Exception (try-catch)

```
import java.io.*;
class kapil
{       public static void main(String args[]) throws Exception
          {  String a="ram",c;char t;a=a.substring(0,3);  }
}
```

The above program does not create any problem. Similarly no problem is caused if a=a.substring(3); or t=a.charAt(2) is used. The trouble is caused by followings t=a.charAt(3); or a=a.substring(0,4); or a=a.substring(4); [Important: t=a.charAt(3); causes trouble but substring(0,3) or substring(3) does not causes trouble.]

```
{ String a="",c;int b,d,t,f[]={12,34,56,78};
  DataInputStream o=new DataInputStream(System.in); a=o.readLine( );
  try{ c=a.substring(2,4); b=Integer.parseInt(a); t=20/(b-21792); d=f[b-40000]; }
  /* catch (StringIndexOutOfBoundsException e){System.out.println("ram");}
     catch (ArrayIndexOutOfBoundsException e){System.out.println("hari");}
     catch (NumberFormatException e){System.out.println("gopal");}
     catch (ArithmeticException e){System.out.println("kapil");} */
     catch (Exception e){System.out.println(e);}
}
```

Observer the output of the program on input tom, shyam, 21792, 40008.
tom: StringIndexOutOfBoundsException   shyam: NumberFormatException
21792: ArithmeticException / by zero   40008: ArrayIndexOutOfBoundsException
Also observe the output by removing comments.

```
k=Integer.parseInt(a);k=k%2;
try{  p="ram";char y=p.charAt(k+2);       Here p=p.substring(k+3); can also be used.
      System.out.println(a+" is an even number");
}catch(Exception m){System.out.println(a+" is an odd number");}
```

Program reads a number and finds whether it is even or odd.

In following problems use of "if" is not permitted.

Letters are printed in seperate line. Numbers are (+) ve int Same letter is not printed by loop

1. Read a string. If it is a number then output its double otherwise output 0. Input 12 output 24 . Input 3 output 6 . Input "12 14" output 0. Input ram output 0.
2. Read a string. If it is a number then output "AC". Otherwise output "BC". You should use only 3 print statements. Every print statement should print only one letter(not Variable)
3. Extend above: If non-number BC, if number less than 5 ADC, otherwise ABC. [Example: ram→BC, 3→ADC, 7→ABC] [use only one try, one catch, 4 prints]
4. Extend above: outputs BC, ADC and AEC respectively. [one try, two catches, 5 prints]
5. Read a string. If number AC, if non-number of size less than 5 BC, otherwise DC. [34→AC, ram→BC, mohan→DC] [two trys, two catchs. 4 prints (one letter each)]
6. Read a string. If it is non number output BC. If it is a number greater than 5 output A. Otherwise AC. [ram→BC, 8→A..., 4→AC][1-try 1-catch(number format)] 3 print
7. Non-number BC. Number ABC. [1-try 1-catch (number format only)] 3 print
8. Non-number BC. Number>5 D(garbage). Number≤5 DEC (1-try 1-catche ; 4 print
9. Non-number AC. x<5 BAC. x>10 BDAC. Otherwise BDEC. (1-try 1-Catch) 5 print
10. Non-number BC. x>10PC, x<5 QC, x=5,6,7RC, x=8,9SC x=10S(..) (1try 4 catches) 6 print
11. Read a string. If its length is less than 5 then output "ram" otherwise "hari".
12. If length is less than 5 then "ram". If more than 9 "hari". Otherwise "dipu". [nested try]
13. Do above problem using only one try (two catches) (no if).[Hint: catch ArithmeticException and StringIndexOutOfBoundsException separately].
14. Read a number. If it an integer then output its double. If it is a float then output it's triple. Do not use indexOf or if. Input 12 output 24. Input 2.72 output 8.16.
15. Read an integer. If it is more than 5 then output "ram". Otherwise "hari" [no if]
16. Read an integer. If more than 5 then "ram". If less then "hari". If equal to then "kapil"
17. Do above using only one try (two catches) (no if)
18. Read a string. If its length is less than 5 then output pal. More than 9 output ramesh. Otherwise output rampal. Use only one try and one catch. [Hint: System.out.print]
19. Read a number. Output ram if it is more than 5 otherwise output hari. Use only NumberFormatException