# *ICP1 REPORT*

```
[6]  char_list = list("PYTHON")
     del char_list[0]
     del char_list[1]
     char_list.reverse()
     resultant_string = ''.join(char_list)
     print(resultant_string)
```

```
NOHY
```

```
n1 = float(input("first number is: "))
n2 = float(input("second number is: "))
add = n1 + n2
subtract = n1 - n2
multiplicate = n1 * n2
division = n1 / n2
print(f"Add: {n1} + {n2} = {add}")
print(f"Subtract: {n1} - {n2} = {subtract}")
print(f"Multiplicate: {n1} * {n2} = {multiplicate}")
print(f"Division: {n1} / {n2} = {division}")
```

```
first number is: 10
second number is: 12
Add: 10.0 + 12.0 = 22.0
Subtract: 10.0 - 12.0 = -2.0
Multiplicate: 10.0 * 12.0 = 120.0
Division: 10.0 / 12.0 = 0.8333333333333334
```

```
(variable) updated_sentence: str
: ")
updated_sentence = sentence.replace('python', 'pythons
print("Updated sentence:", updated_sentence)
```

```
Enter a sentence: python programming
Updated sentence: pythons programming
```

```python
score = float(input("Enter the class score (0-100): "
if 90 <= score <= 100:
    grade = 'A'
elif 80 <= score < 90:
    grade = 'B'
elif 70 <= score < 80:
    grade = 'C'
elif 60 <= score < 70:
    grade = 'D'
elif 0 <= score < 60:
    grade = 'F'
else:
    grade = 'Invalid score'
print(f"The letter grade is: {grade}")
```

```
Enter the class score (0-100): 79
The letter grade is: C
```

```python
x = [1, 'Python', 1.18]
types_list = [type(element) for element in x]
print(x)
print(types_list)
```

```
[1, 'Python', 1.18]
[<class 'int'>, <class 'str'>, <class 'float'>]
```

```python
IT_companies = {'Facebook', 'Google', 'Microsoft', 'Apple', 'IBM', 'Oracle', 'Amazon'}
length_of_IT_companies = len(IT_companies)
print("The length of the set IT_companies is:", length_of_IT_companies)
```

```
The length of the set IT_companies is: 7
```

```
(variable) IT_companies: set[str]
```

```python
IT_companies = {'Facebook', 'Google', 'Microsoft', 'Apple', 'IBM', 'Oracle', 'Amazon'}
IT_companies.add('Twitter')
print("Updated IT_companies set:", IT_companies)
```

```
Updated IT_companies set: {'IBM', 'Twitter', 'Oracle', 'Apple', 'Facebook', 'Microsoft', 'Amazon', 'Google'}
```

```python
IT_companies = {'Facebook', 'Google', 'Microsoft', 'Apple', 'IBM', 'Oracle', 'Amazon'}
new_companies = {'Twitter', 'Netflix', 'Salesforce', 'Adobe'}
IT_companies.update(new_companies)
print("Updated IT_companies set:", IT_companies)
```

```
Updated IT_companies set: {'IBM', 'Twitter', 'Oracle', 'Salesforce', 'Amazon', 'Google', 'Netflix', 'Apple', 'Facebook', 'Microsoft', 'Adobe'}
```

```python
IT_companies = {'Facebook', 'Google', 'Microsoft', 'Apple', 'IBM', 'Oracle', 'Amazon'}
IT_companies.remove('Google')
print("Updated IT_companies set after removal using remove():", IT_companies)
```

```
Updated IT_companies set after removal using remove(): {'IBM', 'Oracle', 'Apple', 'Facebook', 'Microsoft', 'Amazon'}
```

```python
IT_companies = {'Facebook', 'Google', 'Microsoft', 'Apple', 'IBM', 'Oracle', 'Amazon'}
IT_companies.remove('IBM')
IT_companies.discard('Oracle')
IT_companies.discard('NonExistent')
print(IT_companies)
```

```python
A = {19, 22, 24, 20, 25, 26}
B = {19, 22, 20, 25, 26, 24, 28, 27}
joined_set = A.union(B)
print("Joined set using union():", joined_set)
```

```
Joined set using union(): {19, 20, 22, 24, 25, 26, 27, 28}
```

+ Code + Text

```python
A = {19, 22, 24, 20, 25, 26}
B = {19, 22, 20, 25, 26, 24, 28, 27}
intersection_AB = A.intersection(B)
print("Intersection of A and B using intersection():", intersection_AB)
```

```
Intersection of A and B using intersection(): {19, 20, 22, 24, 25, 26}
```

```python
A = {19, 22, 24, 20, 25, 26}
B = {19, 22, 20, 25, 26, 24, 28, 27}
is_subset = A.issubset(B)
print("Is A a subset of B using issubset():", is_subset)
```

```
Is A a subset of B using issubset(): True
```

```python
A = {19, 22, 24, 20, 25, 26}
B = {19, 22, 20, 25, 26, 24, 28, 27}
are_disjoint = A.isdisjoint(B)
print("Are A and B disjoint sets using isdisjoint()?", are_disjoint)
```

```
Are A and B disjoint sets using isdisjoint()? False
```

```python
A = {19, 22, 24, 20, 25, 26}
B = {19, 22, 20, 25, 26, 24, 28, 27}
join_A_with_B = A.union(B)
print("Join A with B:", join_A_with_B)
```

Connected to Python 3 Google Compute Engine backend

```python
A = {19, 22, 24, 20, 25, 26}
B = {19, 22, 20, 25, 26, 24, 28, 27}
join_B_with_A = B.union(A)
print("Join B with A:", join_B_with_A)
```

```
Join B with A: {19, 20, 22, 24, 25, 26, 27, 28}
```

```python
A = {19, 22, 24, 20, 25, 26}
B = {19, 22, 20, 25, 26, 24, 28, 27}
symmetric_diff = A.symmetric_difference(B)
print("Symmetric difference between A and B using symmetric_difference():", symmetric_diff)
```

(method) symmetric_difference: (__s: Iterable[int], /) -> set[int]

```
Symmetric difference between A and B using symmetric_difference(): {27, 28}
```

```python
A = {19, 22, 24, 20, 25, 26}
B = {19, 22, 20, 25, 26, 24, 28, 27}
print("Before deletion:")
print("A:", A)
print("B:", B)
del A
del B
try:
    print("After deletion:")
    print("A:", A)
except NameError as e:
    print("Error:", e)

try:
    print("B:", B)
except NameError as e:
    print("Error:", e)
```

```
Before deletion:
A: {19, 20, 22, 24, 25, 26}
```

Connected to Python 3 Google Compute Engine backend

```
    print("Error:", e)
```

```
Before deletion:
A: {19, 20, 22, 24, 25, 26}
B: {19, 20, 22, 24, 25, 26, 27, 28}
After deletion:
Error: name 'A' is not defined
Error: name 'B' is not defined
```

```python
age = [22, 19, 24, 25, 26, 24, 25, 24]
age_set = set(age)
list_length = len(age)
set_length = len(age_set)
print("Length of the list:", list_length)
print("Length of the set:", set_length)
print("Set created from list:", age_set)
```

```
Length of the list: 8
Length of the set: 5
Set created from list: {19, 22, 24, 25, 26}
```

***My YouTube link: https://youtu.be/oXHkAcBzqGg***

***GitHub Repository link:***
***https://github.com/nithin1086/BDA.git***