7. Flowcharts 2

What did we do last time?

Flowcharts

- Why flowcharting?
- What is a flowchart?
- Principle of good programming
- Flowchart with programming
- Uses (Advantages) of flowcharts
- Disadvantages of flowcharts

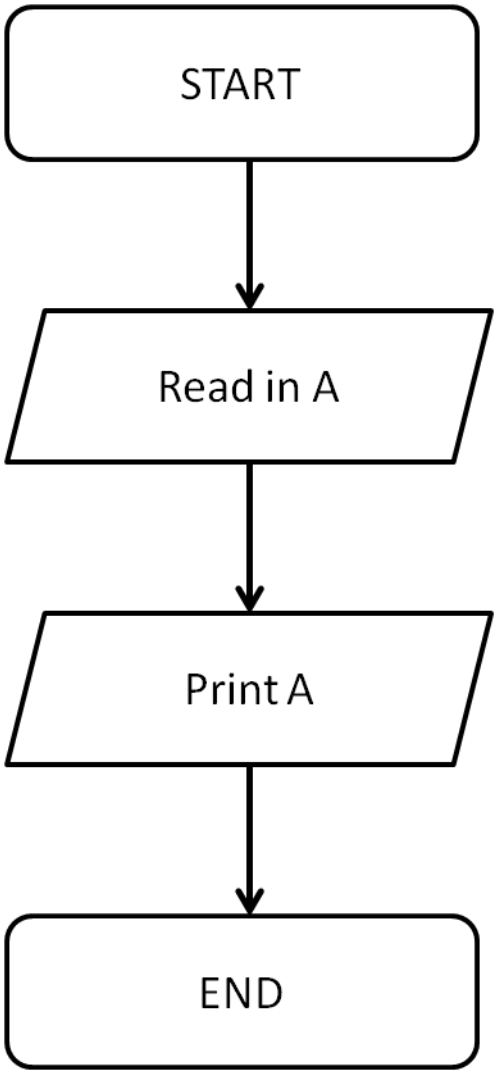
These are the flowcharting symbols

Symbol	Name	Function
	Start/end	An oval represents a start or end point.
	Arrows	A line is a connector that shows relationships between the representative shapes.
	Input/Output	A parallelogram represents input or ouptut.
	Process	A rectangle represents a process.
	Decision	A diamond indicates a decision.

Flowcharts – read in, print out (Problem 1)

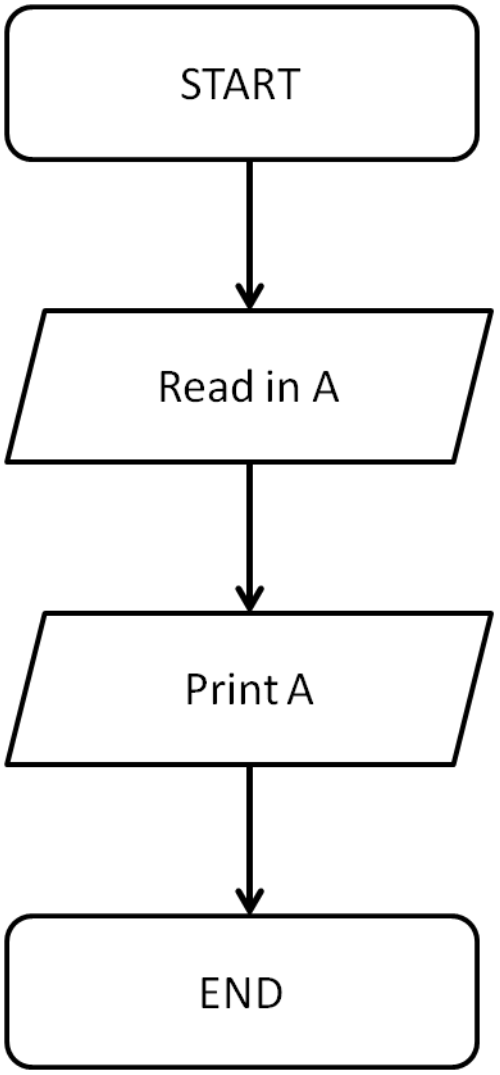
- So let's say we want to express the following algorithm:

Read in a number and print it out.



C version

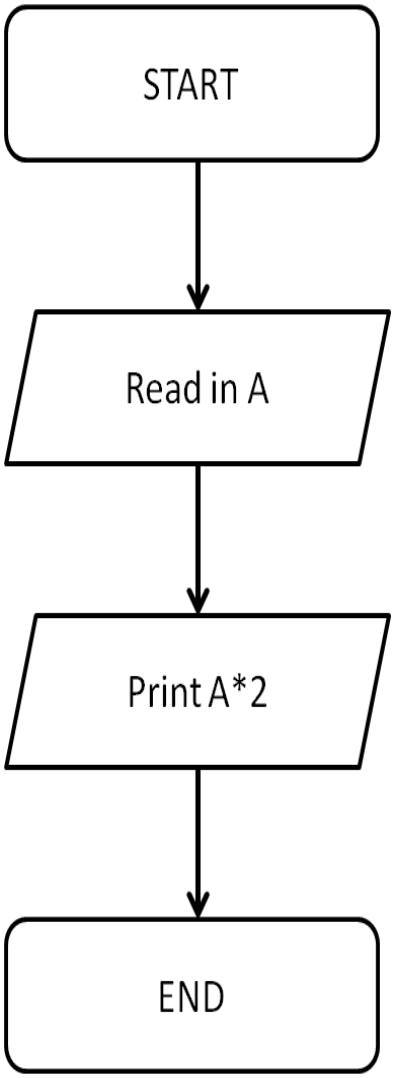
```
#include <stdio.h>
int main()
{
        int A;

        scanf("%d", &A);

        printf("You entered %d", A);

        return 0;
 }
```



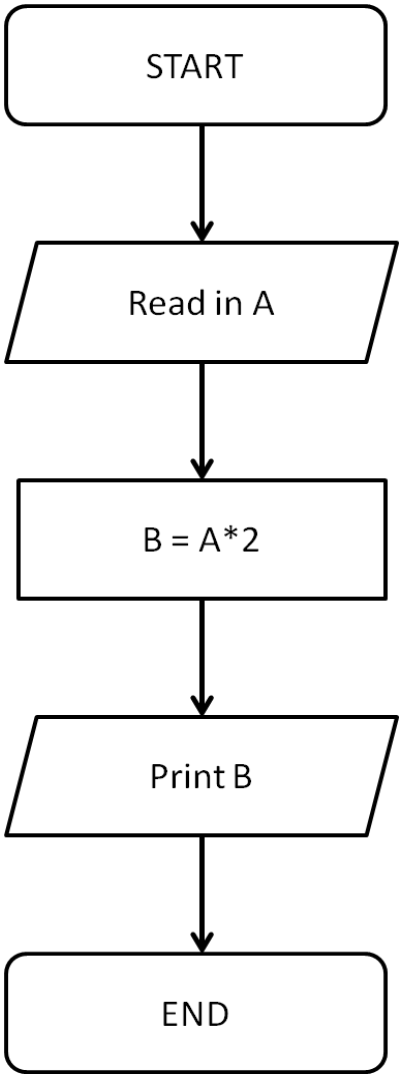
Flowcharts – multiply by 2 (Problem 2)

- So let's say we want to express the following algorithm:

Read in a number and print it out double the number.



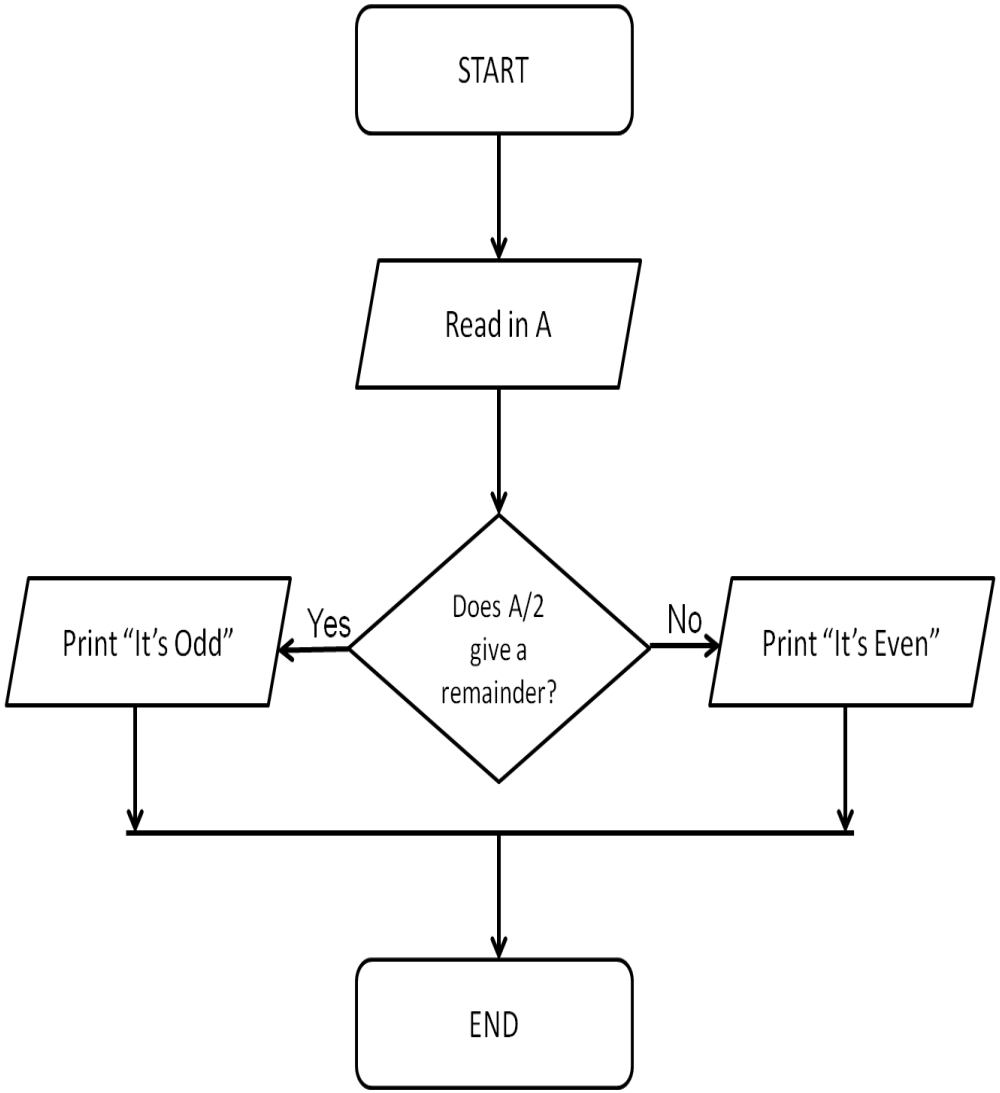
OR



Flowcharts – odd or even? (Problem 3)

- So let's say we want to express the following algorithm:

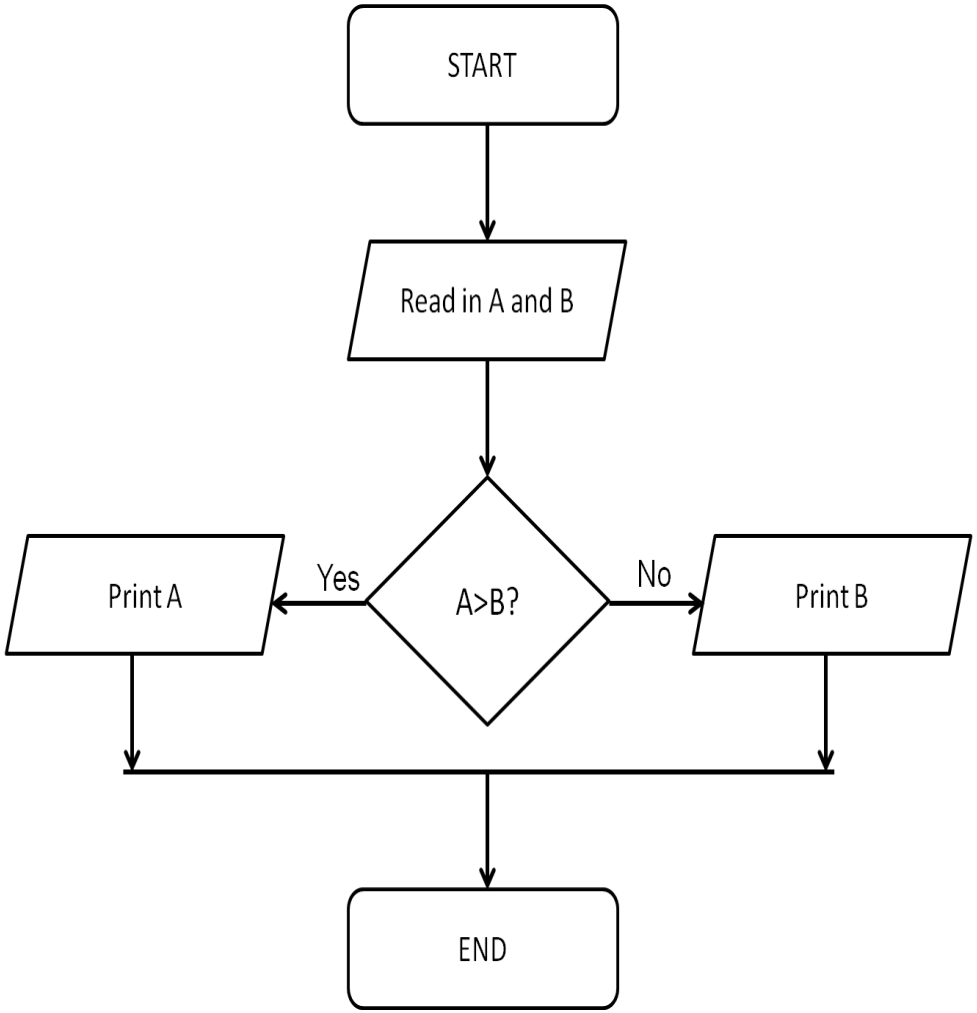
Read in a number, check if it is odd or even.



Flowcharts – is A bigger than B? (Problem 4)

- So let's say we want to express the following algorithm to print out the bigger of two numbers:

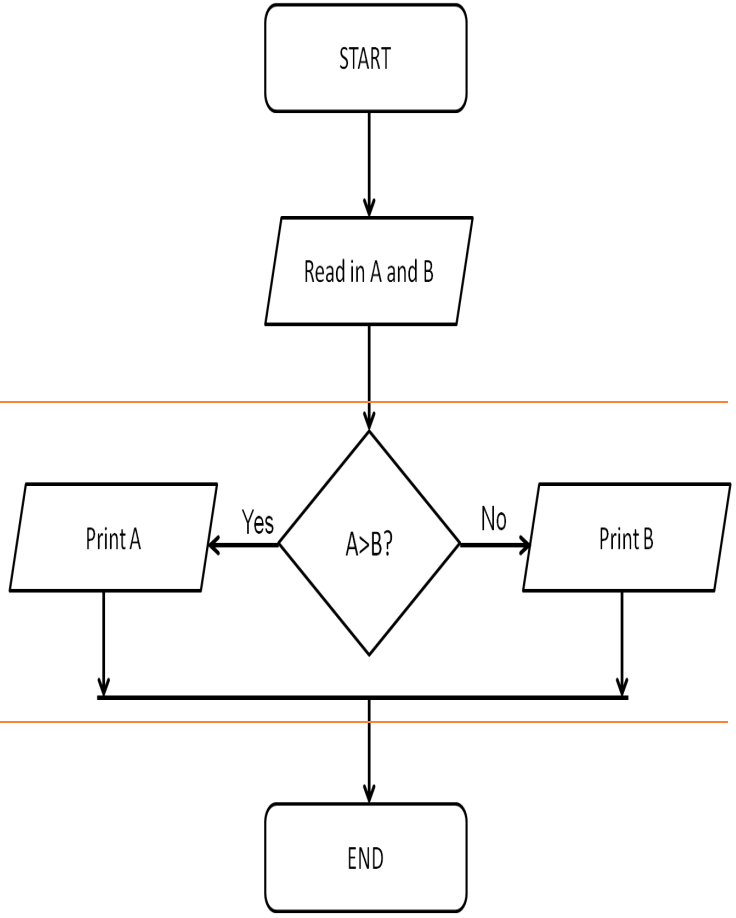
Read in two numbers, call them A and B. If A is bigger than B, print out A, otherwise print out B.



What would happen If A == B ?

C version

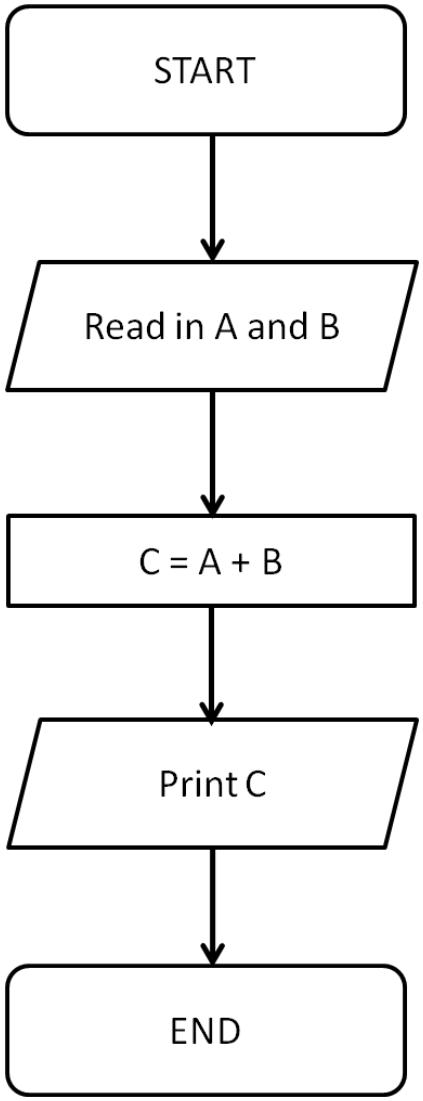
```
#include <stdio.h>
int main()
{
  int A;
  int B;

  printf("Please enter number A:");
  scanf("%d", &A);

  printf("Please enter number B:");
  scanf("%d", &B);

  if(A > B)
  { printf("The biggest is %d", A); }
  else
  { printf("The biggest is %d", B); }

  return 0;
}
```



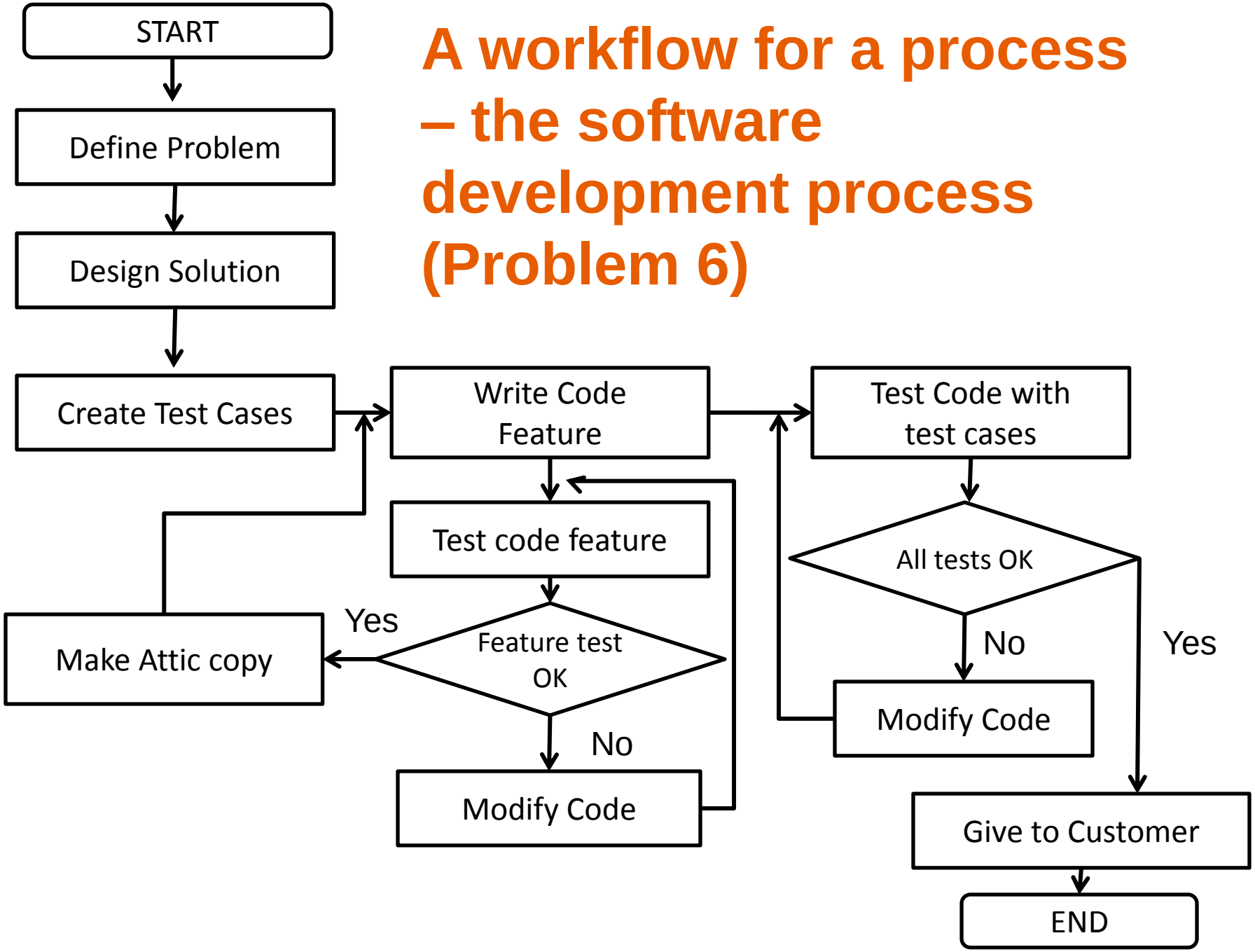
Flowcharts – add two numbers (Problem 5)

- So let's say we want to express the following algorithm to print out the sum of two numbers:

Read in two numbers, call them A and B. Sum A and B, print out the result.



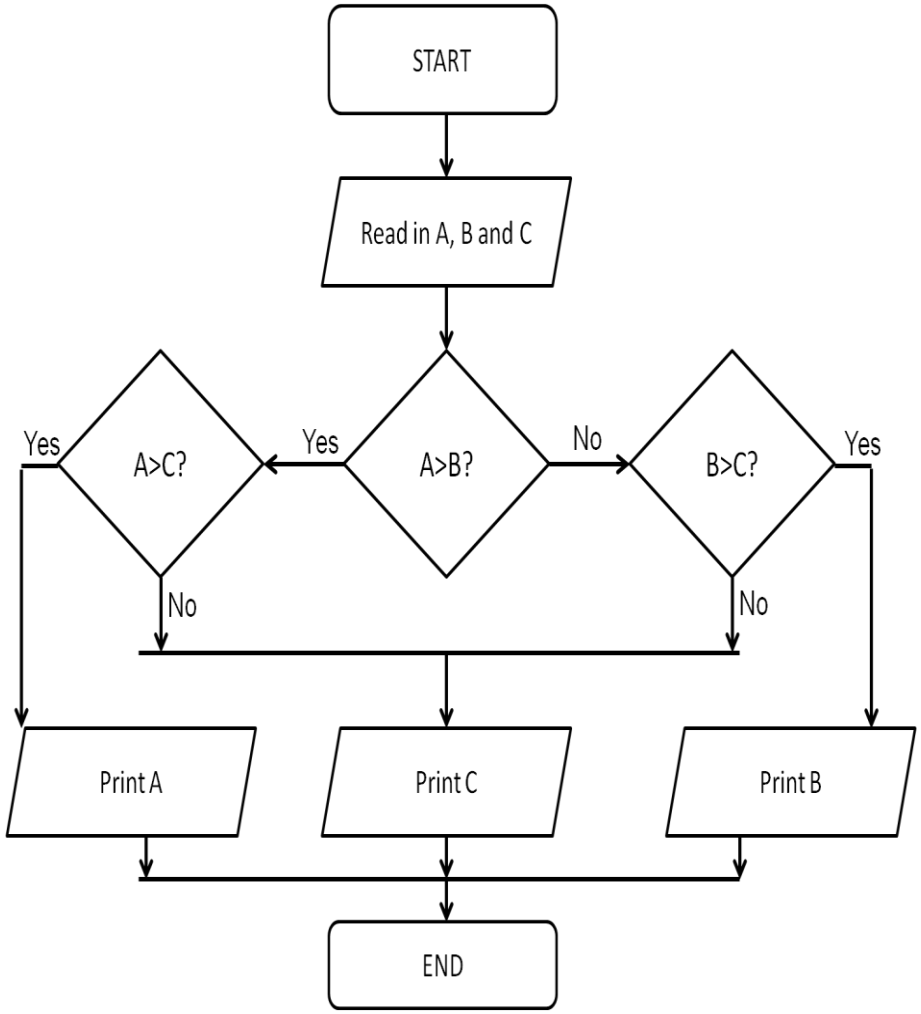
A workflow for a process – the software development process (Problem 6)



Flowcharts – which is bigger, A,B,C (Problem 7)

- So let's say we want to express the following algorithm to print out the bigger of three numbers:

Read in three numbers, call them A, B and C. If A is bigger than B, then if A is bigger than C, print out A, otherwise print out C. If B is bigger than A, then if B is bigger than C, print out B, otherwise print out C.

