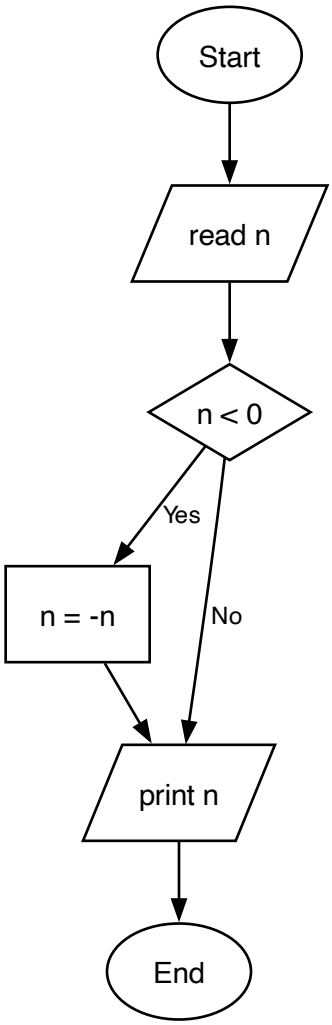
Problem 1.



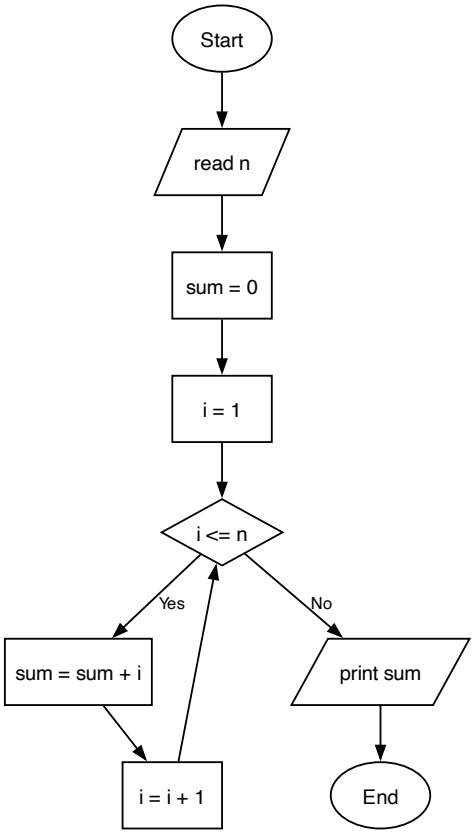
(i) For each input below, follow the flowchart and write the output.

input n	output
7	
-4	
0	
-19	

(ii) Can you tell what this algorithm is doing?

Problem 2.

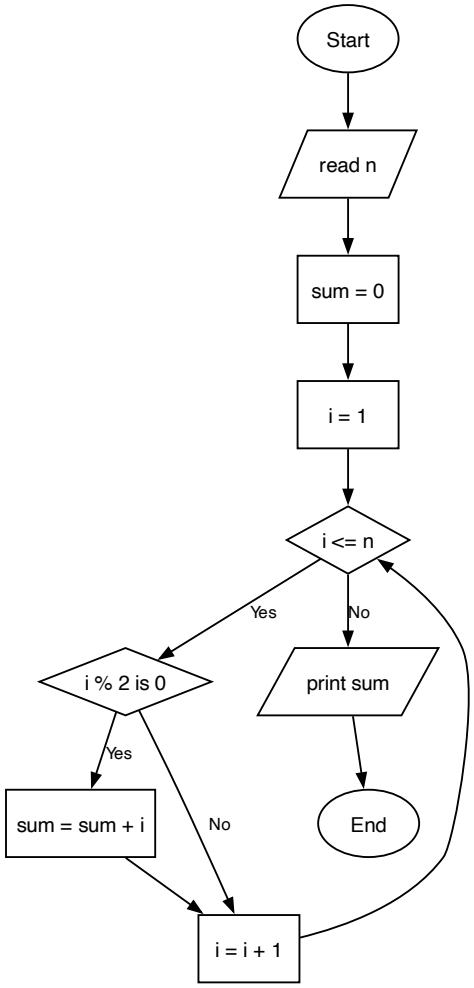
Trace this algorithm for $n = 4$, and complete the trace table.



n	sum	i	output

Problem 3.

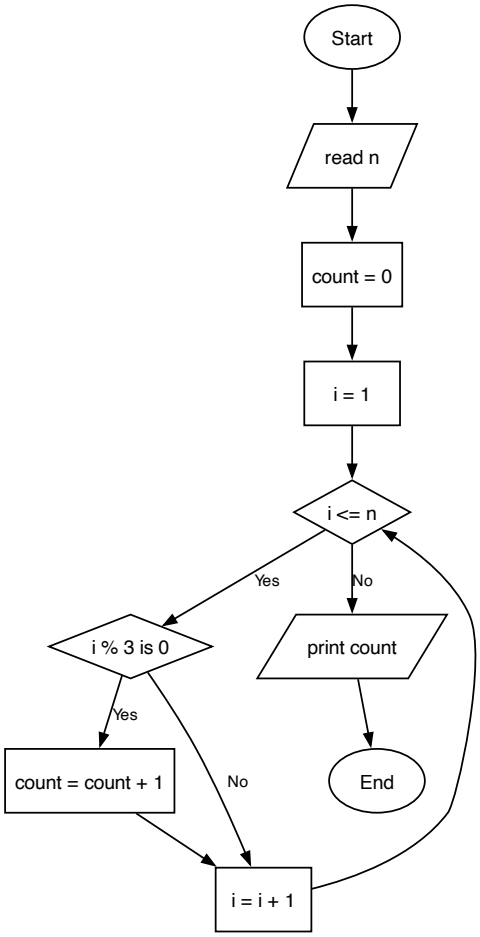
Trace this algorithm for $n = 5$, and complete the trace table.



n	sum	i	output

Problem 4.

Trace this algorithm for $n = 7$, and complete the trace table.

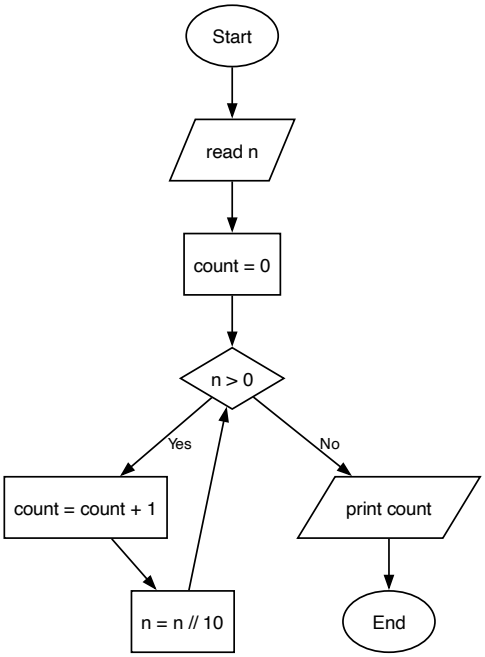


n	count	i	output

Problem 5.

Trace this algorithm for $n = 472$, and complete the trace table.

Note: n // 10 means whole-number division — divide and drop the remainder. For example, 472 // 10 = 47 (not 47.2).



n	count	output

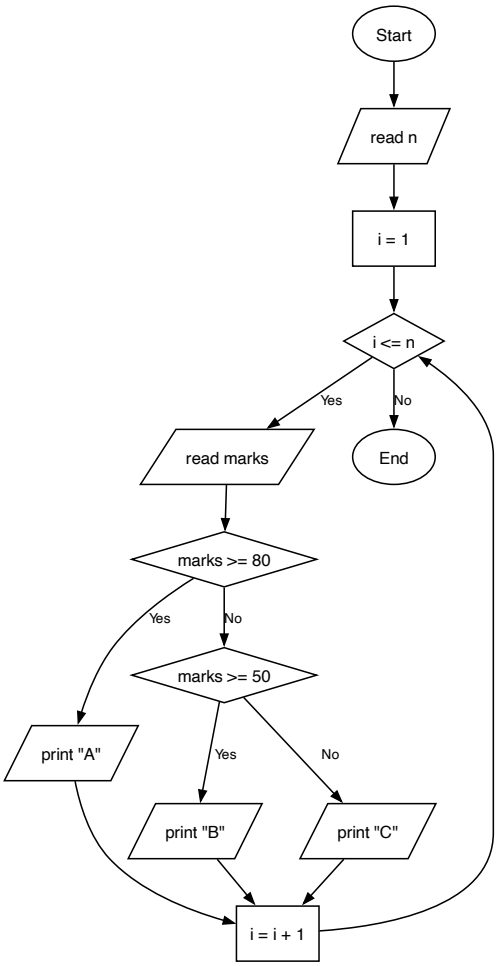
Problem 6.

> *Ask the user for a number n. Then go through the numbers 1, 2, 3, … up to n, one by one: if a number is divisible by 3, print "buzz"; otherwise print the number itself.*

Draw a flowchart for this algorithm:

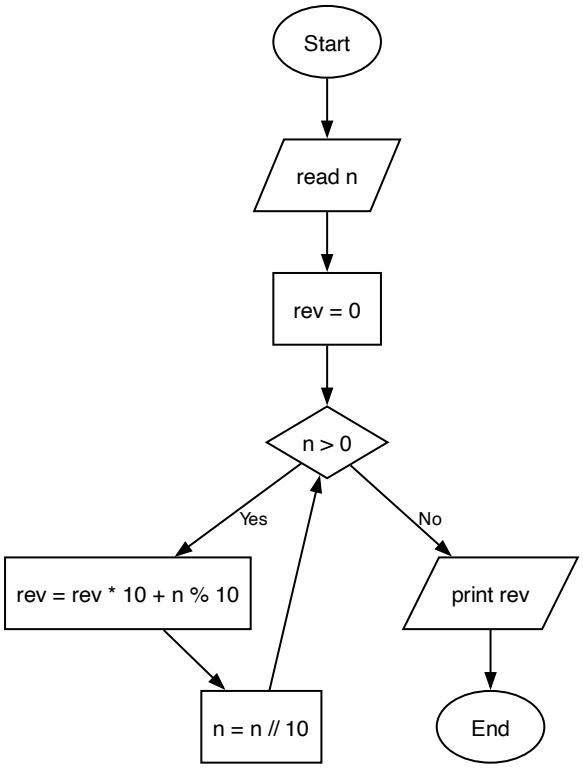
Problem 7.

Trace this algorithm for $n = 3$, where the values of *marks* entered are 55, 82, 40 (in that order), and complete the trace table.



n	i	marks	output

Problem 8.



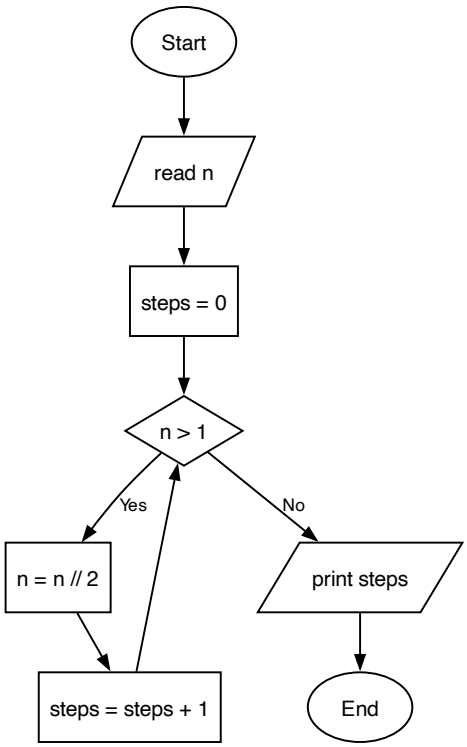
(i) For each input below, follow the flowchart and write the output.
Note: n % 10 gives the remainder after dividing by 10 — the last digit of n. For example, 472 % 10 = 2.

input n	output
123	
47	
5	
470	

(ii) Can you tell what this algorithm is doing?

Problem 9.

(i) Trace this algorithm for $n = 16$, and complete the trace table.



n	steps	output

(ii) For each input below, follow the flowchart and write the output.

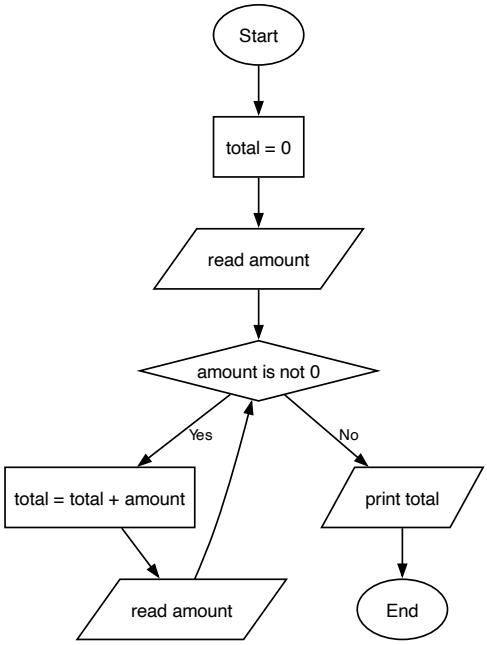
input n	output
8	
16	
100	
1000	

(iii) n grew from 8 all the way to 1000 — but look how little the output grew! Can you explain why?

Problem 10.

At the end of the day, a shopkeeper adds up the money received from each customer. She enters the amounts one by one, and enters 0 when there are no more. The program then shows the total. Notice that the algorithm cannot know in advance how many customers there were!

Trace this algorithm, where the values of *amount* entered are 120, 75, 200, 0 (in that order), and complete the trace table.



total	amount	output

Problem 11.

After an exam, the teacher enters each student's marks one by one, and enters −1 when there are no more. The program should then print three things: how many students took the exam, their average marks, and how many of them passed (40 or more). (Why is −1 a better "no more" signal here than 0?)

Draw a flowchart for this algorithm: