**Code:**

**#Code1**

**def isEmpty(stk):**

**# checks whether the stack is empty or not**

**if stk==[]:**

**return True**

**else:**

**return False**

**def Push(stk,item): # Allow additions to the stack**

**stk.append(item)**

**top = len(stk)-1**

**def Pop(stk):**

**if isEmpty(stk): # verifies whether the stack is empty or not**

**print("Underflow")**

**else:**

**# Allow deletions from the stack**

**item=stk.pop()**

**if len(stk)==0:**

**top = None**

**else:**

**top=len(stk)**

**print("Popped item is "+str(item))**

**def Display(stk):**

**if isEmpty(stk):**

**print("Stack is empty")**

**else:**

**top=len(stk)-1**

**print("Elements in the stack are: ")**

**for i in range(top,-1,-1):**

**print (str(stk[i]))**

**# executable code**

**stk=[]**

**top=None**

**Push(stk,1)**

**Push(stk,2)**

**Push(stk,3)**

**Push(stk,4)**

**print("Stack before popping an element:")**

**Display(stk)**

**Pop(stk)**

**print("\nStack after popping an element:")**

**Display(stk)**

**#Code2**

**#Adding elements to queue at the rear end**

**def enqueue(data):**

**queue.insert(0,data)**

**#Removing the front element from the queue**

**def dequeue():**

**if len(queue)>0:**

**return queue.pop()**

**return ("Queue Empty!")**

**#To display the elements of the queue**

**def display():**

**print("Elements on queue are:");**

**for i in range(len(queue)):**

**print(queue[i])**

**# executable code**

**queue=[]**

**enqueue(5)**

**enqueue(6)**

**enqueue(9)**

**enqueue(5)**

**enqueue(3)**

**print("Queue before popping element: ")**

**display()**

**print("\nPopped Element is: "+str(dequeue()))**

**print("Queue after popping element: ")**

**display()**

**#Code3**

**# importing module**

**import collections**

**# initialising a deque() of arbitrary length**

**linked\_lst = collections.deque()**

**# filling deque() with elements**

**linked\_lst.append('first')**

**linked\_lst.append('second')**

**linked\_lst.append('third')**

**print("Elements in the linked\_list:")**

**print(linked\_lst)**

**# adding element at an arbitrary position**

**linked\_lst.insert(1, 'fourth')**

**print("\nElements in the linked\_list:")**

**print(linked\_lst)**

**# deleting the last element**

**linked\_lst.pop()**

**print("\nElements in the linked\_list:")**

**print(linked\_lst)**

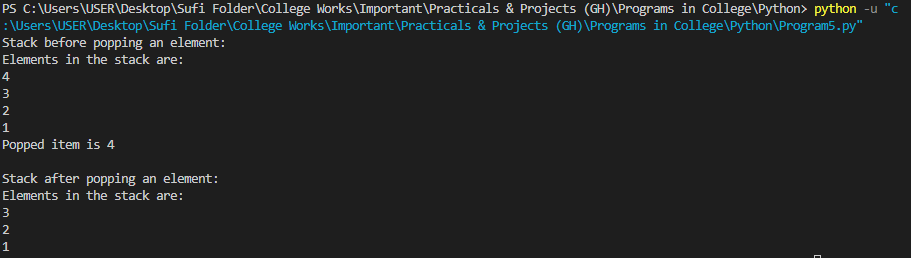
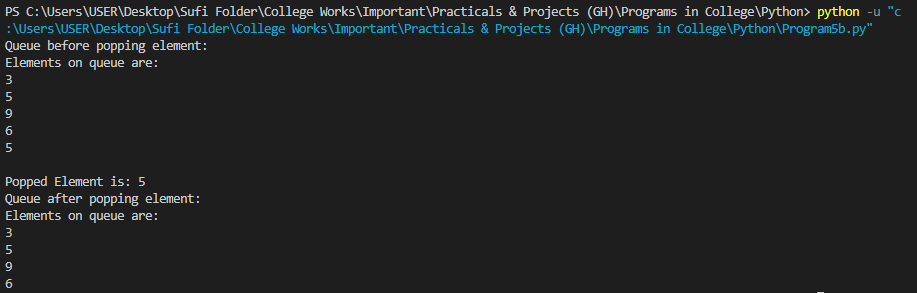
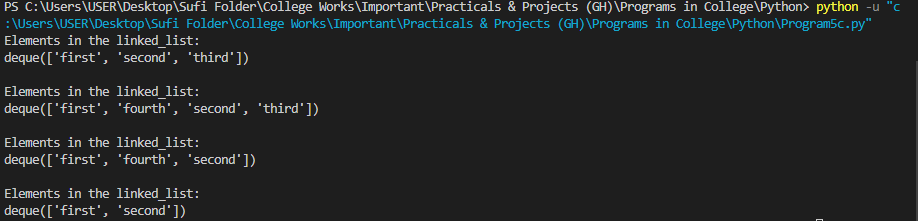
**# removing a specific element**

**linked\_lst.remove('fourth')**

**print("\nElements in the linked\_list:")**

**print(linked\_lst)**

**Outputs:**

****