Experiment 2

Question 1:

- Read an array and display.
- Append a new item to the end of the array.
- To reverse the order of the items in the array (slice operator).
- Get the length in bytes of one array item.
- To append items from another array.
- Remove a specific item from the array.
- Add an item at a specific index in an array.
- Convert array to a string.

Theory

What are arrays?

An array is a collection of items stored at contiguous memory locations. The idea is to store multiple items of the same type together. This makes it easier to calculate the position of each element by simply adding an offset to a base value, i.e., the memory location of the first element of the array (generally denoted by the name of the array).

How to create an array in Python?

To create an array of numeric values, we need to import the array module. For example:

```
import array as arr
a = arr.array('d', [1.1, 3.5, 4.5])
print(a)
```

Dheeraj Lalwani
1902085

How to access Python Array Elements?

We use indices to access elements of an array.
For example:

```
import array as arr
a = arr.array('i', [2, 4, 6, 8])
print("First element:", a[0])
print("Second element:", a[1])
print("Last element:", a[-1])
```

How to slice Python Arrays?

We can access a range of items in an array by using the slicing operator ':'
For example:

```
import array as arr
numbers_list = [2, 5, 62, 5, 42, 52, 48, 5]
numbers_array = arr.array('i', numbers_list)
print(numbers_array[2:5]) # 3rd to 5th
print(numbers_array[:-5]) # beginning to 4th
print(numbers_array[5:])  # 6th to end
print(numbers_array[:])   # beginning to end
```

How to add and change elements in Python Arrays?

Arrays are mutable; their elements can be changed in a similar way as lists. We can add one item to the array using the `append()` method, or add several items using the `extend()` method. We can also concatenate two arrays using + operator.
For example:

```
import array as arr
numbers = arr.array('i', [1, 2, 3, 5, 7, 10])
numbers[0] = 0
print(numbers)
numbers[2:5] = arr.array('i', [4, 6, 8])
print(numbers)
numbers = arr.array('i', [1, 2, 3])
numbers.append(4)
print(numbers)
numbers.extend([5, 6, 7])
print(numbers)
odd = arr.array('i', [1, 3, 5])
even = arr.array('i', [2, 4, 6])
numbers = arr.array('i')
numbers = odd + even
print(numbers)
```

How to remove elements from the array?

Elements can be removed from the array by using built-in remove() function but an Error arises if element doesn't exist in the set. Remove() method only removes one element at a time, to remove range of elements, iterator is used. pop() function can also be used to remove and return an element from the array, but by default it removes only the last element of the array, to remove element from a specific position of the array, index of the element is passed as an argument to the pop() method. We can also delete one or more items from an array using Python's del statement.

```
import array as arr
numbers = arr.array('i', [10, 11, 12, 12, 13])
numbers.remove(12)
print(numbers)
# Output: array('i', [10, 11, 12, 13])


print(numbers.pop(2))   # Output: 12
print(numbers)
# Output: array('i', [10, 11, 13])


number = arr.array('i', [1, 2, 3, 3, 4])
del number[2]
print(number)
# Output: array('i', [1, 2, 3, 4])


del number  # deleting entire array
print(number)  # Error: array is not defined
```

Code

```
import array as arr

print("\n========== Read and Display an array ==========\n")

x = arr.array('i', [])

n = int(input("Enter number of elements: "))

for i in range(n):
    ele = int(input(f"Enter element array[{i + 1}]: "))
    x.append(ele)
print("The array :", end=" ")
for i in range(n):
    print(x[i], end=" ")

print("\n\n=============================================\n")
```

```
print("\n=============== Append new ===============\n")

ip = int(input("Enter an item to be appended: "))
x.append(ip)
for element in x:
    print(element, end=" ")

print("\n=========================================\n")


print("\n=============== Reverse Array ===============\n")

for i in x[::-1]:
    print(i, end=" ")

print("\n=============================================\n")

print("\n========== Length in Bytes of 1 item ==========\n")

print(f"Length in bytes of one array item: {str(x.itemsize)}")

print("\n=============================================\n")

print("\n======== Append items from another array ========\n")

y = arr.array('i', [7, 8, 9])
x.append(y[2])
print(x)

print("\n=============================================\n")

print("\n========== Inserting at a specific location ==========\n")

item = int(input("Enter an item to be appended: "))
pos = int(input(f"Enter the position where {item} is to be appended: "))
x.insert(pos, item)
print(x)

print("\n=============================================\n")

print("\n========== Array to string ==========\n")

print(' '.join(map(str, x)))

print("\n=====================================\n")
```

Dheeraj Lalwani
1902085

Output

```
========== Read and Display an array ==========

Enter number of elements: 7
Enter element array[1]: 7
Enter element array[2]: 5
Enter element array[3]: 9
Enter element array[4]: 0
Enter element array[5]: 3
Enter element array[6]: 1
Enter element array[7]: 6
The array : 7 5 9 0 3 1 6

===============================================
```

```
=============== Append new ===============

Enter an item to be appended: 8
7 5 9 0 3 1 6 8
==========================================
```

```
=============== Reverse Array ===============

8 6 1 3 0 9 5 7
==========================================
```

```
========== Length in Bytes of 1 item ==========

Length in bytes of one array item: 4

==========================================
```

```
======== Append items from another array ========

array('i', [7, 5, 9, 0, 3, 1, 6, 8, 9])

==========================================
```

```
========== Inserting at a specific location ==========

Enter an item to be appended: 77
Enter the position where 77 is to be appended: 4
array('i', [7, 5, 9, 0, 77, 3, 1, 6, 8, 9])

==========================================
```

```
========== Array to string ==========

7 5 9 0 77 3 1 6 8 9

=====================================
```

