' count: 0
Total 'NO' count: 0
Total score of the match: 8
Score given by Jamal: 21
Score given by Hamza: 21
Score given in the 2nd over of Jamal: 4
Score given in the 2nd over of Hamza: 4
```

In [ ]: