	only if the k-th power of each digit sums to N.
iven a positive integer N, return true if and only if it	is an Armstrong number.
xample 1:	
nput:	
53	
•	44.
Output:	
rue	
xplanation:	
53 is a 3-digit number, and 153 = 1^3 + 5^3 + 3^3.	
xample 2:	
<i>Y</i>	
nput:	

Input.				
Input:				
123				
Output:			,	
false				
Explanation:				
123 is a 3-digit number, and 123 != 1^3 + 2^3 + 3^3 = 36.				
3 + 3 + 3 + 3 + 3 + 3 + 3 + 3 + 3 + 3 +			·, ,	
Example 3:		ACS.		
Input:		٠٠	*	
mpot.				
1634			1	
1034				
Output:				
			•	
true				
Note:			**************************************	
	•			
1 <= N <= 10^8				

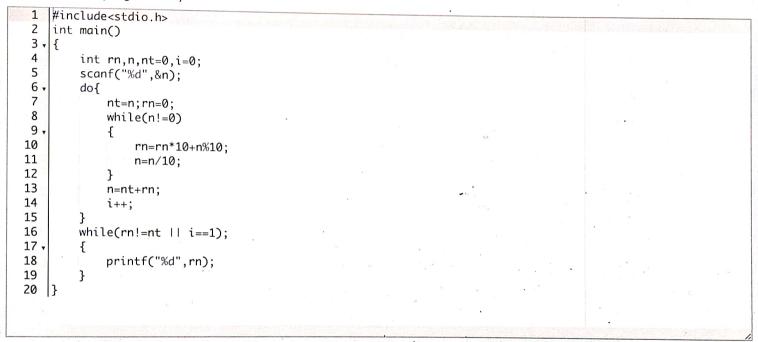
```
#include<math.h>
    int main()
 5
        int N,R=0,sum=0,rem;
        scanf("%d",&N);
 6
        int temp1=N, temp2=N;
8
        while(temp1!=0)
9 ,
10
            temp1/=10;
11
            R++;
12
13
        while(temp2!=0)
14 .
15
             rem=temp2%10;
16
             sum+=pow(rem,R);
17
             temp2/=10;
18
19
        if(sum==N)
20 .
21
             printf("true");
22
23
        else
24 .
25
             printf("false");
26
27
```

#include<stdio.h>

	Input	Expected	Got	
~	153	true	true	~
~	123	false	false	~

Passed all tests! <

Answer: (penalty regime: 0 %)



		Input	Expected	Got	
SPANISHED TO	~	32	55	55	~
<b>有种种的</b>	~	789	66066	66066	~

ucky number is 3, and 2nd lucky number is 4 and 3rd luc not lucky as they have other numbers in it.	,	an leaky Hember 13		15, 40 etc., are
The program should accept a number 'n' as input and dis	splay the nth lucky num	nber as output.		
Sample Input 1:		•		
3				
Sample Output 1:		43	* • ·	
33		. ·		
Explanation:	*1			
Here the lucky numbers are 3, 4, 33, 34., and the 3rd lu	ucky number is 33.			
Sample Input 2:				
34				
Sample Output 2:	•			

```
#include<stdio.h>
   int main()
3 → {
        int n=1, i=0, nt, co=0, e;
        scanf("%d",&e);
6
        while(i<e)
             nt=n;
             while(nt!=0)
10 .
11
                 co=0;
                 if(nt%10!=3 && nt%10!=4)
12
13 .
14
                     co=1;
15
                     break;
16
17
                 nt=nt/10;
18
19
             if(co==0)
20 +
21
                 i++;
22
23
             n++;
24
        printf("%d",--n);
25
        return 0;
26
27
```

	Input	Expected	Got	
~	34	33344	33344	~

Passed all tests! ✓