

# book1-dsa

May 7, 2022

## 1 book1-dsa

Use the “Run” button to execute the code.

```
[1]: !pip install jovian --upgrade --quiet
```

```
[2]: import jovian
```

```
[3]: # Execute this to save new versions of the notebook
jovian.commit(project="book1-dsa")
```

<IPython.core.display.Javascript object>

[jovian] Updating notebook "kishkath/book1-dsa" on <https://jovian.ai>

[jovian] Committed successfully! <https://jovian.ai/kishkath/book1-dsa>

```
[3]: 'https://jovian.ai/kishkath/book1-dsa'
```

**1.0.1 Q1.** Given an array of N integers. Your task is to print the sum of all of the integers.

```
[7]: def func_sum(array):

    res = 0
    for i in array:
        res+=i
    return res
```

```
[15]: N1 = 4
input1 = [1,2,3,4]
print(func_sum(input1))
```

10

```
[16]: N2 = 6
input2 = [5,8,3,10,22,45]
print(func_sum(input2))
```

93

**1.0.2 Q2.** Given an array A[] of N integers and an index Key. Your task is to print the element present at index key in the array.

```
[17]: def func_present(array,N,index):  
       return array[index]
```

```
[14]: n1,index = 5,2  
inp1 = [10,20,30,40,50]  
print(func_present(inp1,n1,index))
```

30

```
[18]: n2,index1 = 7,4  
inp2 = [10,20,30,40,50,60,70]  
print(func_present(inp2,n2,index1))
```

50

**1.0.3 Q3.** Given an sorted array A of size N. Find number of elements which are less than or equal to given element X.

```
[19]: def lessthan_equal(A,element):  
  
       count = 0  
       for i in A:  
           if (i<=element):  
               count+=1  
       return f"{count} elements are less than {element} in the array"
```

```
[20]: size = 6  
A = [1,2,4,5,8,10]  
X = 9  
print(lessthan_equal(A,X))
```

5 elements are less than 9 in the array

```
[21]: size1 = 7  
A = [1,2,2,2,5,7,9]  
x = 2  
print(lessthan_equal(A,x))
```

4 elements are less than 2 in the array

1.0.4 Q4. You are given an array A of size N. You need to print elements of A in alternate order (starting from index 0).

```
[25]: def print_alternative(Array,N):  
  
    print("The Alternative elements of array are: ")  
    for i in range(0,N,2):  
        print(Array[i],end=' ')
```

```
[26]: N = 4  
Array =[1,2,3,4]  
print_alternative(Array,N)
```

The Alternative elements of array are:  
1 3

```
[27]: N2 = 5  
Array1 = [1,2,3,4,5]  
print_alternative(Array1,N2)
```

The Alternative elements of array are:  
1 3 5

1.0.5 Q5. Given an array Arr of N positive integers. Your task is to find the elements whose value is equal to that of its index value ( Consider 1-based indexing ).

```
[45]: def element_equals_index(Arr,length):  
    for i in range(length):  
        if Arr[i]==i+1:  
            print(Arr[i],end=' ')
```

```
[46]: length = 5  
Arr = [15,2,45,12,7]  
element_equals_index(Arr,length)
```

2

```
[47]: length1 = 1  
Arrr = [1]  
element_equals_index(Arrr,length1)
```

1

```
[ ]:
```

1.0.6 Q6. Given an array of size N and you have to tell whether the array is perfect or not. An array is said

1.0.7 to be perfect if it's reverse array matches the original array. If the array is perfect then print

1.0.8 "PERFECT" else print "NOT PERFECT"

```
[55]: def is_perfect(arr):  
  
    count = 0  
    for i in arr[:len(arr)//2]:  
        if (arr[i]==arr[-(i+1)]):  
            count += 1  
  
    #print(count)  
    if (count==(len(arr)//2)):  
  
        return True  
    else:  
        return False
```

```
[56]: arr = [1,2,3,4,5,4,3,2,1]  
  
if (is_perfect(arr)==True):  
    print("PERFECT")  
else:  
    print("NOT PERFECT")
```

PERFECT

```
[57]: arr1 = [0,1,2,3,4,5,6,3,2,1,0]  
if (is_perfect(arr1)==True):  
    print("PERFECT")  
else:  
    print("NOT PERFECT")
```

NOT PERFECT

```
[58]: arr2 = [1,2,3,4,5]  
if (is_perfect(arr2)==True):  
    print("PERFECT")  
else:  
    print("NOT PERFECT")
```

NOT PERFECT

1.0.9 Q7. Given an array of length N, at each step it is reduced by 1 element. In the first step the maximum

1.0.10 element would be removed, while in the second step minimum element of the remaining array would

be removed, in the third step again the maximum and so on. Continue this till the array contains only 1

element. And find the final element remaining in the array.

```
[70]: def remove_minmax(arrn,N):  
  
    for i in range(N):  
        if len(arrn)==1:  
            break  
        elif len(arrn)==2:  
            arrn.remove(max(arrn))  
  
        else:  
            arrn.remove(max(arrn))  
            arrn.remove(min(arrn))  
    return arrn
```

```
[71]: N = 7  
arrn = [7,8,3,4,2,9,5]  
print(remove_minmax(arrn,N))
```

[5]

```
[72]: N1 = 8  
arrn1 = [8,1,2,9,4,3,7,5]  
print(remove_minmax(arrn1,N1))
```

[4]

1.0.11 Q8. Given an array of N distinct elements, the task is to find all elements in array except two greatest elements in sorted order.

```
[75]: def elements(array):  
    array.sort()  
    return array[:-2]
```

```
[76]: array = [2,8,7,1,5]  
print(elements(array))
```

[1, 2, 5]

```
[77]: array1 = [7,-2,3,4,9,-1]
      print(elements(array1))
```

[-2, -1, 3, 4]

**1.0.12 Q9.** Write a program to find the sum of the given series  $1+2+3+ \dots (N \text{ terms})$

```
[82]: def sum_of_series(nums):
      res = 0
      for i in range(nums+1):
          res+=i
      return f"Sum of series of {nums} terms is {res}"
```

```
[83]: print(sum_of_series(3))
```

Sum of series of 3 terms is 6

```
[84]: print(sum_of_series(5))
```

Sum of series of 5 terms is 15

**1.0.13 Q10.** Given a number N. Your task is to check whether it is fascinating or not

```
[ ]:
```

```
[99]: def check_fascinating(n):
      product1 = n*2
      product2 = n*3

      res = ""
      if len(str(n))<3:
          return False
      else:
          res = (str(n)+str(n*2)+str(n*3))

      for i in range(1,10):
          if res.count(str(i))!=1:
              return False
      return True
```

```
[101]: Num = 192
```

```
if (check_fascinating(Num)):
    print(f"{Num} is Fascinating")
```

```
else:
    print(f"{Num} is not Fascinating")
```

192 is Fascinating

```
[102]: Num1 = 853

if (check_fascinating(Num1)):
    print(f"{Num1} is Fascinating")
else:
    print(f"{Num1} is not Fascinating")
```

853 is not Fascinating

[ ]:

1.0.14 Q11. Given an array of even size N, task is to find minimum value that can be added to an element so that

1.0.15 array become balanced. An array is balanced if the sum of the left half of the array elements is equal

1.0.16 to the sum of right half.

```
[141]: def if_balance(arr,SIZE):
        half_size = SIZE//2

        #print(sum(arr[:half_size]))
        left_half = arr[:half_size]
        right_half = arr[half_size:]
        #print(left_half,right_half)
        return abs(sum(left_half)-sum(right_half))
```

```
[142]: EVEN_SIZE = 4
        arraay = [1,5,3,2]
        print(f"{if_balance(arraay,EVEN_SIZE)} has to be added to array to get_
        ↳balanced")
```

1 has to be added to array to get balanced

```
[143]: EVEN_SIZE1 = 6
        arraay1 = [1,2,1,2,1,3]

        print(f"{if_balance(arraay1,EVEN_SIZE1)} has to be added to array to get_
        ↳balanced")
```

2 has to be added to array to get balanced

[ ]:

## 1.1 “Done!”

[ ]: `jovian.commit(project="book1_dsa")`

<IPython.core.display.Javascript object>

[ ]: