Exception (finally)

```
class kapil
{ public static void main(String args[]) throws Exception
  { try{ System.out.println("ram"); int a=25/0; System.out.println("hari");}
    catch(Exception t) {System.out.println("gopal");}
    System.out.println("sani");
  }
}
```

The output of above program is ram gopal sani. When catch(Exception t) is replaced by finally then output is ram gopal. When a=25/0 is replaced by a=25/7 then output in these cases is ram hari sani and ram hari gopal sani respectively.

in try or finally

In each of the following programs least number of print statements should be used. Every print statement should print only one variable (string, int, float) or one letter. [You may use System.out.print in place of System.out.println to continue printing in same line]. Do not use catch unless permitted. Do not use if. Any print statement should not print string by joining two or more strings. ~~Type conversion (int a;String b; b=a+"";) is permitted.~~ NO (+) Should be used

1. Read a string. If it is a number output its square. Otherwise output 100 (i.e. 10^2). 1A. Use * in only print
2. Read a string. If it is a number output its square. Otherwise output "ram". 2A. 1 Print, + permitted
3. Read a string. If non-number then print it as it is. If it is a number then print its double, same number, square. [Example: ram → ram, 25 → 50 25 625] [3 prints] [hari → hari]
4. Modify above. In case of non number print toy. [ram → toy, 25 → 50 25 625][3 prints]
5. Modify above. In case of number do not print the same number. [Example: ram → toy, 25 → 50 625][2 prints] [Hint: type conversion] [int p; String q; q=p+"";] permitted
6. Modify above. In case of number print same number, double, square. [3 prints] q= Integer.to String(p)
7. Read a string. It has one or two numbers. Print squares. [2 prints] [5 → 25 2 5 → 4 25]
8. Do above problem without using any print in try. Use 2 prints. 8(A). No print after finally.
9. Read a string. It has 1, 2 or 3 numbers. Print squares. [3 prints][Order in output may be changed. Input 12 6 10 output 100 144 36 is permitted.]
10. Do above problem so that correct order is maintained.
11. Do above three problems using one catch (no finally) [2/3 prints][0/1 print in try]
12. Read a string. If it is a number then output "ABC". Otherwise output "B". You should use only 3 print statements. Every print statement should print only one letter (no string variable should be used in printing). Output may appear in seperate line.
13. Extend above: If non-number B, if number less than 5 ADBC, otherwise AB. [Example: ram → B, 3 → ADBC, 7 → AB] [use only one try, 4 prints]
14. Read a string. If its length is less than 5 then output "ABC". Otherwise output "B". You should use only 3 print statements. Every print statement should print only one letter. [Hint: k=a.length(); String k="gopal" t=b.chatAt(k);]
15. Read a string. If it is a number output AB otherwise CA. [one try, one catch, finally] 3 Print
16. Read a string. If it is a number output BA otherwise CA. [one try, one catch, finally] 3 Print
17. Modify above. Non number CAB, Number<5DA, Number ≥ 5 DAB, [1T, 1C,F] Assume non-negative number 4 prints
18. Read a number(x). If x=0,1,2,3,4 then output DHBAG. x=5,6,7 o/pEBAG. x=8,9 CBAG. x=10,11 FBAG. [1 try, 4 catches, 8 prints]
19. Extend above. x=2 DBA others unchanged. [use finally, exception in catch 1/(x-2)]
20. Extend above. In case of non number output CBA. x=6 EB [exceptions in finally]
21. Read a string. It has either 1 or 2 numbers. If it has one number find its square. If it has two numbers find product. [Use only one *]
22. Do above problem 1 number square, 2 numbers sum. [one print. *,+ only in print]
23. Read a string. It has either 1 or 2 numbers. If it has only one number find its square. If it has 2 numbers find their sum. [No catch, use finally and many multiplications] [+ permitted]
24. Read a string. If it is number output ram. Otherwise hari.
25. Read a string. If it is integer output kapil. If it is float output ravi. Otherwise mohan.
26. Read a number. Output bimu if it is less than 5. Output tom otherwise. (No if, no catch)
27. Read a number. Output bimu if it is either less than 5 or more than 9. Output tom if it is between 5 and 9. (No if, no catch)
28. Read a string of numbers. Find sum. No loop except while(1==1). (+) permitted

7A. No try. In Case of two numbers order may be reversed

7B. Retain order by using (+).