Sample Paper

1. Find Output

An Employee Management System application is used to maintain information about employees in an organization. In the application, employee details are stored in the ascending order of the employee Ids. Which algorithmic design technique would best fit if an employee needs to be searched based on the employee ld.

A. Greedy Approach

B. Brute Force

C. Divide and Conquer

D. Dynamic Programming

Output

Answer: A

Explanation:

Greedy algorithms are very fast. A lot faster than the two other alternatives (Divide & Conquer, and Dynamic Programming). They're used because they're fast. Most of the popular algorithms using Greedy have shown that Greedy gives the global optimal solution every time.

2. Find Output

```
public class Tester {
    Public static void main (String[ ] args) {
        for(int loop = 0;loop < 5;loop++) {
            if(loop > 2) {
                continue;
            }
            if(loop>4) {
                break;
            }
            System.out.println(loop) ;
        }
    }
}
```

Output

0

1

2

3. Find Output

Which of the following statements is TRUE with respect to Java language being platform independent?

A. The code in the java file is platform dependent

B. The JVM is the same across all operating systems

C. A java program written in a machine with Windows operating system cannot be executed on a machine having Linux operating system though Java is installed accordingly

D. A .class file can be run in any operating system where Java is installed

Output

Answer: D

4. Find Output

```
class Demo{
  public static int specialAdd(int num1) {
        if (num1!=0)
            return (num1+2)+specialAdd(num1-1) ;
        else
            return 3;
  }
  public static int extraordinaryAdd(int num2) {
         if (num2!=0)
             return specialAdd(num2)+extraordinaryAdd(num2-1) ;
          else
             return 0;
  }
   public static void main (String [ ] args) {
            System.out.println( (extraordinaryAdd(5) ) ) ;
  }
}
```

Output

A. 80

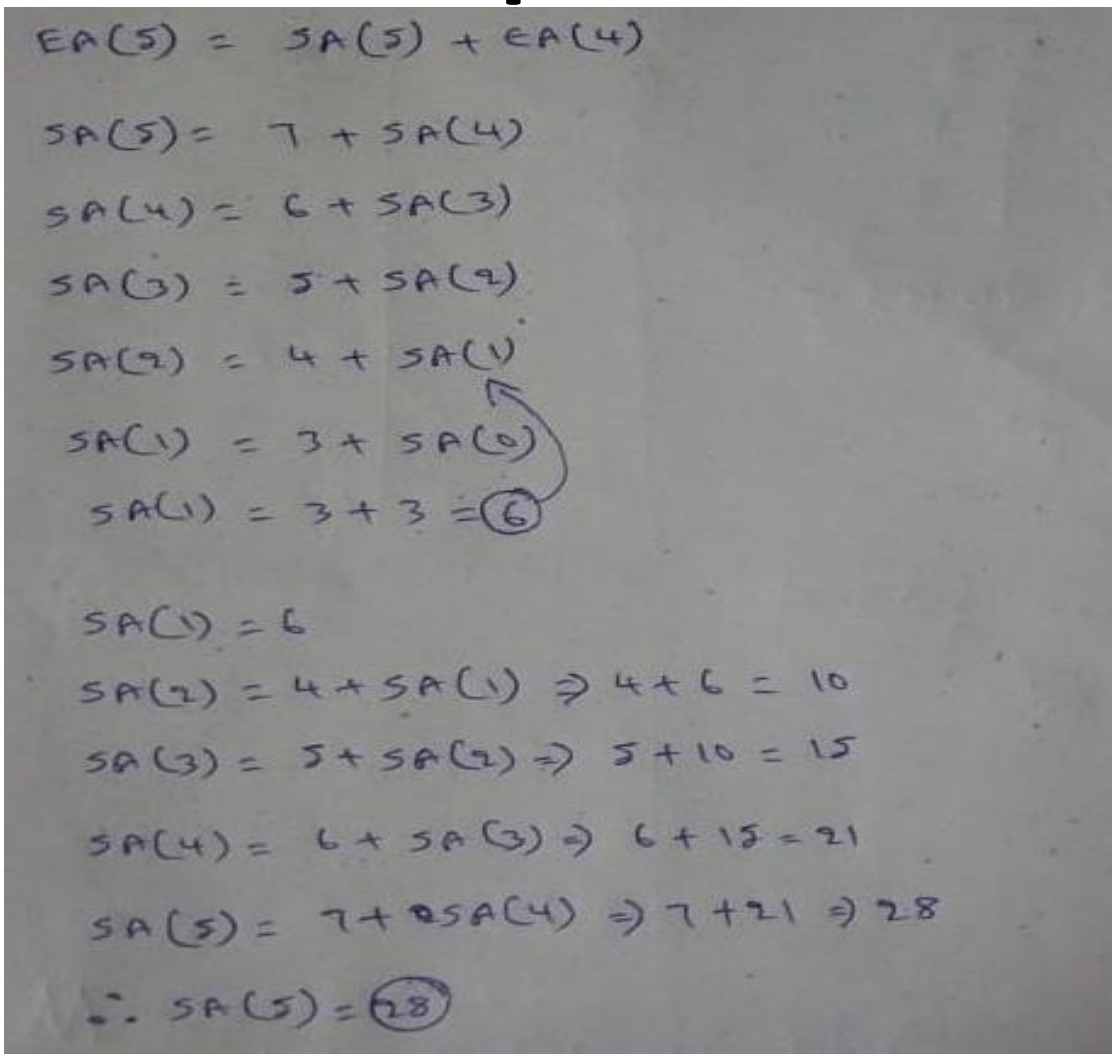
B. 52

C.70

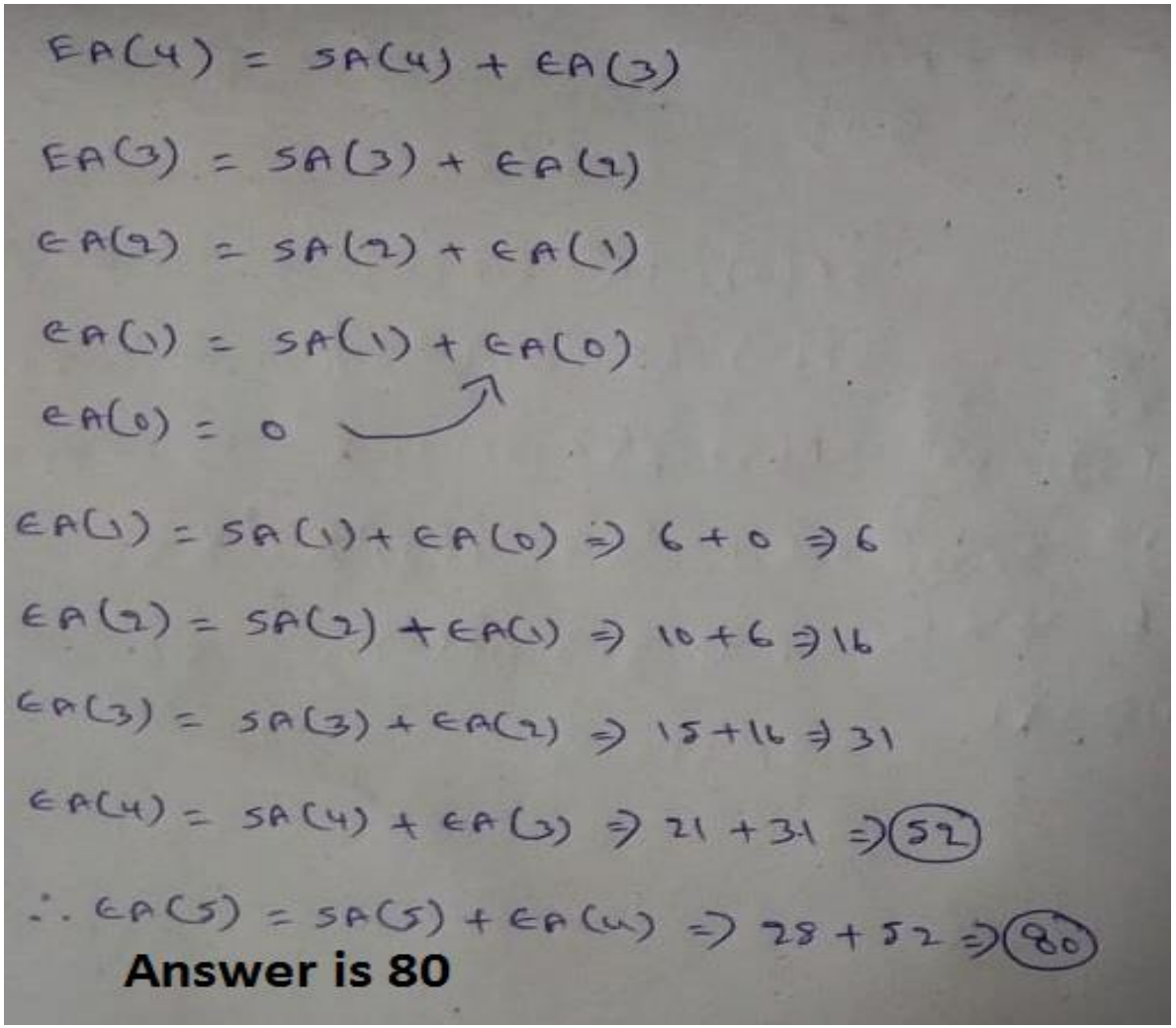
D. 25

Answer: A

Explanation



Explanation



5. Find Output

```
class Computation {
 public int add(int num1, int num2) {
        return num1 + num2 ;
 }
 public int divide(int num1, int num2) {
        return num1 / num2 ;
 }
}
public class TestComputation {
  Computation comput = new Computation ( ) ;
  @Test
   public void testAdd1 ( ) {
        int expected = 5 ;
           int actual = comput.add(2, 3) ;
        Assert.assertEquals(expected, actual) ;
 }
 @Test
  public void testAdd2 ( ) {
         int expected = 7 ;
         int actual = comput.add(2, 5) ;
         Assert.assertEquals(expected , actual) ;
 }
}
```

Output

A. Both testAdd1 and testAdd2 fail

B. testAdd1 fails and testAdd2 passes

C. Both testAdd1 and testAdd2 pass

D. testAdd1 passes and testAdd2 fails

Answer: C

Explanation:

Both would pass as expected and the actual values match in both

6. Find Output

```
class Customer {
   public int custId ;
   public String custName ;
}
public class Tester {
    public static void main (String args{ } ) {
            Customer obj = new Customer ( ) ;
            Customer objOne = new Customer ( ) ;
            Customer objTwo ;
             Customer objThree = obj ;
    }
}
```

Output

A. 3 objects and 1 reference variable

B. 2 objects and 4 reference variables

C. 4 objects and 4 reference variables

D. 2 objects and 3 reference variables

Answer: B

Explanation:

We know that an object is created by using a new keyword in java, a new keyword is used 2 times in the above code and hence 2 objects are created.

There are 4 references created in the above code namely→ ***obj, objOne, objTwo, objThree***

7. Find Output

```
class Student {
  private int studentId ;
  private String studentName ;
  Student (int studentId,String studentName) {
          this.studentId = studentId ;
           this.studentName = studentName ;
  }
}
class College {
  private Student student ;
  private int basicFees ;
  College (Student studentId, int basicFees) {
          this.student = student ;
           this.basicFees = basicFees ;
  }
}
```

Output

Identify the relationship between Student and College classes.

A. Aggregation

B. Association

C. Inheritance

D. The two classes are not related

Answer: A

8. Find Output

```
public class ExceptionExample {
  public void checkForExceptions(int num1, int num2) {
        int intArr [ ] = {1,2,3} ;
        String str = null ;
        System.out.println("Before any exception!") ;
        try{
             str.charAt(0) ;
             System.out.println(num1 / num1) ;
             System.out.println("Enjoy no exception!") ;
        }
        catch (ArithmeticException e) {
                 System.out.println("ArithmeticException handler!") ;
         } catch (NullPointerException e) {
                  System.out.println("NullPointException handler!") ;
         } catch (Exception e) {

                 System.out.println("Default exception handler!") ;
       } finally {
              System.out.println("In finally!");
       }
       System.out.println("After handling exception!") ;
  }
```

8th question continuation

```
public static void main(String [ ] args)
  {
        ExceptionExample exceptionExample = new
ExceptionExample( ) ;
        try {
                exceptionExample.checkForExceptions(2, 0) ;
         } catch (ArithmeticException e) {
                 System.out.println("ArithmeticException handler
in main!") ;
         }
         System.out.println("End of main") ;
   }
}
```

Output

Before any exception!

NullPointerException handler!

In finally!

After handling exception!

End of main

9. Find Output

Consider the problem size as ‘n’. Find the worst-case time complexity of the following algorithm.

```
if num1>num2 then
  for (couter1=1;counter1<=n;counter1=counter1*2)
 print(“num1 is greater than num2”)
  end-for
else
   for(counter2=1;counter2<=n;counter2=counter2+1) {
 print(“num2 is greater than num1”)
   end-for
end-if
```

Output

A. O(n)
B. O(n2)
C. O(log n)
D. O(n log n)

Answer: D

Explanation:

Time complexity of ***for (couter1=1;counter1<=n;counter1=counter1*2)*** would be **O(log n)**

Time complexity of ***for(counter2=1;counter2<=n;counter2=counter2+1)*** would be **O(n)**

So the time complexity would be O(n log n)

10. Find Output

Consider the code given below which is written in the file 'Demo.java'.

```
class Book{
  / /Class definition
}
class Demo{
   public static void main(String [ ] args) {
   }
}
```

Output

How many .class files will be generated for the above code and which class out of the two, Demo or Book, will be loaded into the main memory first when executed?

A. 2, Demo

B. 2, Book

C. 1, Demo

D. 1, Book

Answer: A

11. Find Output

Consider the Binary Search code given below:

```
public static int search(int arrayOfElements [ ], int low, int high, int elementToBeSearched) {
    if (low <= high) {
        int mid = (low + high) / 2 ;
        if (arrayOfElements[mid] == elementToBeSearched)
            return mid;
        if (arrayOfElements[mid] < elementToBeSearched)
            return seach(arrayOfElements, mid + 1, high, elementToBeSearched) ;
        return search (arrayOfElements, low, mid -1, elementToBeSearched);
    }
    return -1;
}
```

Output

Consider the arrayOfElements having 6 elements with low as 0 and high as 5. The elements of the array are as follows.

5 6 9 12 15 29

Find the number of iterations when using binary search if the elementToBeSearched is 6?

A. 1

B. 2

C. 3

D. 4

Answer: C

Explanation:

1st iteration: low= 0
high= 5
mid= (0+5)/2 = 2
a[2]>key, which means 9>6

2nd iteration: low = 0
high = mid-1 = 1
mid = (0+1)/2 = 0
a[0]<key, which means 5<6

3rd iteration: low = mid+1 = 0+1 = 1
high = 1
mid = (1+1)/2 = 1
a[1]==key, which means 6==6.

12. Find Output

```
class Item{
   public String itemId;
   String itemName ;
   protected float itemPrice;
   private int itemDiscount ;
   public Item(String itemId,String itemName)
         this.itemId=this.itemId;
         this.itemName=itemName;
   }
}
```

Output

Identify the access specifier of the data member ‘itemName’.

A. public
B. protected
C. private
D. default

Answer: D

13. Find Output

```
public class Question {
   public static void main (String [ ] args) {
         int var = 22, anotherVar = 7, result ;
         String str = “One” ;
         String anotherStr = “Two” ;
         result = var*anotherVar / anotherVar ;
         if ( result < 22 ) {
             System.out.println(str) ;
         }
         else {
                 System.out.println(anotherStr) ;
         }
    }
}
```

Output

A. Compilation error: incorrect use of operators

B. One

C. No output is displayed

D. Two

Answer: D

14. Find Output

```
class Bill {
   int itemPrice
   public Bill (int itemPrice) {
         this.itemPrice = itemPrice ;
   }
   void display ( ) {
         int itemPrice = 20 ;
         System.out.println (itemPrice) ;
    }
}
class Demo {
    public static void main(String [ ] args) {
         Bill billobj = new Bill (10) ;
            System.out.println(billobj.itemPrice) ;
         billobj.display ( ) ;
    }
}
```

Output

10

20

15. Find Output

```
public class Tester {
   public static void main (String [ ] args) {
        int [ ] tempList = { 1, -1, -2 } ;
        int [ ] numList = {-2, -1, 1 } ;
        int length = numList.length ;
        for (int value : tempList) {
            int tempValue = value ;
            if (value<0) {
                 tempValue = length - Maths.abs(value) ;
            }
            if(value == tempList [tempValue]) {
                if(value<0) {
                     numList [length-tempValue]=value ;

                   }
                else {
                     numList [tempValue]=value ;
                 }
             }
             else {
                  numList [0] = value ;
              }
         }
    }
}
```

Output

What will be the elements of numList after the execution of the above code?

A. [-2,-1,1]
B. [-2,-2,1]
C. [1,-2,1]
D. [-2,-1,2]

Answer: A

16. Find Output

```
class ListExample
{
   public static void main ( String [ ] args)
   {
      List<String> list = new ArrayList<>( ) ;
      list.add (“I”) ;
      list.add (“Love”) ;
      list.add(“Java”) ;
      list.add(“Language”) ;
      Iterator<Object> iter = list.iterator ( ) ;
      while (iter.hasNext ( ) )
           System.out.print ( iter.next ( ) .toString ( ) + “ “) ;
       System.out.println ( ) ;
    }
}
```

Output

Assumption: All classes, interfaces, and necessary methods are available

A. I Love Java Language

B. Error: Incompatible types: String cannot be converted to Object

C. Error: Iterator cannot be created for Object

D. Error: toString() cannot be applied on a String object

Answer: B

17. Find Output

```
class Base {
  private int fun ( ) {
      return 0;
  }
  public int run ( ) {
      return 3;
  }
}
class Derived extends Base {
  private int fun ( ) {
      return 1 ;
  }
  public int run ( ) {
      return fun ( ) ;
  }
}
class Derived1 extends Derived {
  public int fun ( ) {
      return 2 ;
  }
}
class Test {
  public static void main ( String [ ] args) {
       Base baseRef = new Derived1 ( ) ;
       System.out.println(baseRef.run ( ) );   }  }
```

Output

A. 1

B. 2

C. 0

D. 3

Answer: A

Find Output

Output