Theory Question:

1. What is JVM and explain me the Java memory allocation

JVM(Java Virtual Machine) is an abstract machine. It is a specification that provides runtime environment in which java byte code can be executed.
The JVM allocates Java heap memory from the OS and then manages the heap for the Java application. When an application creates a new object, the JVM sub-allocates a contiguous area of heap memory to store it.
An object in the heap that is referenced by any other object is "live," and remains in the heap as long as it continues to be referenced.
Objects that are no longer referenced are garbage and can be cleared out of the heap to reclaim the space they occupy. The JVM performs a garbage collection (GC) to remove these objects, reorganizing the objects remaining in the heap.

2. What is Polymorphism and encapsulation?

Polymorphism: Polymorphism is the ability of an object to take on many forms. The most common use of polymorphism in OOP occurs when a parent class reference is used to refer to a child class object.
Method overriding is a perfect example of Polymorphism.
Encapsulation in Java is a mechanism of wrapping the data.In encapsulation the variables of a class will be hidden from other classes, and can be accessed only through the methods of their current class, therefore it is also known as data hiding.
To achieve encapsulation in Java

- Declare the variables of a class as private.(how to achieve)
- Provide public setter and getter methods to modify and view the variables values.(how to access)

3. What is method overloading and Method over riding?

OverLoading	OverRiding
It Deals with Multiple Methods in same Class	It Deals with 2 methods, one in parent class and other in Child class
It has same name but different signature	It has same name and same signature, but have different implementation
It allows you to define same operation for different data	It allows you define the same operation in different ways for different object type
	Implements Dynamic polymorphism

4. Why string is Immutable?

Once you create any object you can not modified. When a string is created and if the string already exists in the constant pool, the reference of the existing string will be returned, instead of creating a new object and returning its reference.

5. What is the difference between String and String buffer, and StringBuilder?

feature	String	StringBuffer	StringBuilder
Storage Area	String Constant Pool	Heap	Heap
Modifiable	No (Immutable)	Yes	Yes
Thread Safe	Yes	Yes	No
Performance	Fast (Because coming from constant pool)	Slow (because Thread-safe)	Fast

6. What is the difference between array and array list?

Feature	Array	ArrayList
Resizable	Fixed, Static in Size Can not change size, but can copy elements to bigger size array	Dynamic in size. First defined as 10 but will grow by 50% Once it reaches 0 .75 load factor
Data Type	Primitive and object	Only object
Iterating values	Use for Loop	Use iterator to iterate
Length	Can find with length method	Provides by the size()
Generics	Can not use as array instance knows what kind of data type it can hold.	ArayList use Generics to type safety

7. What is the difference between hash map and Hash table?

HashMap	HashTable
Not Synchronized	Synchronized
Not Thread Safe	Thread Safe
Allows Null key and Null values	Does not allow null key or value
Fast	Slow
It inherits Abstract class	it inherits Dictionary class

8. What is a vector in Java?

Vectors implements a dynamic array. It is similar to array list, but with two difference. Vector is Synchronized. Vector contains many legacy methods that are not part of the collection framework

9. What is set in java?

Set is a collection that cannot have duplicate element. Set interface contains only method inherited from collection and adds the restriction that duplicate elements are prohibited. Set have its implementation in various classes like hashSet, TreeSet, LinkedHashSet.

10. What is an abstract class?

A class that declared with Abstract keyword, is known as abstract class. Abstract classes may or may not contain abstract methods, methods without body.

Class can not be instantiated.

To use abstract class you need to inherit it from another class, and provide implementation of abstract methods in it.

can have public, private, protected.

can have static, final, static final variables.

Abstract class can have Constructor:-

when any class extends an abstract class, the constructor of sub class will invoke the constructor of super class either implicitly or explicitly. This chaining of constructors is one of the reasons abstract class can have constructors in Java.

11. What is an interface?

Interface is 100% abstract class, all methods declared abstract by default.

Can not be instantiated.

It does not contain any constructor .

it has only public methods(abstract default) and static final variables.

A class can implements (not extends) many interfaces.

12. Why Java is Platform independent?

Platform independent is write once and run anywhere.

source code(.java file) is compiled by help of javaC compiler to .class file which contains source code in form of byte code(machine language for JVM), so this byte code is dependent on jam not on operating system, To execute you need to have jam on any OS.

13. What are access modifiers? Give me an example?

Access Modifier	within class	within package	outside package by subclass	outside package
Private	Y	N	N	N
Default	Y	Y	N	N
protected	Y	Y	Y	N
Public	Y	Y	Y	Y

14. What are java exceptions? Give me an example

An exception is a problem that arises during the execution of a program. When an Exception occurs the normal flow of the program is disputed and the program terminated abnormally

Checked exception: checked at compile time.

Unchecked exception: Occurs at run time for example:ArrayIndexOutOfBoundsException, FileNotFoundException, Null pointExceptions,

15. What is the difference between throws and throwable?

Throw	Throws
Throw keyword is used to throw an exception	Throws keyword is used to declare an exception
Throw is used within the method	Throw is used with method signature
Throw is followed by instance	it shows the method throw exception(given Name)
You can not throw multiple exceptions	You can declare multiple exceptions

16. What is the difference between Error and exception?

An error is an irrecoverable condition occurring at runtime like out of memory error. These kind of JVM errors cannot be handled at runtime.
Exceptions are because of condition failures, which can be handled easily at runtime.

17. What is the difference between Error, throwable and exception?

Throwable is a super class, and Error and Exception are subclass of it.

Error	Exception
All Errors are type of java.lang.Error	All Exception are type of java.lang.Exception
Errors are unchecked happened at run time	Exceptions can be checked and unchecked type.

Error	Exception
Impossible to recover from error	Possible to recover By using try-catch and throws
Program will terminated if error occurs.	it will continue for execution after recovery.
EX: StackOverFlowError, OutOfMemoryError	IOException, NullPointException,

18. What are collection APIs, give me an example?

Java collection API is a well defined set of interfaces and classes, which deals with storing and manipulating group of data.
it takes the group of data as a single unit. Groups are based on the logic in which the data is stored

19. What is the difference between final and finally?

F**inal**: final is a keyword. The variable declared as final should be initialized only once and cannot be changed. Java classes declared as final cannot be extended. Methods declared as final cannot be overridden.
Finally: finally is a block. The finally block always executes when the try block exits. This ensures that the finally block is executed even if an unexpected exception occurs. But finally is useful for more than just exception handling - it allows the programmer to avoid having cleanup code accidentally bypassed by a return, continue, or break. Putting cleanup code in a finally block is always a good practice, even when no exceptions are anticipated.
Finalize: finalize is a method. Before an object is garbage collected, the runtime system calls its finalize() method. You can write system resources release code in finalize() method before getting garbage collected.

20. Will java supports multiple inheritance?

No, Java does not support multiple inheritance. Diamond problem can happen with multiple inheritance.

21. What are the different types of interface?

List, Set, Map, Queue

22. What are wrapper class? Give me an example

Wrapper Class in java provides the mechanism to convert primitive into object and object into primitive.

23. What is boxing and unboxing in Java? Explain with an example?

Auto boxing and unboxing feature do conversion of primitive and object automatically.
Autoboxing: The automatic conversion of primitive into object is known as Autoboxing.
UnBoxing: The automatic conversion of object to primitive is known as Unboxing.

```
int a=20;
Integer i=Integer.valueOf(a);//converting int into Integer
Integer j=a;//autoboxing, now compiler will write Integer.valueOf(a) internally


Integer a=new Integer(3);
int i=a.intValue();//converting Integer to int
int j=a;//unboxing, now compiler will write a.intValue() internally
```

24. Explain for each loop

This loop is used like a for loop but syntax is different, by using this you can access the all elements of data array.

```
for(int a : dataArray){
SOP(a)
}
```

it will print all element of the array

25. What are iterators, explain with an example

iterators are used to display all the elements in collection , they are same as loop in array.
iterators can used by creating a object of it using .iterator()
Set up the loop that makes a call to hasNext().
within the loop, get each element by calling next().

```
Iterator itr = al.iterator();
      while(itr.hasNext()) {
         Object element = itr.next();
         System.out.print(element + " ");
      }
```

26. How do you access Private variables in different class?

By using getter and setter methods. You can call this methods by using instance of that class.

27. Prepare for one java program to write on the board

Sorting algorithm: merge, quick, selection,

Search: Linear, Binary

28. What is Constructor Over loading?

Constructor overloading is a technique in java in which a class can have any number of constructor that differ in parameter lists.
Compilers differentiate in run time by parameter been passed.
Constrictor overloading helps to create different parameter object of the class.

29. Whit out using sync key word how do you perform synchronization?

`Thread.sleep` causes the current thread to suspend execution for a specified period. This is an efficient means of making processor time available to the other threads of an application or other applications that might be running on a computer system.

30. What is Super keyword ? when and where do you use it ?

The super keyword in java is a reference variable that is used to refer immediate parent class object
It is used inside a sub-class method definition to call a method defined in the super class. Private methods of the super-class cannot be called. Only public and protected methods can be called by the super keyword. It is also used by class constructors to invoke constructors of its parent class.

Programing Questions:

1. Find out the number of days in between two given dates ?

```
public class _01_DaysBetweenTwoDates { public static void main(String[] args) {

} }

Calendar c1=Calendar.getInstance(); c1.set(2016,01,01 );
Calendar c2=Calendar.getInstance(); c2.set(2016,01,31);

Date d1=c1.getTime(); Date d2=c2.getTime();

long diff=d2.getTime()-d1.getTime();
int days=(int)(diff/(1000*24*60*60));
System.out.println("Difference between days are: "+days);
```

************************OUTPUT-1***************************

```
Difference between days are: 30
```

2. How to divide a number by 2 without using / operator?

```
public class _02_DivideWithoutDivOperator {

public static void main(String[] args) {
System.out.println("Please enter any number you want to di-
vide by 2: ");
Scanner input = new Scanner(System.in);
int num = input.nextInt(); System.out.println(num+ "/2 = " +
(num>>1));

} }
```

*******************OUTPUT-2*****************

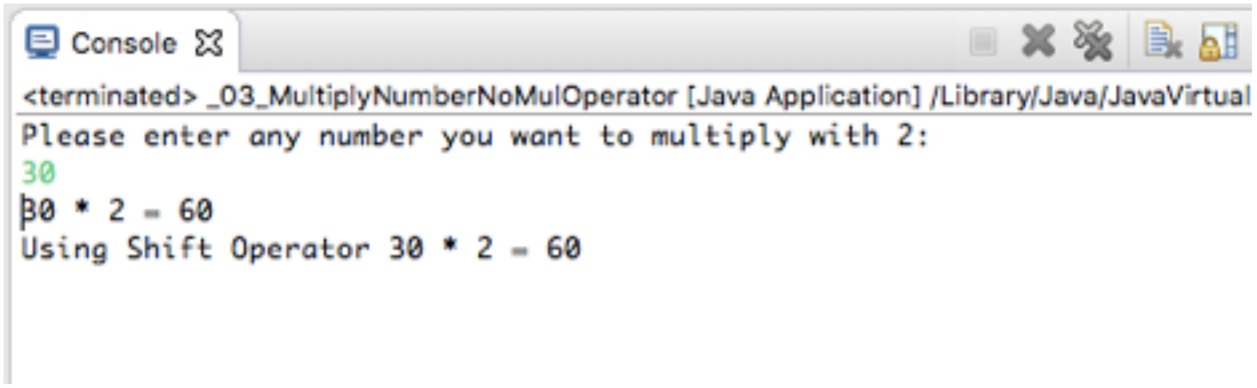
```
Please enter any number you want to divide by 2:
12
12/2  = 6
```

3. How to multiply a number by 2 without using * operator?

```
public class _03_MultiplyNumberNoMulOperator { public
static void main(String[] args) {

System.out.println("Please enter any number you want to
multiply with 2: ");

Scanner input = new Scanner(System.in);

int num = input.nextInt();

int multi = num + num;

System.out.println(num+" * 2 = "+ multi);

System.out.println("Using Shift Operator "+num+" * 2 = "+
(num<<1));

} }
```

**********************OUTPUT-3*****************
****

```
Please enter any number you want to multiply with
2:
12
12 * 2 = 24
Using Shift Operator 12 * 2 = 24
```



```
<terminated> _03_MultiplyNumberNoMulOperator [Java Application] /Library/Java/JavaVirtual
Please enter any number you want to multiply with 2:
30
30 * 2 = 60
Using Shift Operator 30 * 2 = 60
```

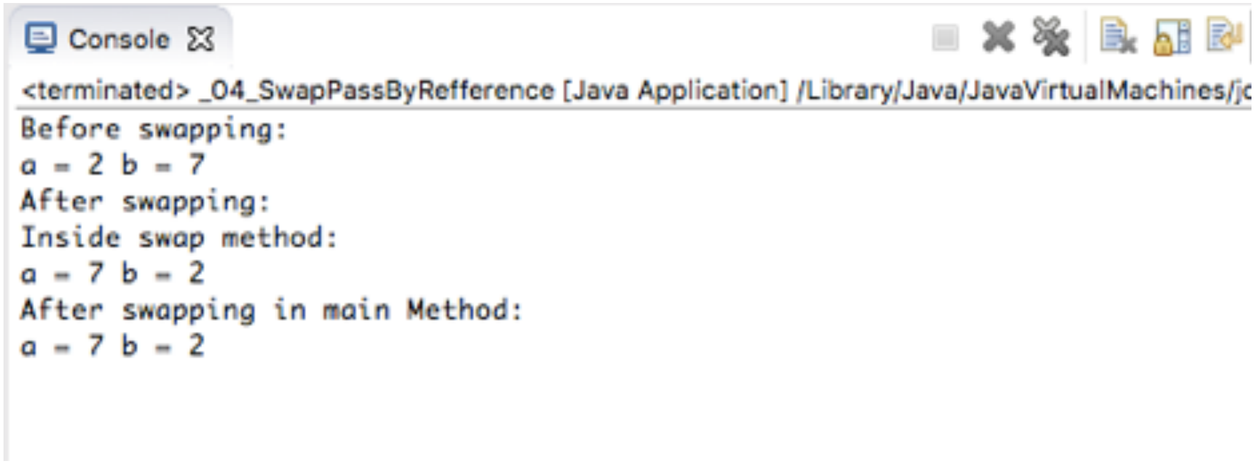
4. How to swap two variables, by using pass by reference method ?

```
public static void main(String[] args) {
swapByReff ob = new swapByReff(2, 7);
System.out.println("Before swapping: ");
System.out.println("a = "+ob.a+" b = "+ob.b); System.out-
.println("After swapping: "); ob.swap(ob);

System.out.println("After swapping in main Method: ");

System.out.println("a = "+ob.a+" b = "+ob.b); }

}
```

```
class swapByReff{ int a, b;

//Constructor swapByReff(int i, int j){

a = i;

b = j; }

void swap(swapByReff ob){ int temp;

temp = ob.a;
ob.a = ob.b;
ob.b = temp;
System.out.println("Inside swap method: "); System.out-
.println("a = "+ob.a+" b = "+ob.b); }
```

*******************OUTPUT-4******************

```
Before swapping: a=2b=7
After swapping: Inside swap method: a=7b=2

After swapping in main Method: a=7b=2
```



```
<terminated> _04_SwapPassByRefference [Java Application] /Library/Java/JavaVirtualMachines/jc
Before swapping:
a = 2 b = 7
After swapping:
Inside swap method:
a = 7 b = 2
After swapping in main Method:
a = 7 b = 2
```

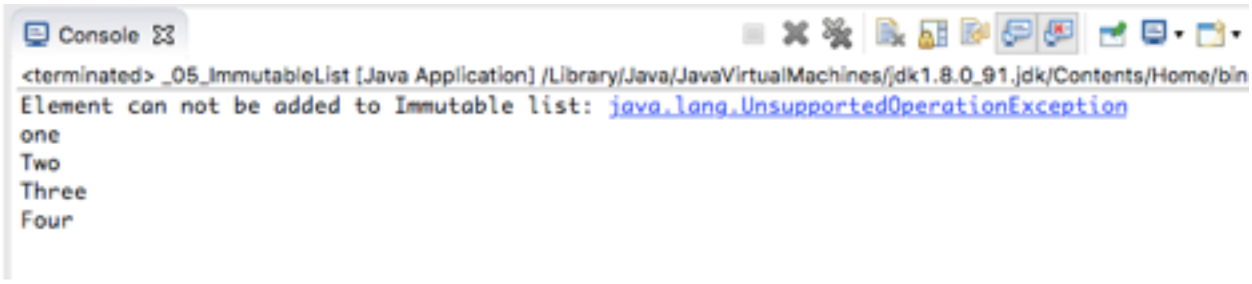
5. How to make a list immutable?

```
public class _05_ImmutableList {

public static void main(String[] args) {
List<String> list = new ArrayList<String>();
list.add("one");
list.add("Two");
list.add("Three");
list.add("Four");
List<String> immutableList =
Collections.unmodifiableList(list); try {

immutableList.add("5");

immutableList.add("Five"); } catch (Exception e) {

//e.printStackTrace();

System.out.println("Element can not be added to Immutable
list: ja- va.lang.UnsupportedOperationException");

}
for(String s : immutableList){

System.out.println(s); }

}
```

*******************OUTPUT-5******************

```
Element can not be added to Immutable list: ja-
va.lang.UnsupportedOper- ationException
```

```
one
Two

Three

Four
```



```
Console
<terminated> _05_ImmutableList [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_91.jdk/Contents/Home/bin
Element can not be added to Immutable list: java.lang.UnsupportedOperationException
one
Two
Three
Four
```

6. Write a sample code to reverse Singly Linked List by iterating through it only once.

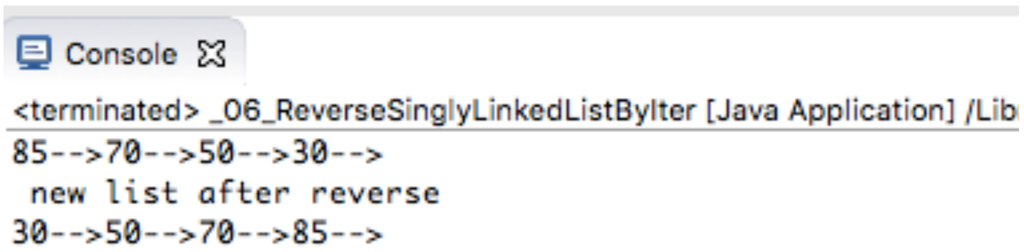
public class _06_ReverseSinglyLinkedListByIter {

public static void main(String[] args) {

_06_ReverseSinglyLinkedListByIter list = new _06_Revers-eSinglyLinkedListBy- Iter();

list.head = new Node(85);
list.head.next = new Node(70); list.head.next.next = new Node(50); list.head.next.next.next=new Node(30); list.print-List(head);
System.out.println("\n new list after reverse "); Node head1 = list.reverseLinkedList(head); list.printList(head1);

}
Node reverseLinkedList(Node node){


Console

```
<terminated> _06_ReverseSinglyLinkedListByIter [Java Application] /Lib
85-->70-->50-->30-->
 new list after reverse
30-->50-->70-->85-->
```

Node head = node; Node prev = null; Node next = null; while(head != null){

next = head.next; head.next = prev; prev = head; head = next;

}
node = prev;

return node; }

static Node head; static class Node{

int data; Node next; Node(int d){

data=d;

next = null; }

}
void printList(Node node){

while(node != null){ System.out.print(node.data+"-->");
node = node.next;

} }

}
****************OUTPUT-6********************

```
85-->70-->50-->30-->
```

```
 new list after reverse
30->50-->70-->85-->
```

7. Write a program to implement ArrayList and Linked list

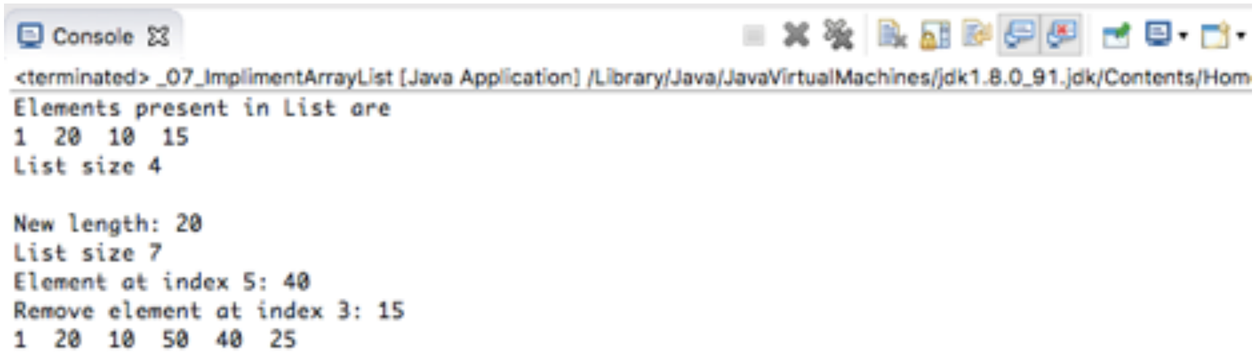
```
public class _07_ImplimentArrayList {

public static void main(String[] args) { MyArrayList list =
new MyArrayList(); list.add(1);
list.add(20);

list.add(10);
list.add(15);
System.out.println("Elements present in List are"); for(int i
=0; i< list.Listsize();i++){

System.out.print(list.get(i) + " "); }

System.out.println("\nList size "+ list.Listsize());
list.add(50);
list.add(40);
list.add(25);

System.out.println("List size "+ list.Listsize()); System.out-
.println("Element at index 5: " + list.get(5)); System.out-
.println("Remove element at index 3: " + list.remove(3));
```

```
for(int i =0; i< list.Listsize();i++){ System.out.print(list.get(i)
+ " ");

} }

}

class MyArrayList{

private Object[] myList; private int size =0;

public MyArrayList(){
myList = new Object[10]; //created a list size 10

}
public Object get(int index){

if(index < size){
return myList[index];

} else {
throw new ArrayIndexOutOfBoundsException();

}}
public void add(Object obj){

if(myList.length-size <= 5){ increaseListSize();

}

myList[size++] = obj; }

public Object remove(int index){ if(index < size){
```

```
Object obj = myList[index]; myList[index] = null;
int tmp = index;

while(tmp < size){
myList[tmp] = myList[tmp+1]; myList[tmp+1] = null;

tmp++; }

size--;

return obj; } else {

throw new ArrayIndexOutOfBoundsException(); }}

private void increaseListSize() {
myList = Arrays.copyOf(myList, myList.length*2); Sys-
tem.out.println("\nNew length: "+myList.length);

}
public int Listsize(){

return size; }}
```

*******************OUTPUT-7**********

```
Elements present in List are
1  20  10  15
List size 4
New length: 20
List size 7
Element at index 5: 40
Remove element at index 3: 15
1  20  10  50  40  25
```



<terminated> _07_ImplimentArrayList [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_91.jdk/Contents/Hom

```
Elements present in List are
1  20  10  15
List size 4

New length: 20
List size 7
Element at index 5: 40
Remove element at index 3: 15
1  20  10  50  40  25
```

```
public class _07_ImplimentLinkedList {
     public static void main(String[] args) {
           linkedList list = new linkedList();
           list.display();
           list.insertAtStart(10);
           list.display();
           list.insertAtEnd(20);
           list.display();
           list.insertAtEnd(25);
           list.display();
           list.insertAtPosition(15,3);
           list.display();
           System.out.println("List size: 
"+list.getSize());
           list.deleteAtPosition(4);
```

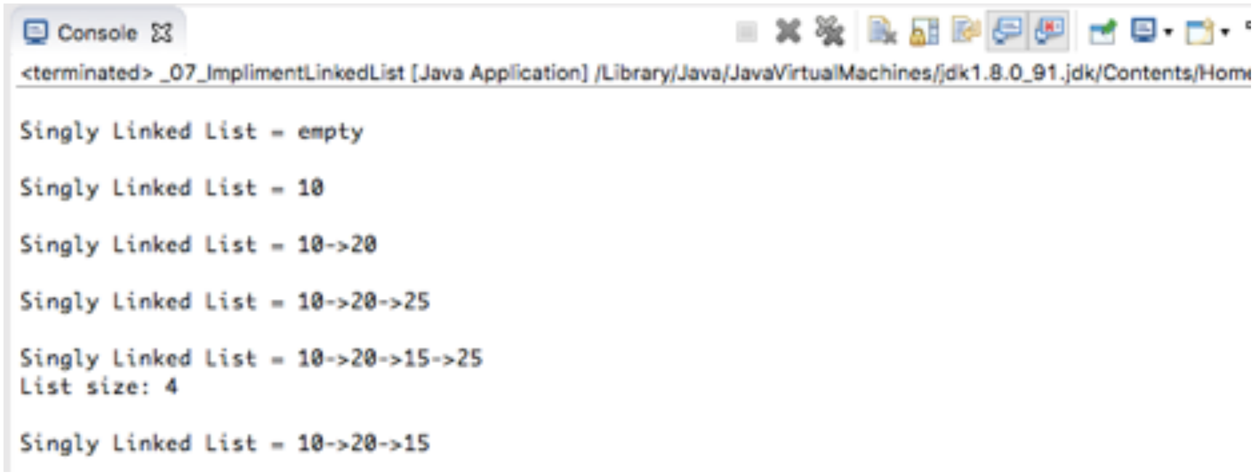
```
            list.display();
      }
}

class Node{
      int data;
      Node link;
      //constructor no arg
      public Node(){
data = 0;

link = null; }

      //constructor with arg
      public Node(int d, Node l){
data = d;

link = l; }

      public int getData() {
            return data;
      }
      public void setData(int d) {
data = d; }

      public Node getLink() {
            return link;
      }
      public void setLink(Node l) {
link = l; }

}

class linkedList{
      Node start;
```

```
     Node end;
     int size;
     //Constructor
     public linkedList(){
           start = null;
           end = null;
           size = 0;
}

     public boolean isEmpty(){
           return start == null;
     }
     public int getSize(){
           return size;
     }
     public void insertAtStart(int val){
           Node nptr = new Node(val, null);
           size++;
           if(start == null){
start = nptr;

end = nptr; }

           else{
                nptr.setLink(start);
                nptr = start;
           }
}

     public void insertAtEnd(int val){
           Node nptr = new Node(val, null);
           size++;
           if(start == null){
start = nptr;
```

```
end = nptr; }
     else{
           end.setLink(nptr);
end = nptr; }
}
public void insertAtPosition(int val, int pos){
     Node nptr = new Node(val, null);
     Node ptr = start;
     pos = pos-1;
     for(int i = 1; i<size; i++){
           if(i == pos){
                Node temp = ptr.getLink();
                ptr.setLink(nptr);
                nptr.setLink(temp);
                break;
}
           ptr = ptr.getLink();
     }
size++; }
public void deleteAtPosition(int pos){
     if(pos == 1){
           start = start.getLink();
           size--;
           return;
     }
     if(pos == size){
           Node s = start;
           Node t = start;
           while (s != end)
           {
```

```
t = s;

                s = s.getLink();
           }
           end = t;
           end.setLink(null);
           size--;
           return;
     }
     Node ptr = start;
     pos = pos - 1 ;
     for (int i = 2; i < size-1; i++)
     {
if (i == pos) {

                Node tmp = ptr.getLink();
                tmp = tmp.getLink();
                ptr.setLink(tmp);
                break;
}

           ptr = ptr.getLink();
     }
size-- ; }

/*  Function to display elements  */
public void display()
{
     System.out.print("\nSingly Linked List = ");
     if (size == 0)
     {
           System.out.print("empty\n");
return; }
```

```
      if (start.getLink() == null)
      {
            System.out.println(start.getData() );
return; }

      Node ptr = start;
      System.out.print(start.getData()+ "->");
      ptr = start.getLink();
      while (ptr.getLink() != null)
      {
            System.out.print(ptr.getData()+ "->");
            ptr = ptr.getLink();
      }
      System.out.print(ptr.getData()+ "\n");
}
      Node reverseList(Node node){
            Node pre = null;
            Node current = node;
            Node next = null;
            while(current != null){
                 next = current.link;
                 current.link = pre;
                 pre = current;
                 current = next;
            }
            node = pre;
            return node;
}

}
********************OUTPUT-7**************
Singly Linked List = empty
Singly Linked List = 10
Singly Linked List = 10->20
Singly Linked List = 10->20->25
Singly Linked List = 10->20->15->25
List size: 4
Singly Linked List = 10->20->15
```



```
<terminated> _07_ImplimentLinkedList [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_91.jdk/Contents/Home

Singly Linked List = empty

Singly Linked List = 10

Singly Linked List = 10->20

Singly Linked List = 10->20->25

Singly Linked List = 10->20->15->25
List size: 4

Singly Linked List = 10->20->15
```

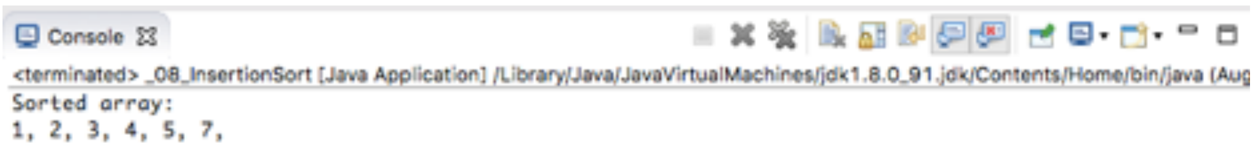
8. Write a program for Insertion Sort in java.

```
public class _08_InsertionSort {
public static void main(String[] args) {

// TODO Auto-generated method stub int[] array =
{7,2,4,1,5,3}; insertionSort(array);
for(int i = 0; i < array.length; i++ ){

System.out.print(array[i] +", "); }

}
public static void insertionSort(int[] array){

int value,key,i; for(i=1;i<array.length;i++) {
```

```
value=array[i];
key=i;
while(key>0 && array[key-1]>value) {

array[key]=array[key-1];

key--; }

array[key]=value; }}}
```

**********************OUTPUT-8*****************

```
Sorted array:
1, 2, 3, 4, 5, 7,
```



```
<terminated> _08_InsertionSort [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_91.jdk/Contents/Home/bin/java (Aug
Sorted array:
1, 2, 3, 4, 5, 7,
```

9. Write a program to get distinct word list from the giv-en file.

```
public class _09_DistinctWordsFromFile {
public static void main(String[] args) throws Exception {

File fPath = new File("/Users/dimple/Desktop/FileRead.txt");

readFile(fPath); }

private static void readFile(File fPath) throws IOException {
// Construct BufferedReader from FileReader
BufferedReader br = new BufferedReader(new
```

FileReader(fPath)); List<String> wordList = new ArrayList<String>();

String line = null;
while ((line = br.readLine()) != null) {

StringTokenizer st = new StringTokenizer(line, " ,.;:\"");
while(st.hasMoreTokens()){

br.close(); }

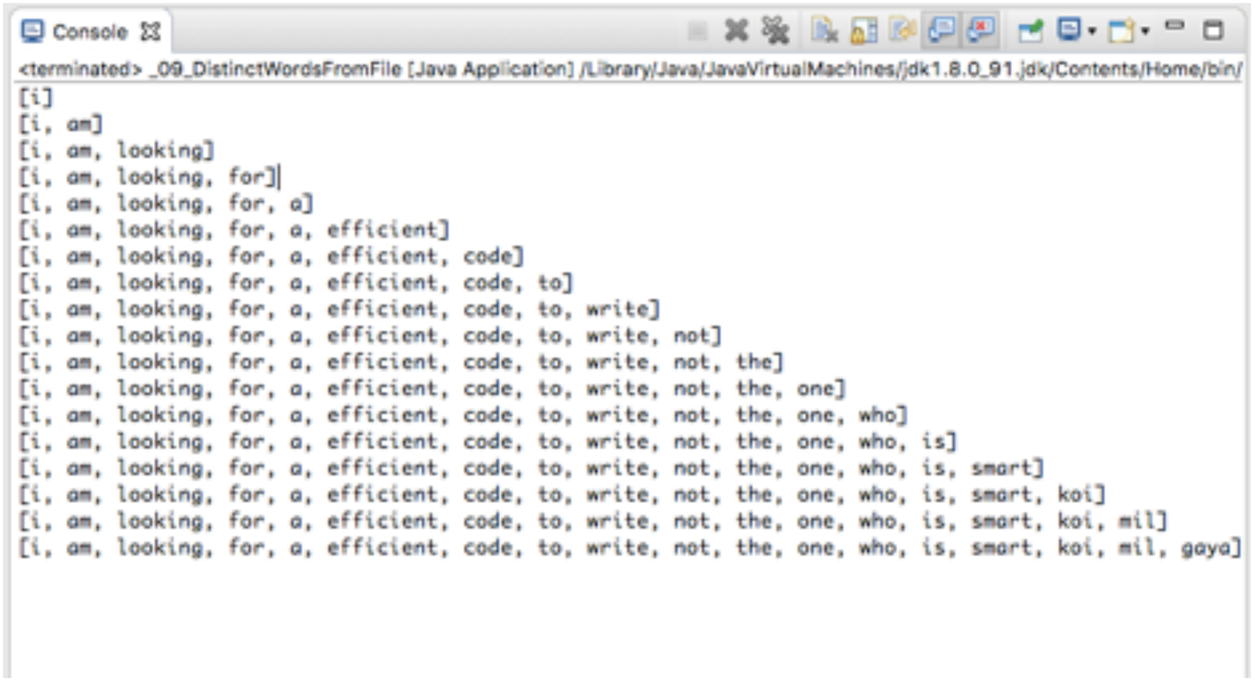
}

String tmp = st.nextToken().toLowerCase(); if(!wordList.-contains(tmp)){

wordList.add(tmp);

System.out.println(wordList); }}}

**************OUTPUT-9***********



```
<terminated> _09_DistinctWordsFromFile [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_91.jdk/Contents/Home/bin/
[i]
[i, am]
[i, am, looking]
[i, am, looking, for]
[i, am, looking, for, a]
[i, am, looking, for, a, efficient]
[i, am, looking, for, a, efficient, code]
[i, am, looking, for, a, efficient, code, to]
[i, am, looking, for, a, efficient, code, to, write]
[i, am, looking, for, a, efficient, code, to, write, not]
[i, am, looking, for, a, efficient, code, to, write, not, the]
[i, am, looking, for, a, efficient, code, to, write, not, the, one]
[i, am, looking, for, a, efficient, code, to, write, not, the, one, who]
[i, am, looking, for, a, efficient, code, to, write, not, the, one, who, is]
[i, am, looking, for, a, efficient, code, to, write, not, the, one, who, is, smart]
[i, am, looking, for, a, efficient, code, to, write, not, the, one, who, is, smart, koi]
[i, am, looking, for, a, efficient, code, to, write, not, the, one, who, is, smart, koi, mil]
[i, am, looking, for, a, efficient, code, to, write, not, the, one, who, is, smart, koi, mil, gaya]
```

10.Find longest substring without repeating characters.

```
public class _10_LongestSubstringNoDupChar {

    public static void main(String[] args) {

    System.out.println("Length of longest substring in 'javaprogramming' is: "+lengthOf-
LongestSubstring("javaprogramming"));

    }

    public static int lengthOfLongestSubstring(String s){

        if (s.length()==0) return 0;

        HashMap<Character, Integer> map = new HashMap<Character, Integer>();

        int max=0;

        for (int i=0, j=0; i<s.length(); ++i){

            if (map.containsKey(s.charAt(i))){

                j = Math.max(j,map.get(s.charAt(i))+1);

            }

            map.put(s.charAt(i),i);

            max = Math.max(max,i-j+1);

        }

        return max;
}}
```

*****************OUTPUT***************

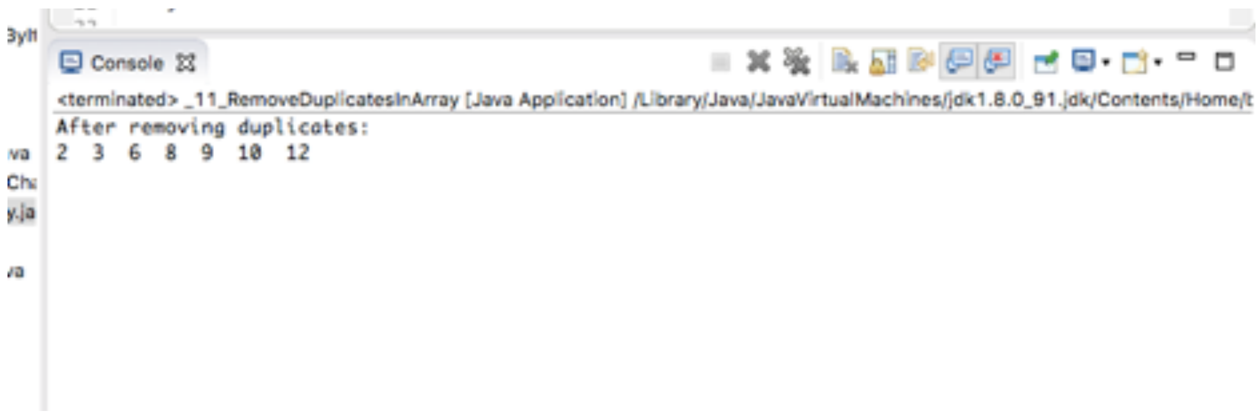
```
Length of longest substring in 'javaprogramming' is: 6
```

11.Write a program to remove duplicates from sorted array

```
public class _11_RemoveDuplicatesInArray { public static
void main(String[] args) {

int[] numArray = {2,3,6,6,8,9,10,10,10,12,12};
ArrayList<Integer> list = new ArrayList<Integer>(); for(int i
=0; i< numArray.length; i++){

if(!list.contains(numArray[i])){ System.out.println(numAr-
ray[i]); list.add(numArray[i]);

} }
```

*****************OUTPUT-11**************

```
After removing duplicates:
2  3  6  8  9  10  12
```



```
<terminated> _11_RemoveDuplicatesInArray [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_91.jdk/Contents/Home/b
After removing duplicates:
2  3  6  8  9  10  12
```

12.Write a program to print fibonacci series.

public class _12_FibonacciSeries { public static void main(String[] args) {

} }

System.out.println("Please enter number for Fibonacci se-ries: "); Scanner in = new Scanner(System.in);
int fibCount = in.nextInt();
int[] fibArray = new int[fibCount];

fibArray[0] = 0;
fibArray[1] = 1;
for(int i =2; i < fibCount; i++){

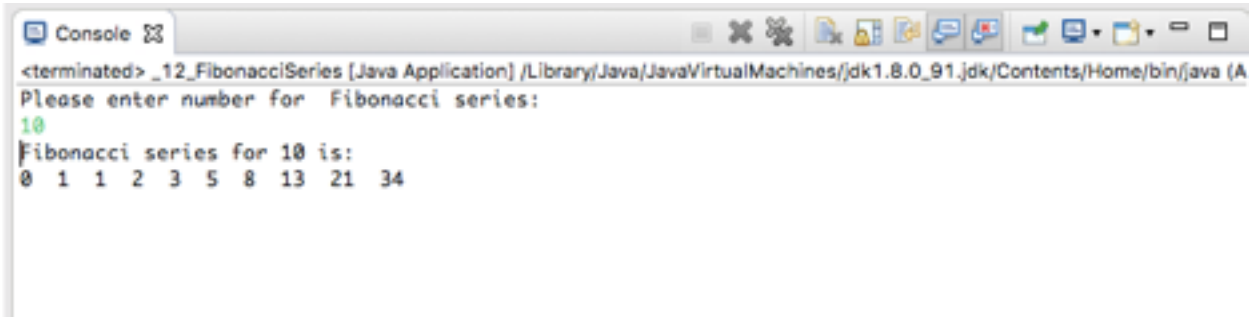
fibArray[i] = fibArray[i-1] + fibArray[i-2]; }

System.out.println("Fibonacci series for " +fibCount+ " is: "); for(int i = 0; i < fibCount; i++){

System.out.print(fibArray[i]+ " "); }

**************************OUTPUT-12******************

```
Please enter number for  Fibonacci series:
5

Fibonacci series for 5 is:
01123
Please enter number for Fibonacci series: 10
Fibonacci series for 10 is:
0 1 1 2 3 5 8 13 21 34
```



```
<terminated> _12_FibonacciSeries [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_91.jdk/Contents/Home/bin/java (A
Please enter number for  Fibonacci series:
10
Fibonacci series for 10 is:
0  1  1  2  3  5  8  13  21  34
```

13.Write a program to find out duplicate characters in a string

public class _13_DuplicateCharInString { public static void main(String[] args) {

String str = "Java Programing";
char[] cArray = str.toCharArray();
HashMap<Character , Integer> map = new HashMap<Char- acter , In-

for(char ch : cArray){ if(map.containsKey(ch)){

map.put(ch, map.get(ch)+1); }

else{
map.put(ch, 1);

} }

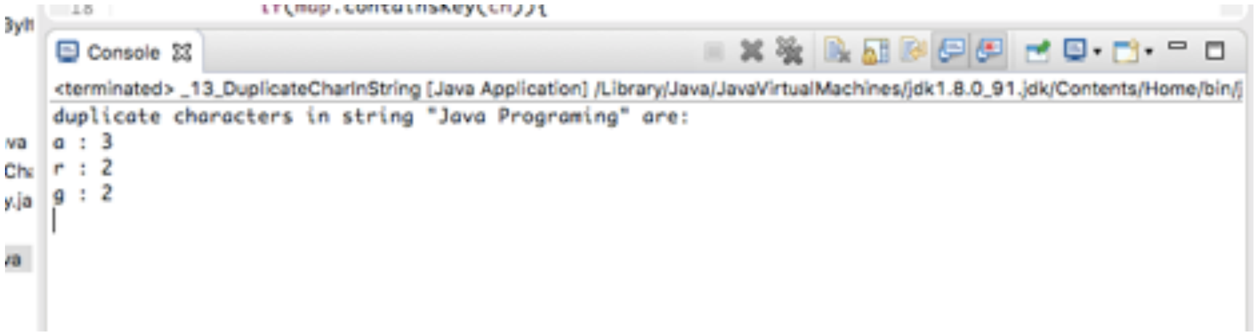
System.out.println("duplicate characters in string \""+str+ "\" are: ") ; Set<Character> keys = map.keySet();
for(Character ch : keys){

if(map.get(ch) > 1){
System.out.println(ch + " : "+map.get(ch));

}}}}

********************OUTPUT-13*********************

```
duplicate characters in string "Java Programing"
are: a:3
r:2
g:2
```



```
<terminated> _13_DuplicateCharInString [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_91.jdk/Contents/Home/bin/
duplicate characters in string "Java Programing" are:
a : 3
r : 2
g : 2
```

14.Write a program to create deadlock between two threads

```
public class _14_CreateDeadLock {
String str1 = "Java";
String str2 = "UNIX";
Thread trd1 = new Thread("My Thread 1"){

public void run(){ while(true){

synchronized(str1){
System.out.println("Thread1 having "+str1); try {
```

