

WORKSHEET

DATA STRUCTURE

1	<p>Stack is a data structure that follows _____ order</p> <p>a. FIFO b. LIFO c. LILO d. FILO</p>
Ans	b. LIFO
2	<p>Queue is a data structure that follows _____ order</p> <p>a. FIFO b. LIFO c. LILO d. FILO</p>
Ans	a. FIFO
3	<p>What will be the output of the following Python code :</p> <pre>mylist = ['java', 'c++', 'python', 'vb', 'vc++', 'php', 'cobol'] print(mylist.pop()) mylist.append('c#') print(len(mylist)) mylist.insert(3, 'basic') print(mylist) mylist.sort() print(mylist) print(mylist.index('php'))</pre>
Ans	<p>cobol</p> <p>7</p> <p>['java', 'c++', 'python', 'basic', 'vb', 'vc++', 'php', 'c#']</p> <p>['basic', 'c#', 'c++', 'java', 'php', 'python', 'vb', 'vc++']</p> <p>4</p>
4	<p>Identify the error in following code:</p> <p>(i) List1 = [10,20,30,40] List1[4]=100</p> <p>(ii) Name="Michael Jackson" Name[2]="k"</p>
Ans	<p>(i) Index 4 is out of range, because List1 contains 4 items (Index 0 to 3 only)</p> <p>(ii) String is immutable type so we can't change any character of string in place.</p>
5	<p>Write down the status of Stack after each operation:</p> <p>Stack =[10,20,30,40] where TOP item is 40</p> <p>(i) Push 70 (ii) Push 100 (iii) Pop an item from Stack (iv) Peek the Stack</p>
Ans	<p>(i) [10,20,30,40,70]</p> <p>(ii) [10,20,30,40,70,100]</p> <p>(iii) [10,20,30,40,70]</p> <p>(iv) 70</p>

6	<p>Write down the status of Queue after each operation: Queue=[10,20,30,40] where FRONT item is 10</p> <p>(i) Insert 70 (ii) Remove item from Queue (iii) Item at the front</p>
Ans	<p>(i) [10,20,30,40,70] (ii) [20,30,40,70] (iii) 20</p>
7	<p>Identify the Data Structure (Stack/Queue) used for following operations in computer?</p> <p>(i) Function calling (ii) Printer spooling (iii) Undo operation (iv) Solving expression (v) Keyboard buffering</p>
Ans	<p>(i) Stack (ii) Queue (iii) Stack (iv) Stack (v) Queue</p>
8	<p>Write down the steps to perform Binary Search for the following items in the given list MYLIST.</p> <p>(i) 17 (ii) 30</p> <p>MYLIST = [5,11,17,19,25,29,30,32,46,90]</p>
Ans	<p>(i) First we will find mid, here low = 0 and high=9 mid = (low+high)//2 => (0+9)//2 = 4 Now we will compare 17 with middle position item MYLIST[4] which is 25 25 > 17 therefore 17 will be towards the left of 25 so we will update high=mid-1</p> <p>Now we will find mid again mid = (0+3)//2 => 1 Now we will compare 17 with middle position item MYLIST[1] which is 11 11 < 17 therefore 17 will be towards the right of 11 so we will update low=mid+1</p> <p>Now we will find mid again mid = (2+3)//2 => 2 Now we will compare 17 with middle position item MYLIST[2] which is 17 17 == 17, therefore item found, search successful, position found is index 2</p> <p>(ii) First we will find mid, here low = 0 and high=9 mid = (low+high)//2 => (0+9)//2 = 4 Now we will compare 30 with middle position item MYLIST[4] which is 25 25 < 30 therefore 30 will be towards the right of 25 so we will update low=mid+1</p>

Now we will find mid again

mid = (5+9)//2 => 7

Now we will compare 30 with middle position item MYLIST[7] which is 32

32>30, therefore 30 will be towards the left of 32, so we will update high=mid-1

Now we will find mid again

mid = (5+6)//2 => 5

Now we will compare 30 with middle position item MYLIST[5] which is 29

29<30, therefore 30 will be towards the right of 29, so we will update low=mid+1

Now we will find mid again

mid=(6+6)//2=>6

Now we will compare 30 with middle position item MYLIST[6] which is 30

30==30, therefore search successful, item found at position index 6

9

Given the list:

MYLIST = [5,11,17,19,25,29,30,30,32,46,90]

Write down the Python statements for the following requirement:

- (i) To find the number of items in MYLIST
- (ii) To find the frequency of item 30 in MYLIST i.e. how many times 30 is in MYLIST
- (iii) Write the code to insert 45 in the above sorted list to its correct position (do not disturb the sorting)
- (iv) Write the code to delete 17 from the above sorted list

Ans

- (i) `print(len(MYLIST))`
- (ii) `print(MYLIST.count(30))`
- (iii) `import bisect`
`bisect.insort(MYLIST,45)`
`print(MYLIST)`
- (iv) `MYLIST.remove(17)`
`print(MYLIST)`

10

Write the function sumAlternate(MYLIST) as argument and calculate the sum of all alternate elements of MYLIST and print it

For e.g. if the elements are

5	11	17	19	25	29	30	32	56	90
---	----	----	----	----	----	----	----	----	----

Output should be :

Total = 133

Ans

```
def sumAlternate(MYLIST):
    sum=0
    for i in range(0,len(MYLIST),2):
        sum+=MYLIST[i]
    print("Total = ",sum)
```

11	<p>Write the function SumEvenOdd(MYLIST) to find the sum of all Even elements and sum of all Odd elements present in MYLIST</p> <p>For e.g if the elements are</p> <table border="1" data-bbox="185 197 1458 235"> <tr> <td>8</td><td>12</td><td>17</td><td>19</td><td>25</td><td>29</td><td>33</td><td>32</td><td>56</td><td>90</td> </tr> </table> <p>Output should be: Even Sum = 198 Odd Sum = 123</p>	8	12	17	19	25	29	33	32	56	90
8	12	17	19	25	29	33	32	56	90		
Ans	<pre>def SumEvenOdd(MYLIST): sume=0 sumo=0 for i in range(len(MYLIST)): if MYLIST[i]%2==0: sume+=MYLIST[i] else: sumo+=MYLIST[i] print("Even Sum = ",sume) print("Odd Sum = ",sumo)</pre>										
12	<p>Write the function CountEvenOdd(MYLIST) to find the count of all Even elements and sum of all Odd elements.</p> <p>For e.g if the elements are</p> <table border="1" data-bbox="185 835 1458 873"> <tr> <td>8</td><td>12</td><td>17</td><td>19</td><td>25</td><td>28</td><td>33</td><td>32</td><td>56</td><td>90</td> </tr> </table> <p>Output should be: Even Count = 6 Odd Sum = 4</p>	8	12	17	19	25	28	33	32	56	90
8	12	17	19	25	28	33	32	56	90		
Ans	<pre>def CountEvenOdd(MYLIST): counte=0 counto=0 for i in range(len(MYLIST)): if MYLIST[i]%2==0: counte+=1 else: counto+=1 print("Even Count = ",counte) print("Odd Count = ",counto)</pre>										
13	<p>Write code using list comprehension for the following:</p> <p>(i) To create a list(mylist) with all even numbers from 2 to 100</p> <p>(ii) To create a list(mylist) with all alternate values of another list(templist)</p>										
Ans	<pre>(i) mylist = [i for i in range(2,101) if i%2==0] (ii) templist=[10,20,30,40,50,60,70,80,90,100] mylist = [templist[i] for i in range(0,len(templist),2)]</pre>										
14	<p>Write a function PrintDiagonal(MATRIX), where MATRIX is a list storing 3 list inside it with each list contains 3 items and print only diagonal elements and also sum of it.</p> <p>For e.g. is the MATRIX element is [[10,20,30],[40,50,60],[70,80,90]]</p> <p>Output should be 10 50 90 Sum=150</p>										

Ans	<pre>def PrintDiagonal(MATRIX): sum=0 for i in range(len(MATRIX)): for j in range(len(MATRIX[i])): if i==j: print(MATRIX[i][j]) sum+=MATRIX[i][j] print('Sum=',sum)</pre>
15	<p>Write a function EvenOdd(MYLIST), which doubles each Odd elements of MYLIST and half each Even element of MYLIST. For e.g. if MYLIST = [10,11,40,4,17,23,45,100,80] The output should be [5,22,20,2,34,46,90,50,40]</p>
Ans	<pre>def EvenOdd(MYLIST): for i in range(len(MYLIST)): if MYLIST[i]%2==0: MYLIST[i]//=2 else: MYLIST[i]*=2</pre>
16	<p>Write a function Sum7End(MYLIST), which display only those items from the list which ends from the digit 7, also find total of these elements. For e.g. if MYLIST = [10,27,15,107,97,5,7,81,47] The output should be 27 107 97 7 47 Total = 285</p>
Ans	<pre>def Sum7End(MYLIST): sum=0 for i in range(len(MYLIST)): if MYLIST[i]%10==7: print(MYLIST[i]) sum+=MYLIST[i] print('Total=',sum)</pre>
17	<p>Write QueueUp(Student) and QueueDel(Student) methods/function Python to add a new Student and delete a Student from a List of Student names, considering them to act as insert and delete operations of the Queue data structure</p>
Ans	<pre>def QueueUp(Student): name = input('Enter any name') Student.append(name) def QueueDel(Student): if len(Student)==0: print('Underflow') else: name = Student.pop(0) print('Deleted Name was ',name)</pre>

18	Write a function in python, Push(Employee) and Pop(Employee) to add a new Employee and delete a Employee from a List of Employee Names, considering them to act as push and pop operations of the Stack data structure.
Ans	<pre> def Push(Employee): name=input('Enter Employee name ') Employee.append(name) def Pop(Employee): if len(Employee)==0: # or if Employee==[]: print('Underflow') else: name = Employee.pop() print('Popped Name was ',name) </pre>
19	Write QueueUp(Student) and QueueDisp(Student) methods/function Python to add a new Student and display Student names from a List of Students, considering them to act as insert and display operations of the Queue data structure
Ans	<pre> def QueueUp(Student): name = input('Enter Name') Student.append(name) def QueueDisp(Student): if len(Student)==0: print('Underflow') else: print('Queue Items Front-to-Rear') for i in range(len(Student)): print(Student[i]) </pre>
20	Write a function in python, Push(Employee) and Show(Employee) to add a new Employee and display Employee names from a List of Employee, considering them to act as push and show operations of the Stack data structure.
Ans	<pre> def Push(Employee): name=input('Enter name :') Employee.append(name) def Show(Employee): if len(Employee)==0: print('Underflow') else: print('Employee Names ') for i in range(len(Employee)): print(Employee[i]) </pre>
21	<p>Write a function Count(Salary,Val) to count and display number of times Val is present in the list Salary.</p> <p>For example If the Salary contains: 50000,40000,50000,60000,70000,50000,1000 and Val contains 50000 The function should display 50000 found 3 Times</p>

Ans	<pre>def Count(Salary,Val): count=0 for S in Salary: if S==Val: count+=1 print(Val,'found',count,'Times')</pre>
22	<p>Consider the randomly ordered numbers stored in a list 55,53,57,51,52,54,56</p> <p>Show the content of list after the First, Second and third pass of the Bubble sort method used for arranging in descending order</p> <p>Note: show the status of all elements after each pass very clearly encircling the changes.</p>
Ans	<p>After First Pass [53, 55, 51, 52, 54, 56, 57]</p> <p>After Second Pass [53, 51, 52, 54, 55, 56, 57]</p> <p>After Third Pass [51, 52, 53, 54, 55, 56, 57]</p>
23	<p>Consider the randomly ordered numbers stored in a list 55,53,57,51,52,54,56</p> <p>Show the content of list after the First, Second and third pass of the Insertion sort method used for arranging in ascending order</p> <p>Note: show the status of all elements after each pass very clearly encircling the changes.</p>
Ans	<p>After Pass 1 [53, 55, 57, 51, 52, 54, 56]</p> <p>After Pass 2 [53, 55, 57, 51, 52, 54, 56]</p> <p>After Pass 3 [51, 53, 55, 57, 52, 54, 56]</p>
24	<p>Consider the randomly ordered numbers stored in a list 55,53,57,51,52,54,56</p> <p>Show the content of list after the First, Second and third pass of the Bubble sort method used for arranging in descending order</p> <p>Note: show the status of all elements after each pass very clearly encircling the changes.</p>
Ans	<p>After Pass 1 [55, 57, 53, 52, 54, 56, 51]</p> <p>After Pass 2 [57, 55, 53, 54, 56, 52, 51]</p> <p>After Pass 3 [57, 55, 54, 56, 53, 52, 51]</p>

25	<p>Raj is a Python programmer working on sorting module. For a small list of values he has written the Bubble sorting code but code is not executing. Help Raj and rewrite the code after removing all the errors and underlining the correction(s) made:</p> <pre>def BubbleSort(num): for i in range(num-1): for j in range(num-1-i): if num[j+1]>num[j]: num[i],num[j+1]=num[j+1],num[i]</pre>
Ans	<pre>def BubbleSort(num): for i in range(len(num)-1): for j in range(len(num)-1-i): if num[j]>num[j+1]: <u>num[j],num[j+1]=num[j+1],num[j]</u></pre>
26	<p>Raj is a Python programmer working on sorting module. For this he has written the Insertion sorting code but code is not executing. Help Raj and rewrite the code after removing all the errors and underlining the correction(s) made:</p> <pre>def InsertionSort(mylist): i=1 while i<len(mylist): key=mylist[i] j=i-1 while j>=0 and mylist[j]<key: mylist[j]=mylist[j+1] j-=1 key = mylist[j+1] i-=1</pre>
Ans	<pre>def InsertionSort(mylist): i=1 while i<len(mylist): key=mylist[i] j=i-1 while j>=0 and mylist[j]>key: <u>mylist[j+1]=mylist[j]</u> j-=1 <u>mylist[j+1]=key</u> i+=1</pre>
27	<p>Write a Python function/method Count5and7(N), to find and display count of all number between 1 to N which are either divisible by 5 or by 7.</p> <p>For e.g. if the N is 20 then output should be: Count=6 As (5,7,10,14,15,20) are the number between 1 to N which are divisible by either 5 or by 7)</p>

Ans	<pre> def Count5and7(N): count=0 for i in range(1,N+1): if i%5==0 or i%7==0: count+=1 print('Count=',count) </pre>
28	<p>Write a Python function/method SwapMiddle(Codes) to swap the first half of the content of the list Codes with second half of the list Codes and display the swapped values.</p> <p>For e.g. if the list Codes contains : [22,44,55,66,88,11] then function should swap and display as: [66,88,11,22,44,55]</p>
Ans	<pre> def SwapMiddle(Codes): i=0 mid = len(Codes)//2 while i<mid: Codes[i],Codes[mid+i]=Codes[mid+i],Codes[i] i+=1 </pre>
29	<p>Raj is a Python programmer working of Data structure Stack to store name of visitors, he has implemented the code for PUSH and POP, but both functions are not producing the correct result. Help Raj in identifying the error(s) and also write the Correct code for specific line number:</p> <pre> def PUSH(VISITOR,name): name.append(VISITOR) #Line 1 top = len(VISITOR)-1 #Line 2 def POP(VISITOR): if len(VISITOR)==1: #Line 3 return "Sorry! No Visitor to delete " #Line 4 else: val = VISITOR.pop(1) #Line 5 if len(VISITOR)==1: #Line 6 top=None #Line 7 else top=len(VISITOR) #Line 8 return val </pre>
Ans	<pre> #Line 1 VISITOR.append(name) </pre>

	<pre> #Line 3 if len(VISITOR)==0: #Line 5 val = VISITOR.pop() #Line 6 if len(VISITOR)==0: #Line 8 top = len(VISITOR)-1 </pre>
30	<p>Raj is a Python programmer working of Data structure Queue to store name of REQUESTNO, he has implemented the code for ENQUEUE and DEQUEUE, for insert and delete operation in Queue resp. But both functions are not producing the correct result. Help Raj in identifying the Line Number where code is incorrect and also write the Correct code for same Line Number.</p> <pre> def Enqueue(REQUESTNO,item): REQUESTNO.add(item) #Line 1 if len(REQUESTNO)==1: #Line 2 front=rear=1 #Line 3 else: rear=len(REQUESTNO)-1 #Line4 def Dequeue(REQUESTNO): if REQUESTNO==0: #Line 5 print("Underflow") #Line 6 else: val = REQUESTNO.pop(len(REQUESTNO)-1) #Line 7 if len(REQUESTNO)==0: #Line 8 front=None #Line 9 return val </pre>
Ans	<pre> #Line 1 REQUESTNO.append(item) #Line 3 front = rear = 0 #Line 5 if len(REQUESTNO)==0: #Line 7 val = REQUESTNO.pop(0) #Line 9 front = rear = None </pre>