Given x and y, print xy

Problem | Submissions | Leaderboard | Discussions

Take x and y digits as integer inputs and then form a number xy from it and then finally print that number.

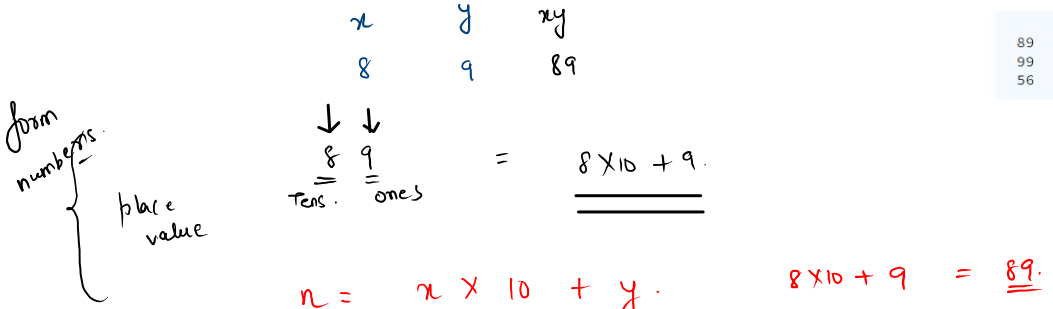
for eg. if you are the given **3** and **4**, then you have to form the number **34** from it and then finally print the number **34**.

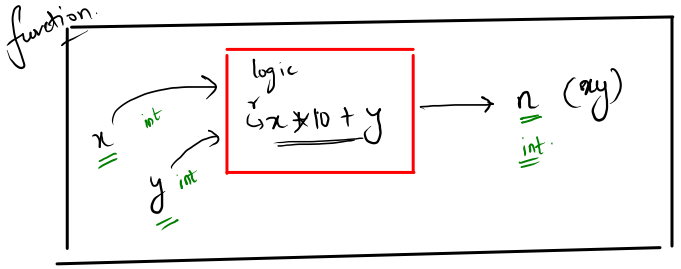
Sample Input 1

```
3
8
9
9
9
5
6
```

Sample Output 1

```
89
99
56
```





x	y	ans	
8	9	89	
9	9	99	... 9X10+9
5	6	56	... 5X10+6

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

public class Solution {
    public static int formNum(int x, int y){
        int res =  x * 10 + y;
        return res;
    }
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int T = scn.nextInt();
        for(int i = 0; i < T; i++){
            int x = scn.nextInt();
            int y = scn.nextInt();

            int ans = formNum(x, y);    //x * 10 + y
            System.out.println(ans);
        }

    }
}
```

T = 2

i=0 0<2 ✓

x = 6

y = 4

i=1 1<2 ✓

x = 5

y = 7

i=2 2<2 ✗

i/p

2
6
4
5
7

o/p

64
57

void function.

```java
public class Solution {
    public static void formNumber(int x, int y){
        int res =  x * 10 + y;
        System.out.println(res);
    }
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int T = scn.nextInt();
        for(int i = 0; i < T; i++){
            int x = scn.nextInt();
            int y = scn.nextInt();

            formNumber(x, y);    //x * 10 + y
        }

    }
}
```

1
5
6

T=1 i=0 0<1 ✓
x=5
y=6

res = 50+6 = 56

1 < 1 ✗

56

Print digit by digit of a three digit number

Problem | Submissions | Leaderboard | Discussions

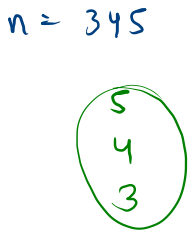
Given a three-digit positive number. Print its digits one by one starting from the digit at one's place to the digit at hundred's place in a separate line.

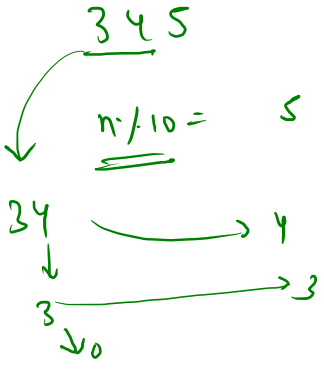
Sample Input 0

```
345
```

Sample Output 0

```
5
4
3
```





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

public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        while(n > 0){
            int d = n % 10;
            System.out.println(d);
            n /= 10;
        }
    }
}
```

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

public class Solution {
    public static void printDigits(int n){
        while(n > 0){
            int d = n % 10;
            System.out.println(d);
            n /= 10;
        }
    }


    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        printDigits(n);
    }
}
```

n = 345 → 34 → 3 → 0

345 > 0

d = 5

5

34 > 0

d = 4

4

3 > 0

d = 3

3

0 > 0 ✗

Reverse a 3 digit number

Problem | Submissions | Leaderboard | Discussions

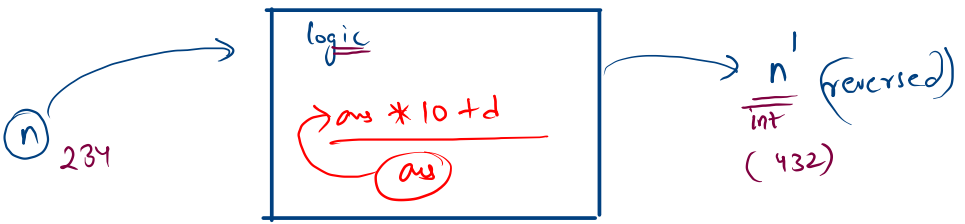
Take in a three digit number as an integer input and reverse its digits. And then print the final reversed number as an integer output.

Sample Input 0

```
3
234
123
456
```

Sample Output 0

```
432
321
654
```



$$432 = 4 \times 100 + 3 \times 10 + 2 \times 1$$

$$= 432$$

d1 = 2
d2 = 3
d3 = 4

ans = d3 × 100 + d2 × 10 + d1

n = 234

ans = ~~0~~ ~~4~~ ~~43~~ 432

d = 4

↳ ans = ans × 10 + d = 4

~~23~~

d = 3

ans = 4 × 10 + 3 = 43

2 d = 2

ans = 43 × 10 + 2 = 430 + 2 = 432

0

n = 562

265

$$[ans = (ans * 10 + d)]$$

2 X 100
6 X 10
+ 5
265

ans = ~~0~~ X ~~26~~ 265

n = ~~56~~2

d = 2

ans = 0 + 2

~~56~~ d = 6 ans = 2X10 + 6

5 d = 5 ans = 26 X 10 + 5 = 265

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

public class Solution {
    public static int reverse(int n){
        int ans = 0;
        while(n > 0){
            int d = n % 10;
            n /= 10;
            ans = ans * 10 + d;
        }
        return ans;
    }

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int t = scn.nextInt();
        for(int i = 1; i <= t; i++){
            int n = scn.nextInt();
            int ans = reverse(n);
            System.out.println(ans);
        }
    }
}
```

2
123
574

t=2

i=1 2

1<=2 ✓

2<=2 ✓

n=574

y/10 = 0

Print the final number xyzw...

Problem | Submissions | Leaderboard | Discussions

Take **n** as an integer input. Then take **n** digits as integer inputs and form a number from it and print that number as an integer output.

Sample Input 0

```
4
1
2
3
6
```

Sample Output 0

```
1236
```

try by yourself

x y z w

1 2 3 6

x1000 x100 x10 x1

= 1236

ans = 0 x 10 123 (1236)

d = x x x 6

{ ans = ans * 10 + d }

= 0 + 1 = 1

= 10 + 2

= 120 + 3

= 1230 + 6

Reverse n digit number

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

public class Solution {
    public static void reverse(int n){
        int ans = 0;
        while(n > 0){
            int d = n % 10;
            n /= 10;
            ans = ans * 10 + d;
        }
        System.out.println(ans);
    }

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int t = scn.nextInt();
        int n = 0;
        for(int i = 0; i < t; i++){
            int d = scn.nextInt();
            n = n*10 + d;
        }
        System.out.println(n);
        reverse(n);
    }
}
```

t=3

n=0 ~~2~~ ~~25~~ 256

0x10+2

2x10+5

25x10+6

i=0 0<3

d=2

1<3

d=5

2<3

d=6

3<3

3

Sample Input 0

3
2
5
6

Sample Output 0

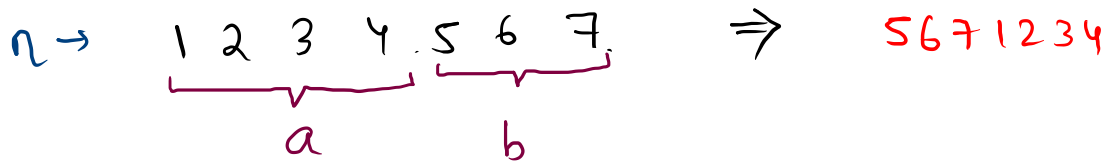
256
652

Rotate 7-digit number to right by three

Problem | Submissions | Leaderboard | Discussions

Take **n** as an integer input, you have to pick the last **3** digits of the number of and put them in the starting.

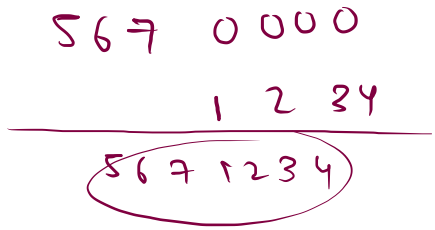
eg. `1234567` is given, then this number should transform to `5671234`.



ans = b X 10000 + a = ?

b = n % 1000

a = n / 1000



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

public class Solution {

    public static void rotate(int n){
        int a = n / 1000;
        int b = n % 1000;
        int res = b * 10000 + a;
        System.out.println(res);
    }

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int t = scn.nextInt();
        for(int i = 1; i <= t; i++){
            int n = scn.nextInt();
            rotate(n);
        }
    }
}
```

1

1234567

t=1

i=1

1<=1 ✓

n = 1234567

a = n/1000 = 1234

b = 567

1234
1000) 1234567
1000
2345
2000
3456
3000
4567
4000
567

res = 5670000
1234
(5671234)

(7)

n

rotate 3.

12 345

$a = n/1000 = 12$

$b = n \% 1000 = 345$

345 12

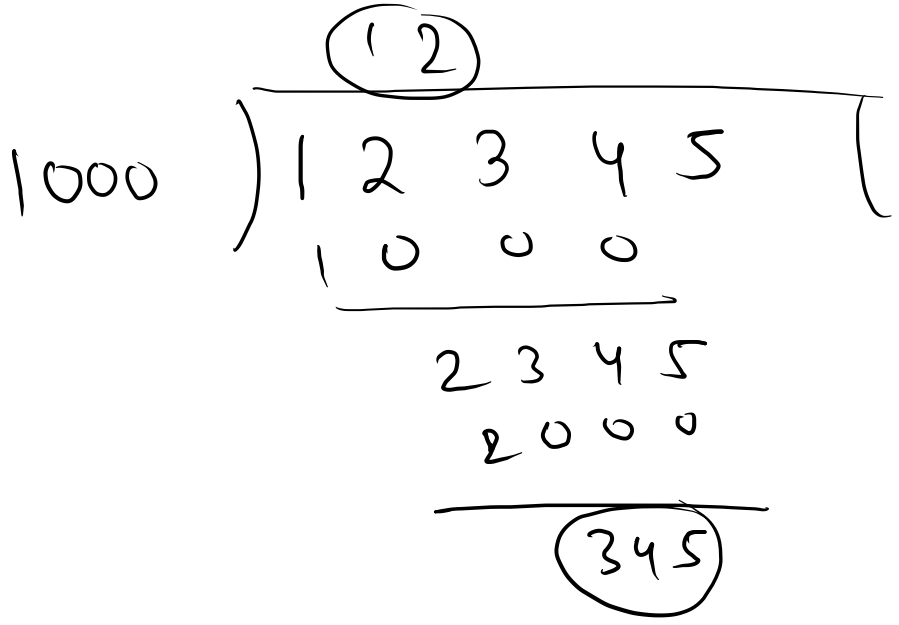
$b \times 10^{r-1} + a$

$= b \times 10^{2} + a$

345×100

$+ 12$

34512


12
1000
12345
1000
2345
2000
345

ntate=3.

r

1 2 3 4 5 6 7 8 9

n%1000

n/1000

n%10^r

n/10^r

Armstrong Number ⇒ sum of cube of each digit should be equal to number

n = 153

$1^3 + 5^3 + 3^3 = n$

$1 + 125 + 27 = 153$

eg. 234

$2^3 + 3^3 + 4^3 =$

$8 + 27 + 64 \neq 234$

1st.

153

$= 3*3*3 + 5*5*5 + 1*1*1$

ans $= 27 + 125 + 1$

n == ans → true.