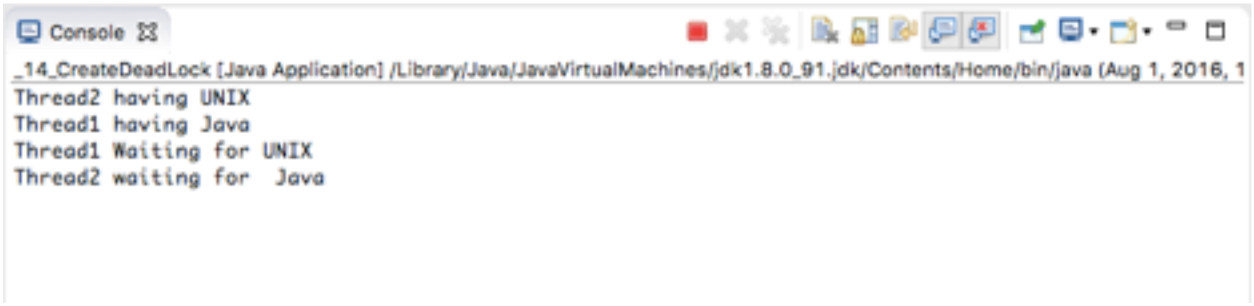
```
Thread.sleep(10);
} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace(); }

System.out.println("Thread1 Waiting for "+str2); synchro-
nized(str2){

System.out.println("Got Both "+str1 + str2);}}}}};

Thread trd2 = new Thread("My Thread 2"){ public void
run(){

while(true){ synchronized(str2){

System.out.println("Thread2 having "+str2); try {

Thread.sleep(10);

} catch (InterruptedException e) {
// TODO Auto-generated catch block e.printStackTrace();

}
System.out.println("Thread2 waiting for "+str1);

synchronized(str1){ System.out.println("Got Both "+str2 +
str1); }}}}};

public static void main(String[] args) {
_14_CreateDeadLock myDeadLock = new _14_Create-
DeadLock();

myDeadLock.trd1.start();
```

myDeadLock.trd2.start(); }

}

*******************OUTPUT-14**************

```
Thread1 having Java
Thread2 having UNIX
Thread2 waiting for  Java
Thread1 Waiting for UNIX
```



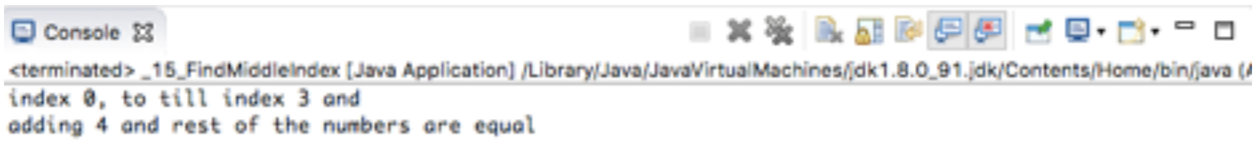
```
_14_CreateDeadLock [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_91.jdk/Contents/Home/bin/java (Aug 1, 2016, 1
Thread2 having UNIX
Thread1 having Java
Thread1 Waiting for UNIX
Thread2 waiting for  Java
```

15.Find out middle index where sum of both ends are equal

```
public class _15_FindMiddleIndex {
public static int findMiddleIndex(int[] numbers) throws Ex-
ception {

int end = numbers.length -1; int start = 0;
int sumRight = 0;
int sumLeft = 0; while(true){

if(sumLeft > sumRight){
sumRight = sumRight + numbers[end]; end--;

} else{
```

```
start++; }

if(start > end){
if(sumRight == sumLeft){

break; }

else{ throw new Exception(

"Could not find middle index where sum is equal at both
sides"); }}}

return end; }

sumLeft = sumLeft + numbers[start];

public static void main(String a[]) { int[] num = { 2, 4, 4,2,2,
5, 4, 1 }; try {

int middleIndex = findMiddleIndex(num); System.out.print-
ln("index 0, to till index "

+ middleIndex + " and");

System.out.println("adding " + (middleIndex+1) +" and rest
of the numbers are equal");

} catch (Exception ex)
{ System.out.println(ex.getMessage());

} }

}
```

************************OUTPUT-15*******************


Console
<terminated> _15_FindMiddleIndex [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_91.jdk/Contents/Home/bin/java

```
index 0, to till index 3 and
adding 4 and rest of the numbers are equal
```

```
index 0, to till index 3 and
adding 4 and rest of the numbers are equal
```

16.Write a program to find the given number is Arm-strong number or not?

public class _16_IsArmstronNumber { public static boolean isArmstrong(int num){

int number = num;
int mod;
int digitOfNum = String.valueOf(num).length(); int sum = 0; while(num != 0){

mod = num%10;
sum = sum + (int)Math.pow(mod, digitOfNum); num = num/10;

}
if(sum == number){

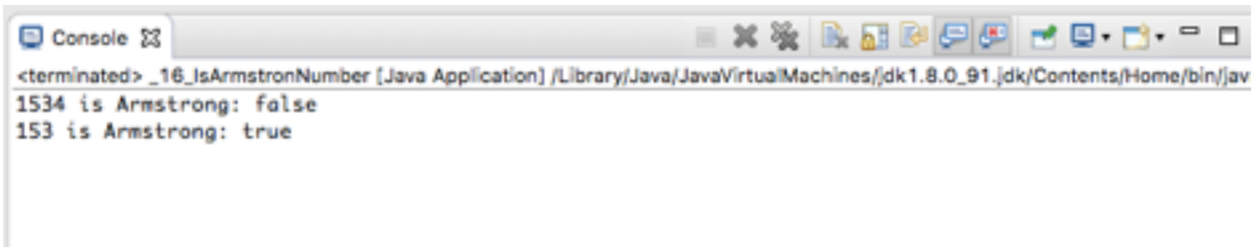
return true; }

else{
return false;

```
} }
public static void main(String[] args)
{ System.out.println("1534 is Armstrong:
"+isArmstrong(1534)); System.out.println("153 is Arm-
strong: "+isArmstrong(153)); }}
```

*******************OUTPUT-16*****************

```
1534 is Armstrong: false
153 is Armstrong: true
```



```
<terminated> _16_IsArmstronNumber [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_91.jdk/Contents/Home/bin/jav
1534 is Armstrong: false
153 is Armstrong: true
```