Given a mathematical equation that contains only numbers and +, -, *, /. Print the equation in reverse, such that the equation is reversed, but the numbers remain the same.

Input Format

The First line will be representing String s.

Constraints

1<=|S|<=10^5

Output Format

Return the reverse equation

Sample Input 0

```
20-3+5*2
```

Sample Output 0

```
2*5+3-20
```

```java
1  import java.io.*;
2  import java.util.*;
3
4  public class Solution {
5
6      public static void main(String[] args) {
7          Scanner sc = new Scanner(System.in);
8          String str = sc.nextLine();
9          Stack<String>  st = new Stack<>();
10         String temp="";
11         int i =0;
12         char ch ;
13         while(i<str.length()){
14             ch = str.charAt(i);
15             if(ch>= 42 && ch<=47){
16                 st.push(temp);
17                 temp="";
18                 st.push(ch+"");
19             }
20             else{
21                 temp+=ch;
22             }
23             i++;
24         }
25         st.push(temp);
26         String revStr="";
27
28         while(!st.isEmpty()){
29             revStr+=st.pop();
30         }
31         System.out.print(revStr);
32     }
33 }
```
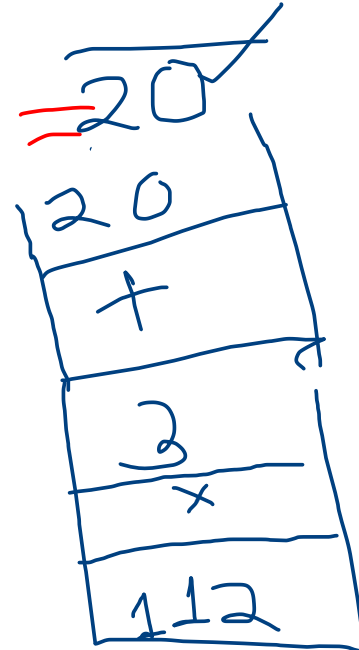
- Given Single Integer T. Next T lines will have a Single Integer N.
- **Print the Reverse** of the Integer N.
- **(Note: You have to use stack to perform this operation)**

Input Format

- Single Integer T.
- Next T lines will have Single *Integer* N.

Constraints

- $1 <= T <= 10^4$
- $0 <= N <= 10^8$

Output Format

- T Lines of **Reverse** of N Integers.

Sample Input 0

```
1234
1001
1000
2340
```

Sample Output 0

```
4321
1001
1
432
```

Explanation 0

- 1234 --> 4321
- 1001 --> 1001
- 1000 --> 0001 --> 1
- 2340 --> 432

```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int t = sc.nextInt();

        for(int i =0;i<t;i++){
            int n = sc.nextInt();
            System.out.println(revNumber(n));
        }
    }
    public static int revNumber(int n){
        Stack<Integer> st = new Stack<>();

        while(n>0){
            st.push(n%10);
            n/=10;
        }
        int revResult=0,multi=1;
        while(!st.isEmpty()){
            revResult+=st.pop()*multi;
            multi*=10;
        }
        return revResult;
    }
}
```