```
public class A implements
Serializable{
B x = new B();
}

Public class B{
}
```

HashMap h = new HashMap

A

B

C

h.put
h.get

Public class MyNode{
Int data;
MyNode next;
MyNode prev;

List<MyNode> myPointers

```
int arr1 = {1,5,7,3,9,10,9}
int arr2 = {20,5,7,33,31,25,30}
x=38
TreeSet<integer>
sortedSet=new TreeSet();
38-20=18

for(int I=0;i<arr1.length;i++)

{

for( int j=0;j<arr2.length;j++)
{
if(array1.[I]+array2[j])
{
System.out.print(“” array1.[I]+
“” +array2[j]))
```

```
}
Else{

sortedSet.add(array1.[I]
+array2[j])
}

}

Else{

}
}
}
}

String[]  words = ['The','man',
'in', 'black', 'fled', 'across', 'the',
'desert','and', 'the',
```

```
'guednslinged', 'followed'];

int frequency =
getLetterFrequency(words,"ed
");
{int count=0;
for(int I=0;i<words.length;i++)
{

if(words.charAt(i).contains(“ed
”)){
count++;
}
}
Return count;
}
```

employee	manager	sal	Dep
Suman	Adarsh	5	1
Adarsh	raja	10	2
raja	pavan	50	1
pavan	Shaji	30	2
Shaji	---	100	1

```
Select  count(*) from
Employee
Group by dep
```

Having sal>20

Select employeeName from
Employee e
Join Employee e1
On e.employeeName

Select Max(salary) from
employee
Where sal <> (Select max(sal)
from employee);