Dheeraj Lalwani
1902085

Question 2:

- Python program to remove prime numbers from an array

Code

```
import array as x

n = int(input("Enter number of elements: "))

nums = map(int, input(f"Enter {n} numbers in a single line with spaces in
between: ").split())

a = x.array('i', nums)

print("Array after removing primes:", end=" ")

b = x.array('i', [])
for s in a:
    for i in range(2, s):
        if(s % i==0):
            break;
        if(i==s-1):
            b.append(s)
for ss in b:
    while ss in a:
        a.remove(ss)

print(a)
```

Output

```
(venv) D:\Dheeraj\Study\SEM 4\PythonPracticals\Experiment2\code>python question2.py
Enter number of elements: 7
Enter 7 numbers in a single line with spaces in between: 7 4 5 9 13 6 42
Array after removing primes: array('i', [4, 9, 6, 42])
```

Dheeraj Lalwani
1902085

Question 3:

- Python program to change all occurrences of a first character of a string to @ except for the first occurrence.

Theory

What are Python Strings?

In Python, Strings are arrays of bytes representing Unicode characters. However, Python does not have a character data type, a single character is simply a string with a length of 1. Square brackets can be used to access elements of the string.

How to create a string in Python?

Strings in Python can be created using single quotes or double quotes or even triple quotes.
For example:

```
# Creating a String
# with single Quotes
String1 = 'Welcome to the Geeks World'

# Creating a String
# with double Quotes
String1 = "I'm a Geek"

# Creating a String
# with triple Quotes
String1 = '''I'm a Geek and I live in a world of "Geeks"'''

# Creating String with triple
# Quotes allows multiple lines
String1 = '''Geeks
            For
            Life'''
```

Accessing characters in Python In Python, individual characters of a String can be accessed by using the method of Indexing. Indexing allows negative address references to access characters from the back of the String, e.g. -1 refers to the last character, -2 refers to the second last character... While accessing an index out of the range will cause an IndexError. Only Integers are allowed to be passed as an index, float or other types will cause a TypeError.

What is string slicing?

To access a range of characters in the String, a method of slicing is used. Slicing in a String is done by using a Slicing operator (colon).
For example:

```
String1 = "TEST STRING"
print("Initial String: ")
print(String1)

# Printing 3rd to 12th character
print(String1[3:12])

# Printing characters between
# 3rd and 2nd last character
print(String1[3:-2])
```

How to delete/update a string?

In Python, Updation or deletion of characters from a String is not allowed. This will cause an error because item assignment or item deletion from a String is not supported. Although deletion of entire String is possible with the use of a built-in del keyword. This is because Strings are immutable, hence elements of a String cannot be changed once it has been assigned. Only new strings can be reassigned to the same name.

Code

```
userInput = input("Please enter a string: ")
ans = ""
cnt = 0
for ch in userInput:
    if(ch == 'a'):
        cnt += 1
        if(cnt > 1):
            ans += '@'
        else:
            ans += ch
    else:
        ans += ch
print(ans)
```

Output

```
(venv) D:\Dheeraj\Study\SEM 4\PythonPracticals\Experiment2\code>python question3.py
Please enter a string: acapella is an art of teamwork
ac@pell@ is @n @rt of te@mwork
```

Dheeraj Lalwani
1902085

Question 4:

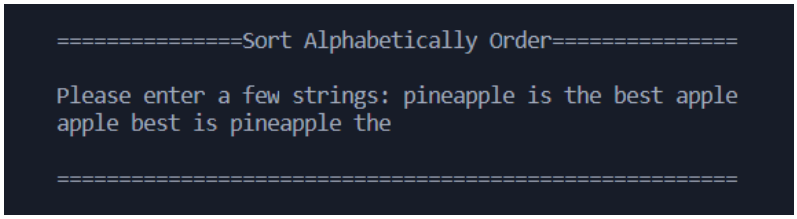
- to sort group of strings into alphabetical order

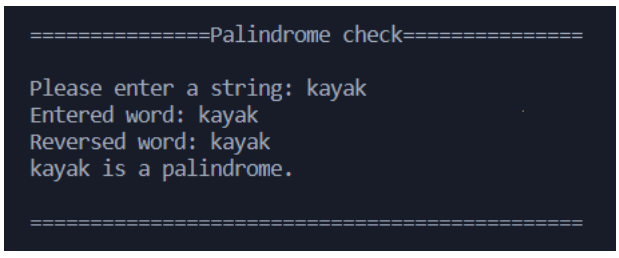
- to check whether the entered string is palindrome or not.

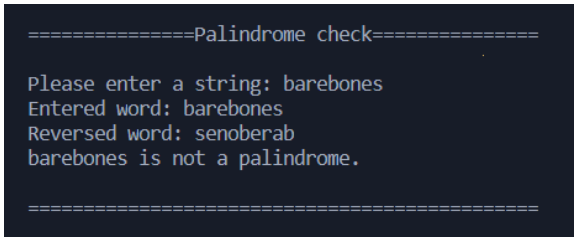
Code

```
print("\n===============Sort Alphabetically Order===============\n")

userInput = input("Please enter a few strings: ")
userInput = userInput.split()
userInput.sort()
for word in userInput:
    print(word, end=" "),

print("\n\n=========================================================\n")


print("\n===============Palindrome check===============\n")

userInput = input("Please enter a string: ")
print(f"Entered word: {userInput}")
print(f"Reversed word: {userInput[::-1]}")
if userInput == userInput[::-1]:
    print(f"{userInput} is a palindrome.")
else:
    print(f"{userInput} is not a palindrome.")

print("\n=================================================\n")
```

Output







Dheeraj Lalwani
1902085