**Repetitive Addition Of Digits**

[ad-hoc](http://www.practice.geeksforgeeks.org/tag-page.php?tag=ad-hoc&isCmp=0)[maths](http://www.practice.geeksforgeeks.org/tag-page.php?tag=maths&isCmp=0)

Given a non-negative integer num, repeatedly add all its digits until the result has only one digit.

**Input:**  
The first line contains 'T' denoting the number of testcases. Then follows description of testcases. The next T lines contains a single integer N denoting the value of N.

**Output:**  
Output the sum of all its digit until the result has only one digit.

**Constraints:**  
1<=T<=30  
1<=n<=10^9

**Example:**  
Input :  
2  
1  
98

Output :  
1  
8

Explanation:  For example, if we conisder 98, we get 9+8  = 17 after first addition. Then we get 1+7 = 8.  We stop at this point because we are left with one digit.

\*\*For More Examples Use Expected Output\*\*

<http://www.practice.geeksforgeeks.org/problem-page.php?pid=235>

#include <iostream>

#include <stdio.h>

using namespace std;

int convertir(int n)

{

struct Helper{

char buffer[100];

std::string to\_string(int k){

sprintf(buffer, "%d", k);

return std::string(buffer);

}

};

Helper h;

while (h.to\_string(n).length() > 1)

{

int sum = 0;

int copia = n;

while (copia > 0)

{

sum += (copia % 10);

copia /= 10;

}

n = sum;

}

return n;

}

int main() {

int t;

scanf("%d", &t);

while(t--) {

int N;

scanf("%d", &N);

int res = convertir(N);

printf("%d\n", res);

}

return 0;

}