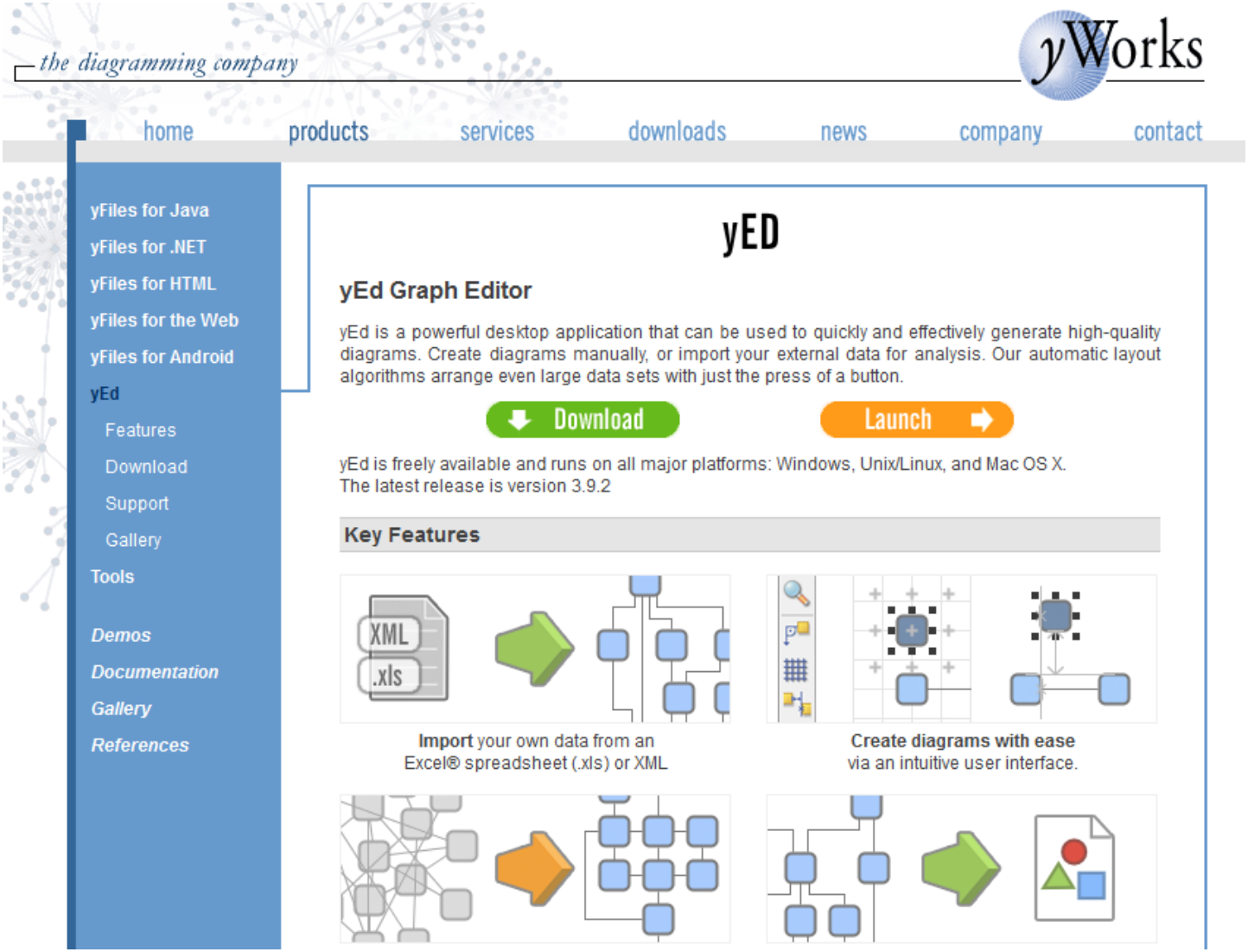


What would happen If A == B == C?

Program Design

OPEN SOURCE DIAGRAM TOOLS

yED



http://www.yworks.com/en/products_yed_about.html

Dia Portable

PortableApps.com
YOUR DIGITAL LIFE, ANYWHERE™

Log in Create new account

Home Download Get Apps Forums Development Support About Us Please Donate Search

New Apps Every Week: KchmViewer, RBTray, Wise Registry Cleaner, WackGet, Quick Blackjack, Sqliteman, more... | Updated Apps All The Time!
New PortableApps.com Platform 11.1 (Oct 8, 2012) | A personal appeal from PortableApps.com founder, John T. Haller

Portable App Directory™
- Accessibility
- Development
- Education
- Games
- Graphics & Pictures
- Internet
- Music & Video
- Office
- Security
- Utilities

User login

Username: *

Password: *

Log in
- Create new account
- Request new password

Follow & Connect With Us

Like 20k

Find us on Facebook twitter

Home » Portable App Directory » Office

Dia Portable

Like 103 Send Tweet 5 +1 5 Share 1 1

diagramming to go

Dia is roughly inspired by the commercial Windows program Visio, though more geared towards informal diagrams for casual use. It can be used to draw many different kinds of diagrams and has special objects to help draw entity relationship diagrams, UML diagrams, flowcharts, network diagrams, and many other diagrams. It's packaged as a portable app, so you can do your technical drawing wherever you are.

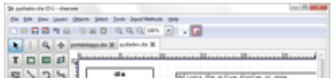
Download Now
sourceforge - Trusted for Open Source
Version 0.97.2 for Windows, Multilingual
15MB download / 25-52MB installed (Details)

Make a Donation - Support PortableApps.com's development and hosting

Dia Portable works best with the PortableApps.com Platform

Features

Dia is a full-featured diagramming program. It can be used to draw many different kinds of diagrams and has special objects to help draw entity relationship diagrams,



http://portableapps.com/apps/office/dia_portable

Graphviz

Graphviz - Graph Visualization Software

Drawing graphs since 1988

Forums | Wiki | Contact Us

Search this site: Search

- **Home**
- About
- Download
- News
- Gallery
- Documentation
- Theory
- Bug and Issue Tracking
- Mailing List
- License
- Resources
- Credits
- Forums
- FAQ
- Wiki

User login

Username: *

Password: *

Log in

- Create new account
- Request new password

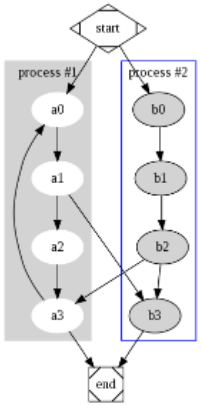
Graphviz

Welcome to Graphviz

Available translations: Belorussian, Romanian, Russian, Russian (more natural?)

What is Graphviz?

Graphviz is open source graph visualization software. Graph visualization is a way of representing structural information as diagrams of abstract graphs and networks. It has important applications in networking, bioinformatics, software engineering, database and web design, machine learning, and in visual interfaces for other technical domains.



Features

The Graphviz layout programs take descriptions of graphs in a simple text language, and make diagrams in useful formats, such as images and SVG for web pages, PDF or Postscript for inclusion in other documents; or display in an interactive graph browser. (Graphviz also supports GXL, an XML dialect.) Graphviz has many useful features for concrete diagrams, such as options for colors, fonts, tabular node layouts, line styles, hyperlinks, rolland custom shapes.

Roadmap

dot "hierarchical" or layered drawings of directed graphs. This is the default tool to use if edges have directionality.

neato "spring model" layouts. This is the default tool to use if the graph is not too large (about 100 nodes) and you don't know anything else about it. Neato attempts to minimize a global energy function, which is equivalent to

Active forum topics

- svg and cmap coordinates seem inconsistent
- html markup in tooltips
- Dot.exe not running under windows 2003 server
- Images in nodes, label below
- Installation on ubuntu

more

New forum topics

- svg and cmap coordinates seem inconsistent
- html markup in tooltips
- Images in nodes, label below
- how to compress a graph?
- Dot.exe not running under windows 2003 server

more

http://www.graphviz.org/

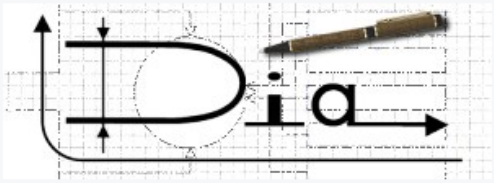
Dia

Home | News | Projects | Art | Suppo

live.gnome.org

Search

Home | RecentChanges | Schedule



Home	News	Screenshots	Examples	Download	Documentation
FAQ	Bugzilla	Development	Python	Links	Contact

Welcome to Dia's new homepage. Dia is a GTK+ based diagram creation program for GNU/Linux, MacOS X, Unix, and Windows, and is released under the GPL license.

News! 2011-Dec-18: Version 0.97.2 has been released. Visit the Download page to get your copy! (Download shortcuts: Windows,Mac OS X)

Dia is roughly inspired by the commercial Windows program 'Visio,' though more geared towards informal diagrams for casual use. It can be used to draw many different kinds of diagrams. It currently has special objects to help draw entity relationship diagrams, UML diagrams, flowcharts, network diagrams, and many other diagrams. It is also possible to add support for new shapes by writing simple XML files, using a subset of SVG to draw the shape.

It can load and save diagrams to a custom XML format (gzipped by default, to save space), can export diagrams to a number of formats, including EPS, SVG, XFIG, WMF and PNG, and can print diagrams (including ones that span multiple pages).

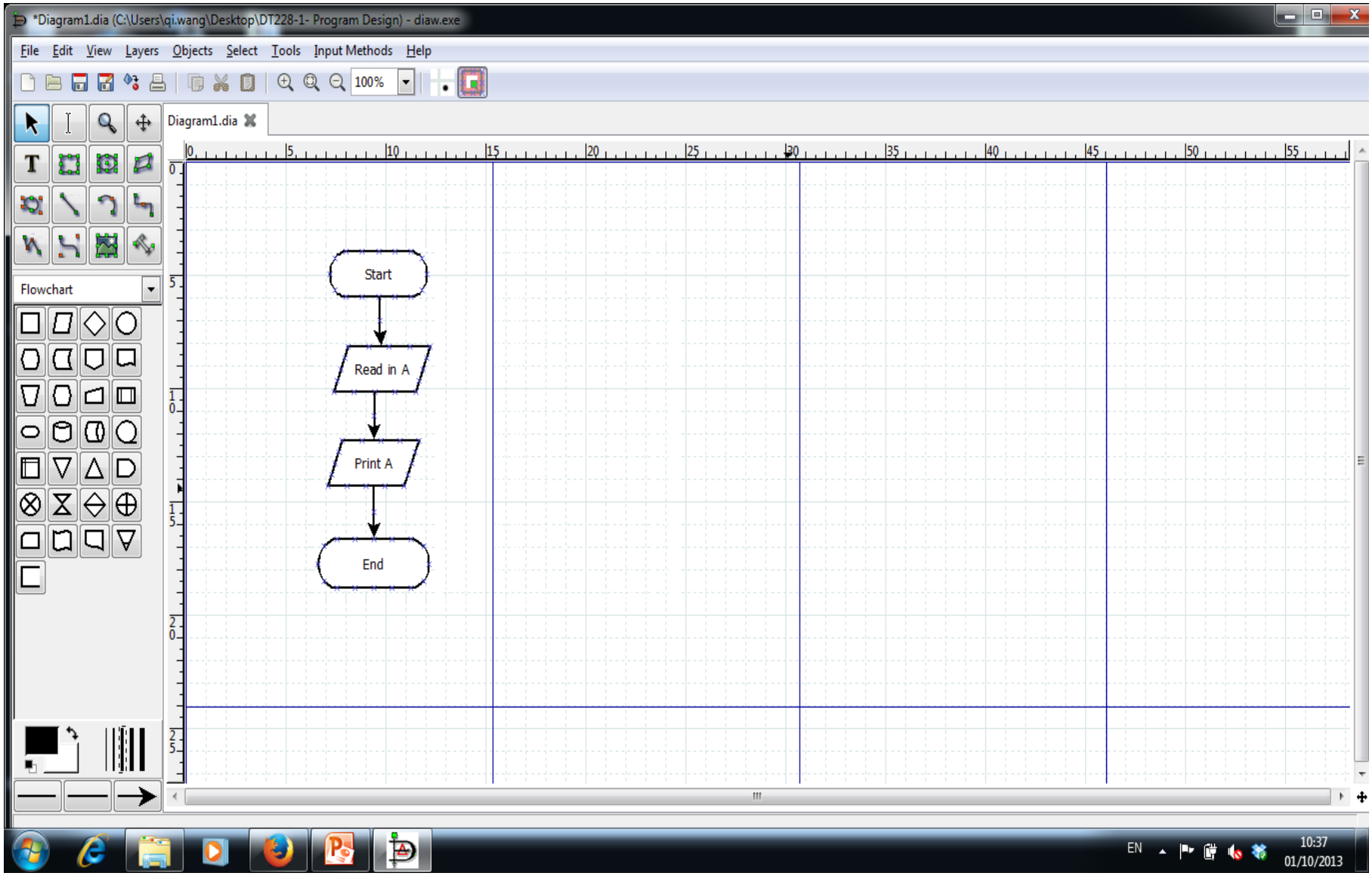
Download Dia and try using it; tell us what you think of it (visit the Contact page), including to report bugs if you find them. You can even read the Development page to find out how to contribute to the code.

