

Instructions for students:

1. Complete the following methods on Arrays,
2. You may use any language to complete the tasks.
3. All your methods must be written in one single .java or .py file. DO NOT CREATE separate files for each task.
4. You need to submit one single file containing all the methods/functions. You will get two weeks to complete your lab.
5. The submission format MUST be maintained. You need to copy paste all your codes in ONE SINGLE .txt file and upload that. If format is not maintained, whole lab submission will be canceled.
6. If you are using JAVA, you must include the main method as well which should test your other methods and print the outputs according to the tasks.
7. If you are using PYTHON, make sure your code has the methods invoked and proper printing statements according to the tasks.
8. The google form link for this lab will be provided in BUX under LAB 1-Arrays subsection under the FALL21 CSE220 lab tab 2 days before the deadline.
9. Deadline for this assignment is on the 28th of October. The submission form will be open for 2 days after the deadline to accept late submissions. 30% marks will be deducted for each day past the deadline.

Linear Arrays

NOTE: Each method carries 5 marks

1. Shift Left k Cells

Consider an array named source. Write a method/function named shiftLeft(source, k) that shifts all the elements of the source array to the left by 'k' positions. You must execute the method by passing an array and number of cells to be shifted. After calling the method, print the array to show whether the elements have been shifted properly.

Example:

```
source=[10,20,30,40,50,60]
```

```
shiftLeft(source,3)
```

After calling shiftLeft(source,3), printing the array should give the output as:

```
[ 40, 50, 60, 0, 0, 0 ]
```

2. Rotate Left k cells

Consider an array named source. Write a method/function named rotateLeft(source, k) that rotates all the elements of the source array to the left by 'k' positions. You must execute the method by passing an array and number of cells to be shifted. After calling the method, print the array to show whether the elements have been shifted properly.

Example:

```
source=[10,20,30,40,50,60]
```

```
rotateLeft(source,3)
```

After calling rotateLeft(source,3), printing the array should give the output as:

```
[ 40, 50, 60, 10, 20, 30]
```

3. Remove an element from an array

Consider an array named source. Write a method/function named remove(source, size, idx) that removes the element in index **idx** of the source array. You must execute the method by passing an array, its size and the idx(that is the index of the element to be removed). After calling the method, print the array to show whether the element of that particular index has been removed properly.

Example:

```
source=[10,20,30,40,50,0,0]
```

```
remove(source,5,2)
```

After calling remove(source,5,2) , printing the array should give the output as:

```
[ 10,20,40,50,0,0,0]
```

4. Remove all occurrences of a particular element from an array

Consider an array named source. Write a method/function named removeAll(source, size, element) that removes all the occurrences of the given element in the source array. You must execute the method by passing an array, its size and the element to be removed. After calling the method, print the array to show whether all the occurrences of the element have been removed properly.

Example:

```
source=[10,2,30,2,50,2,2,0,0]
```

```
removeAll(source,7,2)
```

After calling removeAll(source,7,2), all the occurrences of 2 must be removed. Printing the array afterwards should give the output as:

```
[ 10,30,50,0,0,0,0,0]
```

5. Splitting an Array

Suppose the elements of an array A containing positive integers, denote the weights in kilograms. And we have a beam balance. We want to put the weights on both pans of the balance in such a way that for some index $0 < i < A.length - 1$, all values starting from A[0], A[1], upto A[i - 1], should be on the left pan. And all values starting from A[i] upto A[A.length - 1] should be on the right pan and the left and right pan should be balanced. If such an i exists, return true. Else, return false.

Input: [1, 1, 1, 2, 1] Output : true

Explanation: (summation of [1, 1, 1] = summation of [2,1])

Input: [2, 1, 1, 2, 1] Output: false

Input: [10, 3, 1, 2, 10] Output: true

Explanation: (summation of [10, 3] = summation of [1,2,10]))

6. Array series

Write a method that takes an integer value n as a parameter. Inside the method, you should create an array of length n squared ($n*n$) and fill the array with the following pattern. Return the array at the end and print it.

If,

n=2: { 0,1, 2,1 } (spaces have been added to show two distinct groups).

n=3 : { 0, 0, 1, 0, 2, 1, 3, 2, 1 } ((spaces have been added to show three distinct groups).

n=4 : {0, 0, 0, 1, 0, 0, 2, 1, 0, 3, 2, 1, 4, 3, 2, 1} (spaces have been added to show four distinct groups).

7. Max Bunch Count

A "bunch" in an array is a consecutive chain of two or more adjacent elements of the same value. Write a method that returns the number of elements in the largest bunch found in the given array.

Input: [1, 2, 2, 3, 4, 4, 4] Output: 3

Explanation: There are two bunches here {2,2} and {4,4,4}. The largest bunch is {4,4,4} containing 3 elements so 3 is returned.

Input: [1,1,2, 2, 1, 1,1,1] Output:4

Explanation: There are three bunches here {1,1} and {2,2} and {1,1,1,1}. The largest bunch is {1,1,1,1} containing 4 elements so 4 is returned.

8. Repetition

Write a method that takes in an array as a parameter and counts the repetition of each element. That is, if an element has appeared in the array more than once, then its 'repetition' is its number of occurrences. The method returns true if there are at least two elements with the same number of 'repetition'. Otherwise, return false.

Input: {4,5,6,6,4,3,6,4} Output: True

Explanation: Two numbers repeat in this array: 4 and 6. 4 has a repetition of 3, 6 has a repetition of 3. Since two numbers have the same repetition output is True.

Input: {3,4,6,3,4,7,4,6,8,6,6} Output: False

Explanation: Three numbers repeat in this array: 3, 4 and 6. 3 has a repetition of 2, 4 has a repetition of 3, 6 has a repetition of 4. Since no two numbers have the same repetition output is False.

Circular Array

NB: You need to implement circular arrays to solve the following problems. You cannot convert them to linear arrays.

1. Palindrome

Write a method/function that takes in a circular array, its size and start index and finds whether the elements in the array form a palindrome or not. Return true if the elements form a palindrome, otherwise, return false.

Example:

Input: [20,10,0,0,0,10,20,30] (start =5, size=5) Output: True.

Input:[10,20,0,0,0,10,20,30] (start =5, size=5) Output: False

2. Intersection

Write a method/function that takes two circular arrays, their sizes and start indexes and returns a linear array containing the common elements between the two circular arrays.

Input:

Circular array 1 : [40,50,0,0,0,10,20,30] (start_1 =5, size_1 =5)

[10 20 30 40 50]

Circular array 2 : [10,20,5,0,0,0,0,5,40,15,25] (start_2=8, size_2 =7)

[5 40 15 25 10 20 5]

Output: [10,20,40]

3. Suppose you have been hired to develop a musical chair game. In this game there will be 7 participants and all of them will be moving clockwise around a set of 7 chairs organized in a circular manner while music will be played in the background. You will control the music using random numbers between 0-3. If the generated random number is 1, you will stop the music and if the number of participants who are still in the game is n , the participants at position $(n/2)$ will be eliminated. Each time a participant is eliminated, a chair will be removed and you have to print the player names who are still in the game. The game will end when there will be only one participant left. At the end of the game, display the name of the winner.

[Hint: You will need to invoke a method to generate a random number between 0 (inclusive) to 3 (inclusive)]

Assignment 01

CSE220 (FALL - 21)

In [2]: *# Linear Array*

#Task 1

```
def shiftLeft(source,k):
    for j in range(k):
        for i in range(len(source)-1):
            source[i]=source[i+1]
            source[i+1]=0
ai= [10, 20, 30, 40, 50,60]
shiftLeft(ai,3)
print(ai)
```

[40, 50, 60, 0, 0, 0]

In [3]: *#Task 2*

```
def rotateLeft(source,k):
    for i in range(k):
        temp=source[0]
        for i in range(len(source)-1):
            source[i]=source[i+1]
            source[len(source)-1]=temp
b= [10, 20, 30, 40, 50,60]
rotateLeft(b,3)
print(b)
```

[40, 50, 60, 10, 20, 30]

In [4]: # Task 3

```
def remove(source,size,index):  
    for i in range(index,size-1):  
        source[i]=source[i+1]  
    source[size-1]=0  
    print(source)
```

```
source=[10,20,30,40,50,0,0]  
remove(source,5,2)
```

```
[10, 20, 40, 50, 0, 0, 0]
```

In [5]: # Task 4

```
def removeAll(source,size,element):  
    for iu in range(len(source)):  
        for ju in range(iu,len(source)):  
            if source[ju]==2:  
                for i in range(ju, size-1):  
                    source[i] = source[i + 1]  
                source[size - 1] = 0  
    print(source)
```

```
b=[10,2,30,2,50,2,2,0,0]  
removeAll(b,7,2)
```

```
[10, 30, 50, 0, 0, 0, 0, 0, 0]
```

In [3]: # Task 5

```
def Split(a):
    l=0
    r=len(a)-1
    lsum=a[l]
    rsum=a[len(a)-1]
    for i in range(len(a)):
        if lsum==rsum and l<r-1:
            l+=1
            lsum+=a[l]
        if lsum<rsum and l<r-1:
            l+=1
            lsum+=a[l]
        if rsum<lsum and l<r-1:
            r-=1
            rsum+=a[r]
    if lsum==rsum:
        return "true"
    else:
        return "false"

source=[2, 1, 1, 2, 1]
answer=Split(source)
print(answer)
```

false

In [3]: # Task 6

```
def arraySeries(s):
    n=s*s
    a=[0]*n
    temp=s
    m=temp-1
    for j in range(n):
        if s<1:
            s=temp
        a[j]=s
        s-=1
    for j in range(0,n,temp):
        for i in range(j,m+j):
            a[i]=0
        m-=1
    return a

source=arraySeries(3)
print(source)
```

```
[0, 0, 1, 0, 2, 1, 3, 2, 1]
```

In [3]: # Tasl 7

```
def bunchArray(a):
    j=0
    k=0
    for i in range(len(a)-1):
        if a[i]==a[i+1]:
            j+=1
        else:
            if k<=j:
                k=j
            j=0
    if k<j:
        k=j+1
    else:
        k+=1

    print(k)

source = [1,1,2,2,1,1,1,1,]
bunchArray(source)
```

In [5]: *#Task 8 (Linear Array)*
#Hasib's Code

```
def arrayRepetition(repeatedArray):
    test = False
    dictionary = {}
    for x in range(len(repeatedArray)):
        sum = 1
        y = x + 1
        if repeatedArray[x] not in dictionary:
            while y < len(repeatedArray):
                if repeatedArray[x] == repeatedArray[y]:
                    sum += 1
                    y += 1
            dictionary[repeatedArray[x]] = sum
        else:
            pass
    newARR = [0]*len(dictionary)
    h = 0
    for x in dictionary:
        newARR[h] = x
        h += 1
    for x in range(len(newARR)):
        k = x + 1
        while k < len(newARR):
            if dictionary[newARR[x]] == dictionary[newARR[k]] and dictionary[newARR[x]] != 1:
                test = True
                break
            else:
                k += 1
    print(test)

arrayRepetition([4,5,6,6,4,3,6,4])
```

True

In [3]: # Circular Arrays

Task 1

```
def Intersection_cir(cir_1,start_1,size_1,cir_2,start_2,size_2):
    new=[]
    lin_1=[0]*size_1
    lin_2=[0]*size_2
    for i in range(size_1):
        lin_1[i]=cir_1[start_1]
        start_1=(start_1+1)%len(cir_1)
    for i in range(size_2):
        lin_2[i]=cir_2[start_2]
        start_2=(start_2+1)%len(cir_2)
    for j in range(len(lin_1)):
        if lin_1[j] in lin_2:
            new[len(new):]=[lin_1[j]]

    return new

print(Intersection_cir([40,50,0,0,0,10,20,30],5,5,[10,20,5,0,0,0,0,5,40,15,25],8,7))
```

[10, 20, 40]

In [4]: # Task 2

```
def Palindrome(a,start,size):
    findex=start
    lindex=(start+size-1)%len(a)
    for i in range(0,len(a)):
        if a[lindex]==a[findex]:
            findex+=1
            lindex-=1

            if lindex<0:
                lindex=len(a)-1
            if findex==len(a):
                findex=0
            if findex==lindex:
                break
        else:
            print("False")
            break

    if lindex==findex:
        print("True")

source=[20,10,0,0,0,10,20,30]
Palindrome(source,5,5)
```

True

In [1]: # Task 3 *Cannot do it fully!!!*

```
random=3
p=["Asif","Fahim","Niloy","Aman","Jay",0,0,0,"Sondhi","Sakib"]
size=7
start=8
index=start
lidx=(start+size-1)%len(p)
for i in range(random+1):
    if i==1:
        n=size
        temp1=n//2
        for x in range(temp1,n):
            j=index+x
            if j<len(p):
                p[j]=p[j+1]
            if j==len(p):
                j=0
                p[j]=p[j+1]

            if j>len(p):
                j=j-len(p)
                p[j]=p[j+1]
            n-=1
        p[lidx]=0
        lidx-=1
        if lidx<0:
            lidx=len(p)-1
        print(p)

        size-=1
    else:
        l=(start+size-1)%len(p)
        temp=p[l]
        i=size-1
        while i>=1:
            j=index+i
            if j<len(p):
                p[j]=p[j-1]
            if j==len(p):
                j=0
```

```
        p[j]=p[j-1]

    if j>len(p):
        j=j-start-2
        p[j]=p[j-1]
    i-=1
    p[start]=temp

    print(p)
```

```
['Sakib', 'Asif', 'Fahim', 'Niloy', 'Aman', 0, 0, 0, 'Jay', 'Sondhi']
['Sakib', 'Fahim', 'Niloy', 'Aman', 0, 0, 0, 0, 'Jay', 'Sondhi']
['Sondhi', 'Sakib', 'Fahim', 'Niloy', 0, 0, 0, 0, 'Aman', 'Jay']
['Jay', 'Sondhi', 'Sakib', 'Fahim', 0, 0, 0, 0, 'Niloy', 'Aman']
```

In []:

In []: