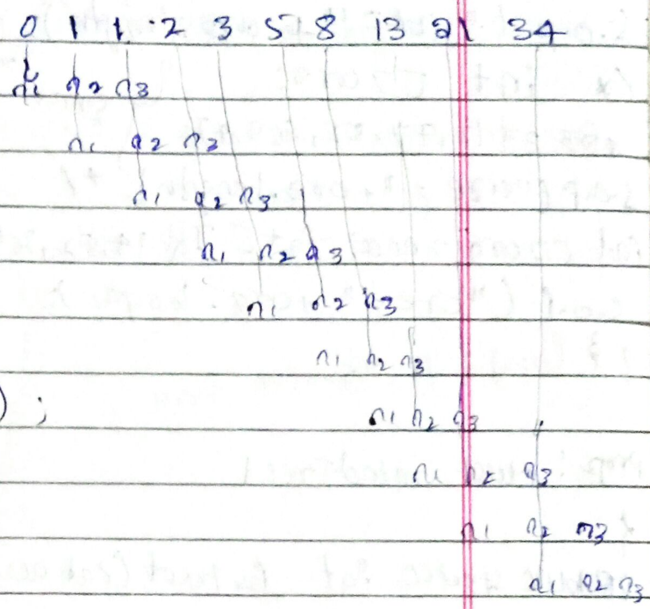
```
psvm (string[] args)
{
    int res= lcm(30, 45);
    s.o.p ("res= " +res);
  }
}
```



<u>Fibonacci</u>

```
class Fibonacciseries
{ public static void is Fibbo (0,1);
psvm (int n1, int n2)
{
s.o.p (n1+ ",");
s.o.p(n2+ ",");
for (int i=0; i<8 ; i++)
{
  int n3= n1+n2;
  s.o.p( n3+ ","); --> s.o.p (n1+",");  -> for single s.o.p all fully
  n1=n2
  n2 = n3;
}
}
psvm( string [] args
{
   isFibo (0,1);
   }
  }
```

```
int n1=0;
int n2= 1
 s.o.p (n1);
  s.o.p (n2);
  int count=0;
  while( count<8)
 {
  int n3= n1+n2;
   s.o.p (n3)
```

<u>Array programs</u>

```
class program1
{
psvm (string[] args)
{
  int [] ar= new int[5];
 s.o.p (" size = '+ar.length);
 int [] ar1;
  ar1 = new int [4];
 s.o.p ("size=" +ar1.length);
```

```
int []arr2 = { 12, 47, 52, 6, 3, 4 };
S.o.p ("size =" + arr2.length);
/* int []arr3;
arr3 = { 12, 47, 52, 6, 3, 4 };      } error
S.o.p ("size =" + arr3.length); */
int []arr4 = new int[]{ 14, 52, 36 };
S.o.p ("size =" + arr4.length);
}}
```

Prg2: class mindTree1

```
{
public static int isFact(int num)
{
 int fact = 1;
 for(int i = num; i>1 ; i--)
 {
   fact = fact * i;
 }
 return fact;
}
public static int testcase (int []arr)
{
 return (isFact(arr.length)-1);      // factorial of only length
}
psvm (String [] args)
{
 int arr [] = { 12, 6, 8, 10, 2 };
 int res = testcase(arr);
 S.o.p ( res );
}}
```

1) WAP to determine missing elements in an array

arr = { 12, 17, 18, 25, 30 }

i = 0	0 < 5	10/5/19
j = 13	13 < 16	
j = 14	14 < 16	
j = 15	15 < 16	
j = 16	16 < 16 X	

13, 14, 15, 16 — 19, 20, 21, 22, 23, 24 — 26, 27, 28, 29

i=1 1<5
j=17 17<20
i=18 17<20
i=19 19<20
j=20 20<20

prg1:-

```
class missingnumbers
{
public static void num(int[] ar)
{
for(int i=0; i<ar.length-1; i++)
{
for(int j=ar[i]+1; j<ar[i+1] ;j++) // loop to display missing numbers
{
S.O.P(j);
}}}
psvm(String[] args)
{
int[] ar={12,16,20,22,25};
num(ar);
}}
```

```
for(int i=0; i<ar.length-1; i++
{
for(int j=ar[i]+1; j<ar[i+1];
                          j++)
S.O.P(i);
}
}
```

Here ar length is a final

prg2:- write a program to reverse the ~~given~~ array element

```
int[] ar={1,2,3,4,5};
```

i=1 1<5
rev[1]=ar[3]

i=2 2<5
rev[2]=ar[4-2]

i=3 3<5
rev[3]=ar[4-3]

i=4 4<5
rev[4]=ar[4-4]

ar → | 1 | 2 | 3 | 4 | 5 |
0 1 2 3 4

rev → | 5 | 4 | 3 | 2 | 1 |

```
class reverse array
{
Public static int[] rev(int[] ar)
{
int[] rev=new int[ar.length];
for(int i=0; i<ar.length; i++)
rev[i]=ar[ar.length-1-i];
return rev;
}
```

}
public s v m (String [] args)
{
int ar [] = { 1, 2, 3, 4, 5 };
{
int arr [] = rev (ar);
for (int i = 0 ; i < arr . length ; i++)
{
s.o.p (arr [i]);
}}

i=0 0<5	ar[3] = 2	ar[i] =
temp = 1	i=2 2<5	i=0 => ar[0] <=> ar[4]
ar[0] = ar [4-0]	temp = 3	i=1 => ar [1] <=> ar[3]
ar[4] = 1	ar [2] = ar [4-2]	
i = 1 1<5	i=3 3<5	
temp = 2	temp = 2	
ar[1] = ar[3]	ar[3] = ar[4-3] ar[1]	

class Reverse Array
{
public static int [] rev (int [] ar)
{
int [] arr = new int [ar. length];
for (int i = 0; i < ar. length ; i++)
{
arr [i] = ar [ar. length - 1 - i]; → here creating new array
}
return arr ;
}

public static int [] rev2 (int [] ar)
{
for (int i = 0 ; i < ar. length/2 ; i++)
{
int temp = ar [i]; → ar [i] → same array
ar [i] = ar [ar. length - 1 - i];
ar [ar. length - 1 - i] = temp;
}
return ar ;
}

```
psvm (String [] args)
{
int arr [] = {1,2,3,4,5};
int new [] = new2 (arr);
for (int i = 0; i < new.length; i++)
{
S.o.p (new[i]);
}}}
```

prg3: write a program to determine first maximum element present in the array along with it's index.

```
class FirstMax
{
public static void First (int [] arr)
{
int max = arr[0];
int index = 0;
for (int i = 1; i < arr.length; i++)
{
if (max < arr[i])
{
max = arr[i];
index = i;
}
}
S.o.p ("the maximum element is" + max);
S.o.p ("it is present at " + index);
}
psvm (String [] args)
{
int [] arr = {12, 41, 35, 3, 16};
first (arr);
}
```

```
max = 12;
i = 1 1<5
if (12<40)
max = 40;
i = 2 2<5
if (40<13)
i = 3 3<5
if (40<35)
i = 4 4<5
if (40<4)
```

> Assignment

1) way to determine first two max numbers in an array
2) way to determine first three max number
3) way to determine nth max number in a given array, where n is decided by user.

21/8/19

Prg:- int [] ar = {12, 4, 28, 3, 15};

```
class firstmax
{
public static void first (int [] ar)
 {
 int max1 = ar[0];
 int max2 = ar[1];
 int index1 = 0;
 int index2 = 1;
 for(int i=1; i<ar.length; i++)
 {
  if (max1 < ar[i])
 {
 max2 = max1;
  index2 = index1;
 max1 = ar[i]
 index1 = i;
 }
 else if (max2 < ar[i])
  {
  max2 = ar[i];
 index2 = i;
 }
 }
 S.o.p(" the first maximum element is " + max1);
 S.o.p(" it is present at " + index1);
 S.o.p(" the second maximum element is " + max2);
 }
 psvm (String [] args)
 {
 int [] ar = {12, 4, 28, 3, 15};
 first ar;
 }
}
```

i=1 1<5
12<4
i=2 2<5
12<28
max2 = 12
max1 = 28
i=3 3<5
28<3
i=4 4<5
28<15
12<15

Prg:- WAP to sor

```
int [] ar = {12,
for (int i=0; i<ar
}
for (int j = i
}
if (ar[i] > ar
{
int temp = ar[i]
ar[i] = ar[j]
ar[j] = temp;
}
```

Check value

```
class Sortarray
{
public static i
{
for (int i=0
}
for (int j=i
{
if (ar[i] >
{
int temp=
ar[i] = ar[
ar[j] = te
}
}
}
return ar;
}
psvm (String [
{
int [] ars
int [] ar
```

21/5/19

i=1 1<5
1<4
i=2 2<5
12<28
max2=12
max1=28
i=3 3<5
28<3
i=4 4<5
28<15
12<15

Prg:- WAP to sort the given array in ascending order

```
int [] ar = {12,3,18,1,4};
for (int i=0; i<ar.length; i++)
{
for (int j=i ; j<ar.length; j++)
{
if (ar[i] > ar[j])
{
int temp = ar[i];
ar[i] = ar[j]
ar[j] = temp;
}
}
```

ar -> | 1 | 3 | 18 | 12 | 4 |

ar | 12 | 3 | 18 | 1 | 4 |
0 1 2 3 4

i=0 0<5
j=1 1<5
temp=12
ar[0]=3
ar[1]=12
j=2 2<5
3>18
j=3 3<5
3>1
j=4 4<5
1>4

i=1 1<5
j=1 1<5
12>12
j=2 2<5
12>18
j=3 3<5
12>18
swap
i=1 1<5
j=1 1<5
temp=12
ar[2]=18
12>12
ar[1]=12
i=2 2<5
12>18
ar[1]=3
j=3 3<5
3>4
ar[1]=3

Check here

```
class Sortarray
{
public static int [] sort (int [] ar)
{
for (int i=0; i<ar.length; i++)
{
for (int j=i+1; j<ar.length; j++)
{
if (ar[i] > ar[j])
{
int temp = ar[i];
ar[i] = ar[j];
ar[j] = temp;
}
}
}
return ar;
}
psvm (String [] args)
{
int [] ar = {12,3,18,1,4};
int [] ar = sort(ar);
```

wal);
val2);

```
prg3:-
for (int i=0; i<arr.length ;i++)
{
 s.o.p (arr[i]);
}
}
```

ar -> | 1 | 3 | 4 | 12 | 8 |

prg3:- nth maximum
same as previous prg.

1st max -> 4 index
2nd max -> 3 index
3rd max -> 2 index
4th max -> 1 index
5th max -> 0 index

```
public static int max (int [] ar, int n)
{
  if (n> ar.length)
  {
    return -1;
  }
  else
  {
  ar = sort (ar);
  return ar[ar.length -n];
  }
}
psvm (String [] args)
{
int[] ar = {12, 3, 8, 1, 4};
int n= 5
int res = max (ar, n);
if (res != -1)
s.o.p (res);
else
s.o.p ("invalid max number");
}
```

prg 4:- WAP to determine nth minimum no from a given array

ar -> | 1 | 3 | 4 | 12 | 18 |

same as above

```
public static int min (int [] ar, int n)
{
  if ( n> ar.length)
  {
    return -1;
  }
  else
```



```
}
ar = sort (ar);
return ar[n
}
}
```

// insert an e
array

```
int [] ar = {
int ele = 25;
int index = 2;
int arr [] =
  arr [index]
for ( int i=0
{
  arr[i] = ar
for (int i = inde
{
   arr[i] =
}
int index = 3
int element
arr [3] = 25
i=0    0<3
arr[0] = ar[0]
i=1    1<3
arr[1] = ar[
```

prg 1:- Insert class

```
{
public static
{
 if ( index <=
 {
 int [] arr =
 arr[index]
 for ( int i =
 { / /
  arr[
 }
```

```
}
arr.sort(ar);
return arr[n-1];
}
}
```

Hacker rank java datastructure
Hacker rank — java.util
Hacker rank — file in java

// insert an element at specified position in the given array

```
int[] ar = {12, 4, 13, 42, 5};
int ele = 25;
int index = 2;
int narr[] = new int[ar.length+1];
narr[index] = ele;
for (int i=0; i< index; i++)
{
narr[i] = ar[i];
}
for (int i = index+1; i< narr.length; i++)
{
narr[i] = ar[i-1];
}
```

ar -> | 12 | 4 | 13 | 42 | 5 |

narr -> | 12 | 4 | 25 | | | |

0 1 2 3 4 5

narr[2] = 25

i = 0 0<2

narr[0] = ar[0];

i = 1 1<2

narr[i] = ar[i];

22 | 5 | 4

ar | 1 | 2 | 3 | 4 | 5 |

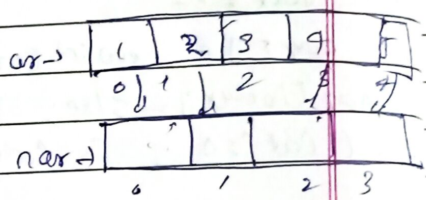
narr | 1 | 2 | 3 | 25 | 4 | 5 |

int index = 3
int element = 25
narr[3] = 25
i=0 0<3
narr[0] = ar[0];
i=1 1<3
narr[1] = ar[1]

i=2 2<3
narr[2] = ar[2]
i=3 3<3 ×
i=3 3<5
narr[4] = ar[3]
i=4 4<5
narr[5] = ar[4];
i=5 i<5 ×

```
import java.util.Scanner;
Prg1:- class TeacherMark1
{
public static int[] insert(int[] ar, int ele, int index)
{
if (index <= ar.length)
{
int[] narr = new int[ar.length+1];
narr[index] = ele;
for( int i=0; i< index; i++)
{ narr[i] = ar[i];
}
```

```
for (int i = index ; i < arr.length ; i++)
{
  narr[i+1] = arr[i];
}
return narr;
}
else
{
s.o.p ("invalid index cannot perform the operation");
   return arr;
}
}
```

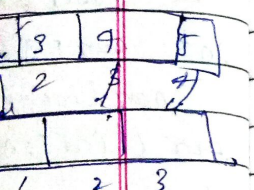



```
int arr[] = {1,2,3,4,5};
int index = 2;
int narr[] = new int [arr.length-1];
for (int i = 0 ; i < index ; i++)
{
  narr[i] = arr[i]
}
for (int i = index ; i < narr.length ; i++)
  {
     narr[i] = arr[i+1];
  }

public static int[] delete (int [] arr, int index)
{
  if (index >= arr.length)
{
int [] arr = new int [arr.length-1];
for ( int i = 0 ; i < index ; i++)
{
 narr [i] = arr[i] ;
}
for (int i = index+1 ; i < arr.length ; i++)
{
     narr[i-1] = arr[i];
}
return narr;
}
else
```

```
{
s.o.p ("invalid
return arr;
}
}
l -> array name
n -> size array
a -> queries
 insert
 x - index
 y -> element

psvm (main []
{
  Scanner s =
s.o.p ("Enter
int size = s.n
int [] arr =
for ( i = 0   i
 {
   s.o.p (
   String
 }
 s.o.p ("ente
int q = sc.ne
for (int i = 0
{
  s.o.p ("ente
String s1 = sc.
if (s1.equals
{
  s.o.p ("ent
int index =
arr = delet
}
else if (s
{
s.o.p
```




```
{
s.o.p ("invalid index cannot perform the operation");
return arr;
}
}
```

l -> array name	5
n -> size array	12 0 1 78 12
q -> queries	2
insert	delete	insert
x - index	x - index	5 23
y -> element		Delete
		0

```
psvm (main [] args)
{
Scanner s = new Scanner (System.in)
sop ("enter a array size");
int size = s.nextInt(); // size of the array
int [] arr = int [size];
for ( i = 0 ; i<size ; i++ ) // loop to take array elements
{
    sop (arr[i]) s.o.p ("enter the element at "+i+" index");
    String arr = s.next[];  arr[i] = sc.nextInt();
}
s.o.p ("enter the no of queries");
int q = sc.nextInt(); // no of queries;
for (int i=0 ; i<q ; i++) // no of steps for q times
{
  s.o.p ("enter the operation to perform");
String s1 = sc.next(); // operation to perform.
if (s1.equals ("insert"))
{
  s.o.p ("enter the index");          int size = sc.nextInt();
int index = sc.nextInt();
arr = delete (arr, index)              arr = insert (arr, size, index);
}
else if (s1.equals ("delete"))
{
  s.o.p ("enter the index");
```

```
int index = sc.nextInt();
 ar = delete (ar, index);
 }
 }// end of queries for loop
s.o.p ("the modified array ~~list~~ is");
for (int i=0 ;i<ar.length ;i++)
{
 s.o.p (ar[i]);
 }
 }
}
```

args:- print like 1
2
3
Hell ~~world~~
world

```
class genericmethod
{
public static void printArray (object []ar)
{
 for (int i=0; i<ar.length ;i++)
 {
 s.o.p (ar[i]);
 }
}
public static void main (string[] args)
{
 Integer [] ar;
 String [] str = {"hello", "world"};
 print Array (ar);
 print Array (str);
 s.o.p ("Hello world!");
 }
}
```

1) int []ar = {2,1,2,3,1,~~3~~,4};
o/p = {2,1,3,4}

2) int []ar = {3,3,1,1,5,5,3,4};
o/p {{3,1,4,5};

1) WAP to delete
n=6
i=0 0<6
i=1 1<6
if(ar[0]==ar[1]
2==2
1<5
ar[1]=ar[2];
2<5
ar[2]=ar[3];
3<5
ar[3]=ar[4];
4<5
5<5

2] ? 1
n=6
i=0 0<6
j=1, 1<6
if (ar[0] == ar[
2 == 2
k=1
while(1<5)
ar[1] = ar[2]
k=2
ar[2] = ar[3]
k=3
ar[3] = ar[4
k=4
ar[4] = ar[5

1) WAP to delete Repeated element in an Array. 23/5/19

n=6
i=0 0<6
j=1 1<6
if(ar[0]==ar[1])
2==2
1<5
ar[1]=ar[2];
2<5
ar[2]=ar[3];
3<5
ar[3]=ar[4];
4<5
5<5

n=5
i=0 0<5 x
i=1
j=2 2<5
if (ar[0]==ar[2])
2==3 x
j=3 3<5
if (ar[0]=ar[3])
2==2
k=3
3<4
ar[3]=ar[4]
k=4 4<4
n--
n=4
4<4

ar[1]=ar[2] i=0 0<6
ar[2]=ar[3] j=1 1<6
ar[3]=ar[4] ar[0]=ar[1]
ar[4]=ar[5] 2==2

i=1
j=2 2<4
if ar[1]==ar[2]
1==3
j=3 3<4
if (ar[1]==ar[3]
1==1
k=3
3<3
n--
n=3
j=4 4<3
i=2 2<3
j=3 3<3

1<5
ar[1]=ar[2]
j=2
ar[2]=ar[3]
j=3
ar[3]=ar[4]
j=4 4<5
ar[4]=ar[5];
j=5
5<5 x

{2 2 1 1 3 2}
n=6
i=0 0<6
j=1 1<6 {2 1 1 3 2}
if (ar[0] == ar[1]
2==2
k=1
while(1<5)
ar[1]=ar[2] → {2}
k=2
ar[2]=ar[3] {2 1 3 2}
k=3 → {1}
ar[3]=ar[4] → {1}
k=4
ar[4]=ar[5] {1 3 2}
→ {3}

2 2 1 1 3 2

n=6

i=0 0<6

j=1 1<6

if (2==2)

k=1 1<5

arr[1] = arr[2] 2 1

2<5

arr[2] = arr[3] 2 1 1

3<5

arr[3] = arr[4]; 2 1 1 3

4<5

arr[4] = arr[5]; 2 1 1 3 2

5<5 X

n=5 we missing 1 iteration

j=2 2<5

if (arr[0] = = arr[2])

2==1 X

j=3 3<5

if (arr[0] == arr[3]) 2 1 1 3 2

2==3 X -1 2 1 1 3

j=4 4<5

if (2==2)

k=4

k+4 X

n--

n=4

j=5 5<4

n=3

j=3 3<3 X

i=2 n=3 2<3

j= 3<3 X

i=3 n=3 3<3 X

n=4

i=1 1<4

j=2 2<4 2 1 3

if (1==1)

k=2

2<3

arr[2] = = arr[3]

3<3

n--

int [arr] = {2,2,2,1,3,1};

n=6

i=0 0<6

j=1 1<6

if (2==2)

k= 1<5

arr[1] = arr[2];

k=2 2<5

arr[2] = arr[3];

k = 3<5

arr[3] = arr[4];

4<5

arr[4] = arr[5];

5<5]

n=5

j=1 2<5

if (arr[0] == arr[2])

2==1

arr -> 2 [2] 1 3 1

0 1 2 3 4 5

j we are missing iteration

so we need to go for helper 2

class deleteRepeated

{

public static int[] ...

{

int n = arr.length; ...

// loop for ...

for (int i=0; i<n; ...

{

for (int j=i+1; j<...

{

if (arr[i] == arr[j] ...

{

// logic for shift ...

int k = j;

while (k<n-1)

{

arr[k] = arr[k+1] ...

k++;

} // end of while

n--; // ignore ...

j--; // next com... element

} // end of if

} // end of j

} // end of i

int [] new = n...

for (int i=0 i...

{

new [i] = ar...

}

return new;

} // end of ...

PSVM (String [] a...

{

int [] arr = {2,2...

int new [] = ...

for (int ...

{ S.O.P (...

y iteration

2 1 1 3 2
-1 2 1 1 3

2 1 3

[a] 1 3 1
j, 2 3 4 5
one one miss one iteration

class deleterepeated

```
{
public static int[] delete(int[] ar)
{
int n= ar.length; // length variable
// loop for comparison
for(int i=0; i<n; i++)
{
 for (int j=i+1; j<n; j++)
 {
 if (ar[i] == ar[j] // repeated element condition
 {
 // logic for shifting of element
 int k = j;
 while (k<n-1)
 {
  ar[k] = ar[k+1];
  k++;
 } // end of while
 n--; // ignore last character B/q we shifted already
 j--; // next comparison should happen from same shifted element
 } // end of if
 } // end of j loop
} // end of i loop
int [] nar = new int[n];
for (int i=0 i<n ;i++)
 {
  nar [i] = ar[i];
 }
return nar;
 } // end of method
Psvm(String [] args)
{
 int [] ar = {2,2,2,1,3,1};
 int nar [] = delete (ar);
 for(int i=0; i<nar.length ; i++)
 { s.o.p ( nar [i] ); }
```

WAP to count the no of repeated element in an array

int [] ar = {1,2,2,3,1,3,1,2};   24/5/19

ar = {1,2,2,1,2,1}
0 1 2 3 4 5

1 -> is repeated 3 times
2 -> is repeated 3 time

```
int count=0
int n=ar.length
for( int i=0; i<n; i++)
{
for(int j=i+1; i<n; j++)
{
if (ar[i] == ar[j])
{
count ++;
} //end of for loop
S.o.p( ar[i] +" is repeated "+count+" times");
} end of i loop
}
```

count=0 n=6
i=0 0<6
j=1 1<6
if(1==2)x
j=2 2<6
if((==2)x
j=3 3<6
if(1==1)
count=1;
j=4 4<6
if(1==2)x
j=5 5<6
if(1==1)
count=2;
i=6 6<6

i=1 1<6 count=0;	i=2 count=0	i=6 6<6x
j=2 2<6	j=3 3<6	i=4 4<6
if (2==2)	if(2==1)x	count =0
count=1	j=4 4<6	j=5 5<6
j=3 3<6	if(2==2)	j=5 5<6
if (2==1)x	count=1;	if(2==1) x
j=4 4<6	j=5 5<6	j=6 6<6 x
if (2==2)	if(2==1)x	i=5 5<6
count=2;	j=6 6<6 x	j=6 6<6
j=5 5<6	i=3 3<6	
if (2==1) x	count = 0	
j=6 6<6 x	j=4 4<6	
	if (1==2) x	
	j=5 5<6	
	if(1==1)	
	count=1;	

class countrepea...
{
public static vo...
{
int n=ar.lengt...
// loop f...
for(int i=0; i...
{
int count =0...
for (int j=i+1;...
{
if (ar[i] ==...
{
count++;
//logic for
int k=j;
while (k<...
{
ar[k] =ar...
k++;
}// end of ...
n--; // ignor...
j--> // reit...

} // end ...
}// end
S.o.p (ar[i]...
}// end of
}// end of
psvm(...)
{
int[] ar ...
count (...
}
}

```
class countrepeated
{
public static void counter(int[] arr)
{
int n=arr.length; // length variable
// loop for comparision
for(int i=0; i<n; i++)
 {
 int count =0;
 for(int j=i+1; j<n ;j++)
 {
 if(arr[i] == arr[j]) //repeated element condition
  {
  count++;
 //logic for shifting of elements
 int k=j;
  while(k<n-1)
 {
   arr[k] =arr[k+1];
   k++;
 }// end of while
 n--; // ignore last character
 j--; // next comparision should happen from same shifted
                  element
 } // end of if loop
 } // end of j loop
 s.o.p(arr[i] + " is repeated" +count + " times ");
 }// end of i loop
 }// end of method
psvm(...)
{
 int[] ar ={2,2,2, 1,3,1}
   count(ar);
  }
  }
```

WAP to determine the sum of non-repeated elements in an array

```
class countRepeated
{
public static int counter(int[] ar)
{
 int n=ar.length;
 int sum=0;
 for(int i=0; i<n; i++)
 {
 int count=0;
 for(int j=i+1; j<n; j++)
 {
 if(ar[i]==ar[j])
 {
 count++;
 int k=j;
 while(k<n-1)
 {
 ar[k]=ar[k+1];
 k++;
 }
 n--;
 j--;
 }
 }
 if(count==0)
 {
 sum=sum+ar[i];
 }
 }
 return sum;
}
public static void main(----)
{
 int[] ar={12,12,24,1,3,1};
 int sum=counter(ar);
 S.o.p(sum);
}
}
```

WAP con successive
```
int[] ar={ 1,
 1st
 2 in

class countrepeat
{
public static vo
{
int n=ar.length
for(int i=0;
{
 int count=1
 for(int j=i
 {
 if(ar[i]==
 {
 count++;
 int k=j;
 while(k<n-1
 {
 ar[k]=ar[k
 k++
 }
 n--;
 j--;
 }
 }
 S.o.p(ar[i]
}
}
PSVM(String[]
{
 int[] ar={
 counter(ar);
}
}
```

repeated in an array

o/p consecutive.
int[] arr = { 1,1,1, 2, 1,2,3};
1 is occurring 3 time   1 is occurring 1 time
2 is occ 1 time   2 is occurring 1 time

```
class countrepeated
{
public static void counter (int [] arr)
{
int n=arr.length; // length -> beside kids see in precious pg.
for (int i=0; i<n; i++)
{
int count=1;
for (int j = i+1; j<n ;j++)
{
if (arr[i] == arr[i+1])
{
count++;
int k=j;
while (k<n-1).
{
arr[k] = arr[k+1];
k++
}
n--;
j--;
}
}
S.o.p (arr[i]+" is occuring "+ count+"times");
}
}

psvm(String [] args
{
int [] arr={1,1,1, 2, 1, 2,2,3};
counter (arr);
}
}
```

1) i/p --> "ababcad"
o/p -> "abcd"

2) i/p -> "aaabcdaa"
o/p -> a -> 3 times
b -> 1 times
c -> 2 times
d -> 1 times

3) i/p -> "aaaabccc"
o/p -> a3b1c3

1) Delete repeated element
2) count repeated / occurring element
3) sum of non repeated element

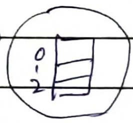
Assignment

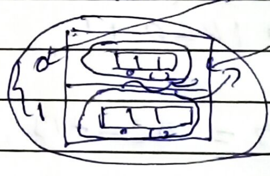
1) W.A.P matrix rotate left 90° length
2) W.A.P matrix to rotate right 90° length
3) W.A.P — " — right 180 length
4) W.A.P — " — left — " —

2D - arrays

at 15/14

Datatype v.n = value;	DATATYPE [] a = new Datatype [int size];	DATATYPE [][] ar = new DATATYPE [int size1] [int size2];
int a = 45;	int [] ar = new int [3];	int [][] ar = new int [2][3]
a ->	ar ->	ar ->





eg1:- class program1

```
psvm (String [] args)
{
    // 2D array
    int [][] ar = new int [2][3];
    ar [0][2] = 25;
    ar [1][0] = 40;
    s.o.p ("outer array size:" + ar.length);   => 2
    s.o.p (ar[0].length);                       3
    s.o.p (ar[1].length);                       3
    for (int i=0; i < ar.length; i++)           0 0 25
    {                                           40 0 0
        for (int j=0; j < ar[i].length; j++)
        {
            s.o.p (ar [i] [j] + " ");
        }
        s.o.p ();
    }
```

int [] ar = {
int [][] ar =

2) W.A.P to

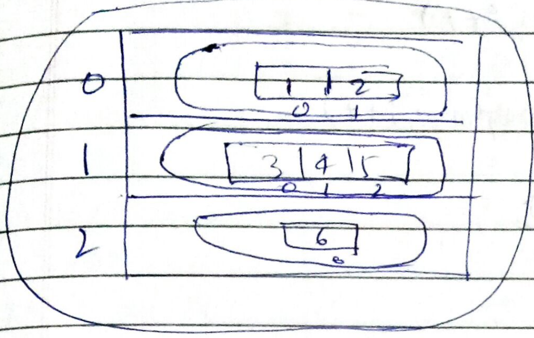
for

class clockwise
{
psvm (String []
int [][] ar
int count = 1;
for (int start =
{
for (int i =
{
ar [start] [i
}
for (int i =
{
ar [i]
}
for (int i
{
ar [end]
}
for (int i
{
ar
}
}

without new operator.

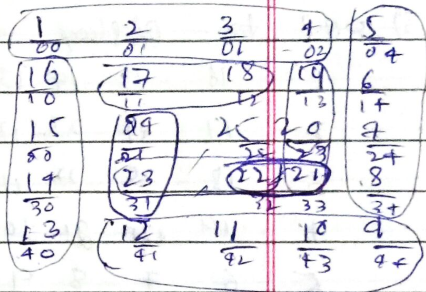
int [] arr = {1,2,3} → single

int [] [] ar = {{1,2}, {3,4,5}, {6}};



// occurring element
repeated element

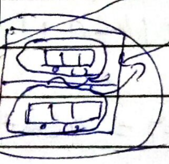
2) wap to display the numbers in clockwise direction



1st for

class clockwise direction
{
psvm(String [] args)
int [] [] ar = new int[5][5];
int count = 1;
for(int start = 0; end = ar.length-1; start < end; start++; end--)
{
for (int i = start; i < end ; i++)
{
ar[start][i] = count++;
}
for(int i = start; i < end ; i++)
{
ar [i] [end] = count++;
}
for (int i = end ; i > start ; i--)
{
ar[end] [i] = count++;
}
for (int i = end, i > start ; i--)
{
ar [i] [start] = count++;
}

27 | 5 | 14
[][]ar = new DATATYPE
[int size] [int size]
ar = new int[2][3]



DATE / /
PAGE NO.
ARUN'S

```
if (ar.length %2 != 0)
ar[ar.length/2][ar.length/2] = count;
for (int i=0 ; i<ar.length ; i++)
{
  for (int j=0 ; j<ar[i].length ; j++)
  {
    s.o.p (ar[i][j]+ " ");
  }
  s.o.p ();
}}
```

3) WAP to Display the numeric in Anti clock wise direction

1	16	15	14	13
2	17	24	23	12
3	18	25	22	11
4	19	20	21	10
5	6	7	8	9

WAP to determine

1 (00)	2 (01)	3 (02)
4 (10)	5 (11)	6 (12)
7 (20)	8 (21)	9 (22)

=>

```
i=0; i<3
j=0; 0<3
nar[0][0] = ar[0][0]
j=1  1<3
n[0][1] = ar[1][0]
j=2  2<3
n[0][2] = ar[2][0]
```

prg1:-
class Transpose matr

```
{
public static int [][]
{
  int [][] nar =
  for (int i=0 ; i
  {
    for ( int j=0
    {
      ar[i][j] =
    }
  }
  return nar;
}
psvm (String []
{
  int [][] ar =
  s.o.p (" the in
  for ( int i=0;
  {
    for (int j=0;
    {
      s.o.p
    }
    s.o.p (
```

WAP to determine transpose of given matrix 28/5/19

```
[1  2  3 ]      [1  4  7 ]
 00 01 02        00 10 20
[4  5  6 ]  =>  [2  5  8 ]
 10 11 12        01 11 21
[7  8  9 ]      [3  6  9 ]
 20 21 22        02 12 22
```

```
int [][] arr = {{1,2,3},{4,5,6},{7,8,9}}
int [][] narr = new int [3][3];
for (int i=0; i< narr.length; i++)
{
for( int j=0; j< narr.length; j++)
{
narr [i][j] = arr[j][i];
}
```

i=0; i<3 ;	j<3 3<3 X
j=0; 0<3	i=1 1<3
narr (0)(0) = arr (0)(0);	j=0 0<3 ✓
j=1 1<3 ✓	a[1][0] = arr [0][1]
n[0][1] = arr[1][0]	
j=2 2<3	
n [0][2] = arr [2][0] = 7	

you do
local wage directly

```
prg lic
class Transpose matrix
{
public static int [][] transpose (int[][] arr)
{
int [][] narr = new int[arr[0].length] [arr.length];
for (int i=0 ; i< narr.length , i++)
{
for( int j=0; j<arr.length ; j++)
{
arr[i][j] = arr [j][i];
}
}
return narr;
}
psvm (string [] args
{
int [][] arr = {{1,2,3,4},{5,6,7,8}};
S.o.p ("the input array is ..");
for ( int i=0; i<arr.length ; i++)
{
for (int j=0; j< arr[i]. length ;j++ )
{
S.o.p (arr[i][j] + " ");
}
S.o.p ();
```

```
[1 2 3 4]     [1 5]
[5 6 7 8]  => [2 6]
   2x4        [3 7]
              [4 8]
               4x2
```

Q) W.a.p to reverse the element present in the row in a given matrix

1	2	3	4	arr[0][0] <=> arr[0][3]
5	6	7	8	arr[0][1] <=> arr[0][2]
7	8	3	5	arr[1][0] <=> arr[1][3]
2	4	3	6	arr[1][1] <=> arr[1][2]

i=0 0<4

j=0 0<2

temp = arr[0][0] = 1;

arr[0][0] = arr[0][4-1-0]

= arr[0][3];

arr[0][3] = 1;

j = 1 1<2

temp = arr[0][1] = 2

arr[0][1] = arr[0][4-1-1]

= arr[0][2]

arr[0][2] = 2;

j = 2 2<2

```
class RowElementReverse
{
public static int[][] rowelement (int[][] arr)
{
for (int i=0 ; i<arr.length ; i++)
{
for (int j=0 ; j<arr[i].length/2 ; j++)
{
int temp = arr[i][j];
arr[i][j] = arr[i][arr.length-1-j];
arr[i][arr.length-1-j] = temp;
}
}
return arr;
}
public S.v.m (String[] args)
{
int[][] arr = {{1,2,3,4},{5,6,7,8},{4,7,8,5},{4,7,3,1}};
S.o.p ("array is---");
```

for (int i=0;

{

for (int j=0;

{

S.o.p (arr

}

S.o.p();

}

}

2) [0] element

public static

{

for (int i=0

{

for (int i=0

{

int temp =

arr[i][j] = ar

arr[arr.length

}

}

return arr;

}

in main me

int[][]

S.o.p ("the

for (int i=0

{

for (int i=

{

S.o.p

}

S.o.p

~~S.o.p~~ }

}

}

```
for (int i=0; i<arr.length ;i++)
{
for (int j=0 ; j< arr[i].length ; j++)
{
s.o.p (arr[i][j] + " " );
}
s.o.p ();
}
}
```

2) col element reverse same as above prg combine

```
public static int [][] colelerev (int [][] arr)
{
for ( int i=0 ; i< arr.length/2 ; i++)
{
for ( int i=0 ; i<arr[i].length ; i++)
{
int temp =arr[i][i];
arr[i][i] = arr[arr.length-1-i][i];
arr[arr.length-1-i][i] = temp;
}
}
return arr;
}
```

in main method

```
int[][] arr1 = colelerev (arr);
s.o.p (" after col elements reversed arrays is");
for ( int i=0; i<arr1.length ;i++)
{
for (int i=0 ; i < arr1[i].length ; i++)
{
s.o.p (arr1[i][i] + " ");
}
s.o.p ();
}
}
```

// anti clock rotation

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix} \Rightarrow \begin{bmatrix} 3 & 6 & 9 \\ 2 & 5 & 8 \\ 1 & 4 & 7 \end{bmatrix}$$

3 6 9
2 5 8
1 4 7

$$\begin{bmatrix} 8 & 9 \\ 5 & 6 \\ 4 & 7 \end{bmatrix}$$

pr 1:- java is easy 30/5/19

class ReverseEachWord edu is java

```
class ReverseEachWord
{
public static String reve(String s1)
String out = ""; (nospace) char ch[] = s1.toCharArray();
for(int i=0; i<ch.length; i++)
{
String res = ""; (nospace)
while(i<ch.length && ch[i] != ' ')
{
res = res+ch[i]
i++;
}
out = res+" "+out;
}
return out;
}
psvm(String[] args)
{
String out = reve("java is easy");
s.o.p("out="+out);
}
}
```

pr 2:- class ReverseEachWord i/p -> java is edu

java4 is2 edu3

```
class ReverseEachWord
{
public static String reve(String s1)
{
char[] ch=s1.toCharArray();
String out=""; (nospace)
for( int i=0; i<ch.length; i++)
{
String res=""; (nospace)
while(i<ch.length && ch[i] != ' ')
{
res = res+ch[i];
i++;
}
out = out+res+res.length()+" ";
}
return out;
```

30/5/19

to char Array (),

.. is easy

.. is2 sally

```
}
psvm(string[] args)
{
string static void main(
string out= reve (" java is easy");
s.o.p ("out=" +out);
}
}
```

31 WAP to convert to given number into words.

```
class convertWord
{
static string[] first = {"", "one", "two", "three", "four", "five", "six"
"seven", "eight", "nine", "ten", "eleven", "twelve", "thirteen", "fourteen",
"fifteen", "sixteen", "seventeen", "eighteen", "nineteen"};
static string[] second = {"", "", "twenty", "thirty", "Fourty", "fifty",
"sixty", "seventy", "eight", ninety"};
public static void print ( int num, string s1)
{
if (num>19)
{
s.o.p( second [num/10]+" "+first [num% 10]);
}
else
{
s.o.p (first [num]);
}
if (num>0)
s.o.p ("s1");
}
public static void main (string[] args)
{
int num = 123456789;
if (num ==0)
{
s.o.p ("zero")
}
```

```
else
{
print(num/10000000, "crore");
print((num/100000)%100, "lack");
print((num/1000)%100, " thousand");
print((num/100)%10, " hundred");
print(num%100, " ");
}
}
}
```

4) WAP to ~~determine~~ count the no. of words in a given string

```
i/p -> "Java is easy";                java
char[] ch = s1.toCharArray();
int count=1;
for(int i=0; i<ch.length; i++)
{
if(ch[i]==' ' && ch[i+1]!=' ')
{
count++;
}
}
return count
```

j	a	v	a			i	s			e	a	s	y
0	1	2	3	4	5	6	7	8	9	10	11	12	13

```
class Countword
{
public static int counter(String s1)
{
char ch[] = s1.toCharArray();
int count=1;
if(ch[0]==' ')
  count=0;
for(int i=0; i<ch.length-1; i++)
{
if(ch[i]==' ' && ch[i+1]!=' ')
{
count++;
}
}
```

```
return count;
}
psvm (string[] args)
{
int res = counter(" java is a programming language ");
s.o.p ("res = " + res);
}
}
```

Anagram

hackerrank java
java Anagrams

```
string res = "" res=res
for(int i=0 ; i<ch.length ; i++)
{
 if (ch[i] != ' ')
  res = res + ch[i];
}
return res;
```

31 | 5 | 19

madam IN LAW
0 1 2 3 4 5 6 7 8 9 10 11 12
i=0
if (ch[0] != ' ') 0<12
i=1 1<12
if (ch[1] != ' ')

1) remove space
2) compare length => not equal
3) convert to one case
4) sort
5) compare

in a given

```
class Anagram
{
public static string removeSpace(string s1)
{
char[] ch = s1.toCharArray();
string res = "";
for( int i=0; i<ch.length ; i++)
{
 if (ch[i] != ' ')
  res = res + ch[i];
}
return res;
}
public static string toUpperCase(string s1)
{
 char ch[] = s1.toCharArray();
 for( int i=0; i<ch.length ; i++)
 {
  if (ch[i] >= 97 && ch[i] <= 122)
   ch[i] = (char) (ch[i] - 32);
```

```
return new String(ch);
}
public static String sort(String s1)
{
char[] ch = s1.toCharArray();
for(int i=0; i<ch.length; i++)
{
for(int j=i+1; j<ch.length; j++)
{
if(ch[i] > ch[j])
{
char temp = ch[i];
ch[i] = ch[j];
ch[j] = temp;
}
}
}
return new String(ch);
}
public static boolean compare(String s1, String s2)
{
char c1[] = s1.toCharArray();
char c2[] = s2.toCharArray();
for(int i=0; i<c1.length; i++)
{
if(c1[i] != c2[i])
return false;
}
return true;
}
public static void main(String[] args)
{
String s1 = "mother in law";
String s2 = "HITLER woman";
System.out.println("the given input string");
System.out.println("s1=" +s1);
```

```
s.o.p("s2 = "+s2);
s1 = removespace(s1);
s2 = removespace(s2);
s.o.p("the string after removing spaces");
s.o.p("s1 = "+s1)
s.o.p("s2 = "+s2)
if(s1.length() == s2.length())
{
  s1 = toupper case(s1);
  s2 = to upper case(s2);
s.o.p("the after converting to one case");
s.o.p("s1 = "+s1);
s.o.p("s2 = "+s2);
    s1 = sort(s1);
    s2 = sort(s2);
s.o.p("the string after after sorting are
s.o.p("s1 = "+s1);
s.o.p("s2 = "+s2);
boolean res = compare(s1,s2);
if(res)
s.o.p("strings are anagrams");
else
s.o.p("strings are not anagrams");
}
else
{
  s.o.p("strings are not anagrams");
}
}
```

I/P -> computer
-> program
O/P -> code

```
string res = "";
for( int i=0; i<c1.length ;i++)
{
 for (int j=0; j<c2.length ;j++)
 {
  if (c1[i]!=c2[j])
    res = res+c1[i];
  }
 }
 return res.
```

i=0 0<8

j=0 0<7

c1[0] != c2[0]

'c' != 'p'

```
class stringmodify
{
public static string unique (string s1, string s2)
{
 char [] c1 = s1. toCharArray();
 char [] c2 = s2. toCharArray();
  string res = "";
 for (int i=0; i<c1.length ; i++)
 {
   int counter = 0;
 for (int j=0; j<c2.length; j++)
 {
  if (c1[i] == c2[j])
  {
    counter ++;
  }}
 if ( counter ==0)
 res = res+c1[i];
 }
 return res;
}
```

ARUN'S PAGE NO. DATE / /

```
psvm (String ... 
{
 String s1 = " co
 String s2 = " pr
 String res = un
 s.o.p ("res=" 
 }
}
```

sub string

```
{        String 
 char[] c1 = s1. to
 char[] c2 = St
 for( int i=0; i<c
 {
   int i=0;
  while (c[i
  {
    i++ ;
    j++ ;
```

i=0 0<1

j=0 k=

while (0<12 && c

j == c2

0 == 2

i = 1 1<12

k = 1 j=0

u == v

0 == 0

```
class subst
{
 public stat
 {
  char []
  char []
```

ARUN'S NO.

psvm (String [] args)
{
String s1 = " computer ";
String s2 = " program ";
String res = unique (s1, s2);
S.o.p ("res = " + res);
}
}

substring 1/6/19

{ String s1 = " java is easy " String s2 = " va ".
char[] c1 = s1. toCharArray ();
char[] c2 = s2. toCharArray ();
for (int i = 0; i < c1. length ; i++)
{
int j = 0;
while (c[i] == c2[j])
{
i++;
j++;

i = 0 0 < 12
j = 0
'j' == 'v'
i = 1 1 < 2
j = 0
'a' == 'v'
i = 2
j = 0
'v' == v,
i = 3 j = 1
a == a

i = 4 j = 2
c1[4] == c2[

i = 0 0 < 12
j = 0 k = 0
while (0 < 12 && 0 < 2 && c1[0] == c2[0])
j == v
j == c2
0 == 2

i = 2 2 < 12
k = 2 j = 0
v == v
k = 3 j = 1
a == a k = 4 j = 2
2 == 2
return true

i = 1 1 < 12
k = 1 j = 0
a == v
0 == 0

class substring
{
public static boolean sub (String s1, String s2)
{
char [] c1 = s1. toCharArray ();
char [] c2 = s2. toCharArray ();

ARUN'S PAGE NO. DATE

```
for(int i=0; i<c1.length ;i++)
{
int j=0;
int k=i;
while (k<c1.length && j<c2.length && c1[k]==c2[j])
{
k++;
j++;
}
if (j==c2.length)
{
return true;
}
}
return false;
}
psvm(string []args)
{
boolean res= sub( "java is easy ,"va");
if(res)
{
s.o.p ("it's a substring");
}
else
{
s.o.p (" it's not a substring");
}
}
```

repeated sub string

```
class substring
{
public static int sub(string s1, string s2)
{
char []  c1= s1. to charArray ();
char []  c2= s2. to charArray ();
int count =0;
```

```
for(int i =0; i
{
int j=0;
int k=i;
while ( k<c1.len
{
k++;
j++;
}
if(j==c2.len
{
count++;
}
}
return count
}
psvm( string[]
{
int res= sub(
s.o.p ("res"
}
}
```

1) h
2) d

3) u = 5
x y

1 3 => 7
3 4 =>

= c2[j])

```
for(int i=0; i<c1.length ; i++)
{
int j=0;
int k=i;
while( k<c1.length && j<c2.length && c1[k]==c2[j])
{
  k++;
  j++;
}
if(j==c2.length)
{
  count++;
 }
}
return count;
}
psvm( String[] args)
{
int res= scob("java-va-eava","va");
s.o.p ("ans "+res);
 }
}
```

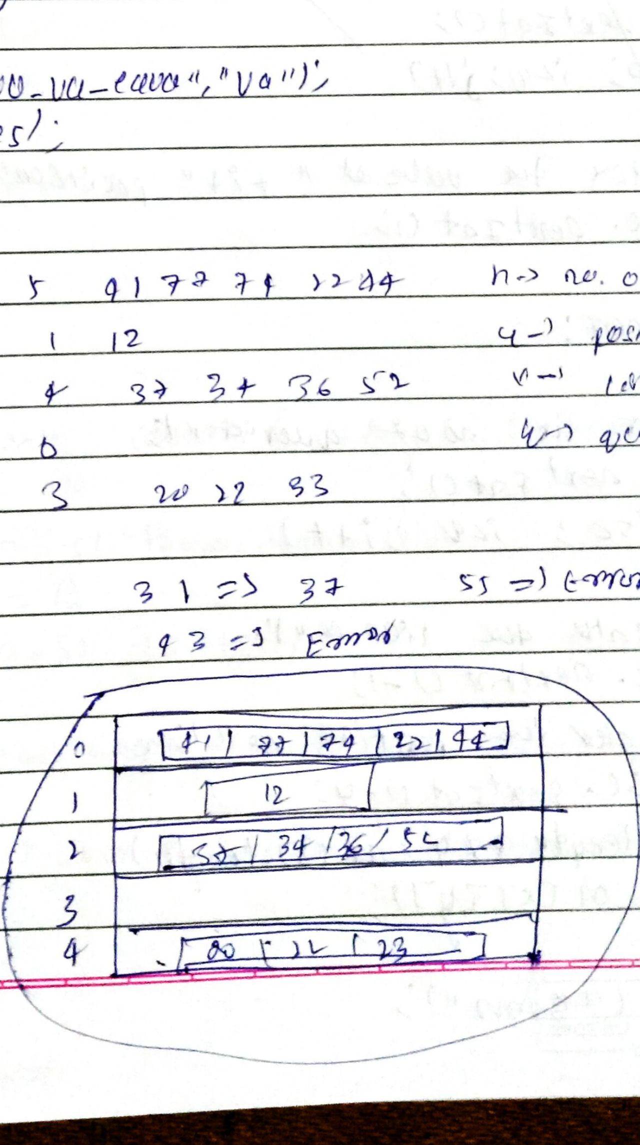
1) n
2) d
≡
3) q = 5
x y

5	41 77 74 22 44	n → no. of line
1	12	q → position no
4	37 34 36 52	v → line no
0		q → queries
3	20 22 23	

1 3 => 74
3 4 => 52

3 1 => 37
4 3 => Error

55 => Error



ARUN'S PAGE NO. DATE / /

```
import java.util.Scanner;
class HackerRank
{
public static void main(String[] args)
{
Scanner sc = new Scanner(System.in);
s.o.p("enter the no of lines");
int n=sc.nextInt();
int[][] ar= new int[n][];
for(int k=0; k<n; k++)
{
s.o.p("enter the size of the "+(k+1)+" array");
int d=sc.nextInt();
int d = sc.nextInt(); int [] arr = new int[d];
ar[k] = arr; for (int i=0 ; i<d ; i++)
{
s.o.p("enter the no of queries");
int q= sc.nextInt();
for (int j=0; j<q; j++)
{
s.o.p("enter the value at "+i+" position");
arr[i] = sc.nextInt();
}
ar[k] = arr;
}
s.o.p ("enter the no of queries");
int q=sc.nextInt();
for (int i=0 ; i<q ; i++)
{
s.o.p( "enter the line no");
int x= sc.nextInt()-1;
s.o.p ("enter the position no");
int y = sc.nextInt()-1;
if (x<ar.length && y<ar[x].length)
s.o.p ( ar[x][y]);
else
s.o.p ("error");
}
```

```
String s1 = "ja
String [] str =

str -> 0 | java
       1 | is
       2 | Easy

String s1 = "java
       s2 = "
char [] ch = s1
   int temp =0
String [] str =
 for ( int i=0
}
  String (ch
}
 res = restch
  i++;
}
str [temp+
}

class split De
{
Public static
{
  char ch
   int co
   if (ch
    count
for( int i=
 {
 if(ch[i]
  {
```

3/5/19

```
String: s1 = "java is easy";
s string [] str = s.split(" ");
                                  ("a");
str -> 0 | java |          str -> 0 | j    |
       1 | is   |                 1 | v    |
       2 | easy |                 2 | is e |
                                  3 | sy   |

string s1 = "java is easy";
       s2 = " ";
char [] ch = s1.toCharArray();
   int temp = 0;
string [] str = new string [counter (s1)];
for ( int i=0 ; i< ch.length ; i++)
}
  string (ch[i] != ' ')
}
  res = res+ch[i];
   i++;
}
 str [temp++] = res;
}

class splitDemo
{
public static int counter ( string s1, string s2)
{
    char  ch[] = s1.toCharArray ();
     int  count = 1;
     if (ch[0] == s2.charAt(0))
         count = 0;
for ( int i=0; i< ch.length -1 ; i++)
 {
  if( ch[i] == s2.charAt(0) && ch[i+1] != s2.charAt(0))
   {
        count ++;
   }
  return count;
```

```
public static string[] split (string s1, string s2)
{
string str[] = new string[counter(s1, s2)];
char ch[] = s1. tocharArray ();
int temp =0;
for (int i=0; i<ch.length ; i++)
{
 string res="";
while (i<ch.length && ch[i] ! = s2.charAt(0))
{
res=res+ch[i];
i++;
}
str[temp++]= res;
}
return str;
}
psvm (string[] args)
{
string s1 = " this is code";
string[] str= splits(s1, "i");
for (int i= 0; i<str.length ; i++)
{
  s.o.p (str[i]);
  }
 }
}
```

in that program we calling count word program

args :-

call classes.

```
class stringDemo
{
psvm static void main (string[] args)
{
 string s1= "Java is a programming language";
string str[]= splitDemo. split (s1, " ");
string out= "";
```

```
for (int i=0 ;
{
  if (i%2 ==
out =out + str
else
out =out + st
}
 s.o.p (out);
 }
 }
  i/p :- " jav
      " Jav

class string De
{
public static
{
 char ch[] =
 string out= "
 for (int i=0;
{
  string res
 while (i<
  {
  if (res
  }
  if (ch[
  ch[i] =
  }
  res = re
  i++)
 }
out = out +res
 }
 return ou
}
```

```
for (int i=0; i<str.length; i++)
{
  if (i%2==0)
out = out + str[i]+" ";
else
out = out + StringReverse.reverse(str[i])+" ";
}
S.O.P(out);
}
}
```

I/P :- "Java is a programming language"

"Java Is A Programming

```
class StringDemo3
{
public static string firstcharupper(string s1)
{
char ch[] = s1.tocharArray();
string out = "";
for(int i=0; i<ch.length; i++)
{
  string res = "";
  while(i<ch.length && ch[i] != ' ');
  {
    if(res.length()==0)
  {
    if(ch[i]>=97 && ch[i]<=122)
    ch[i] = (char)(ch[i]-32);
  }
    res = res+ch[i];
    i++;
  }
out = out+res+" ";
  }
  return out;
}
```

```
psvm(String[] args)
{
String s1 = "learn is a programming language";
String res = firstCharUpper(s1);
s.o.p("res=" + res);
}
}

// Logic to remove the vowels from the given string.
class Vowels
{
public static String removeVowels(String s1)
{
char[] ch = s1.toCharArray();
String ocat = "";
for(int i=0; i<ch.length; i++)
{
if(ch[i] != 'a' && ch[i] != 'e' && ch[i] != 'o' && ch[i] != 'i'
    && ch[i] != 'u')
{
}
}
return ocat;
}
psvm(String[] args)
{
String res = removeVowels("Education");
s.o.p("res" + res);
}
}
```