https://live.gnome.org/Dia

http://dia-installer.de/download/index.html

http://dia-installer.de/download/index.html

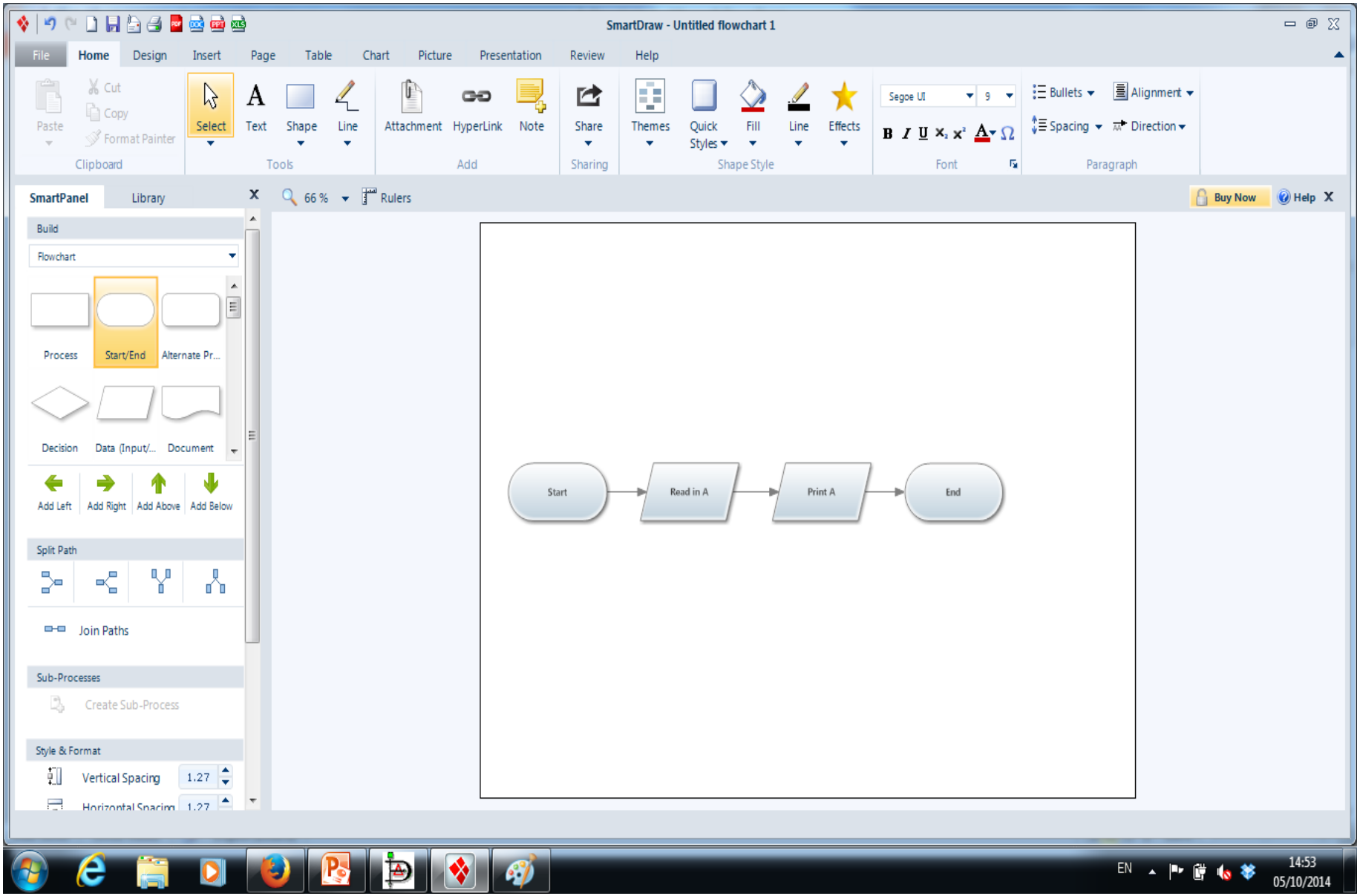
We use Dia Diagram Editor – an example



SmartDraw

http://www.smartdraw.com/

SmartDraw – an example



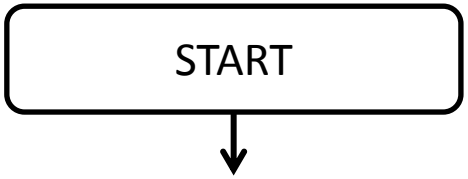
More examples on flowcharts

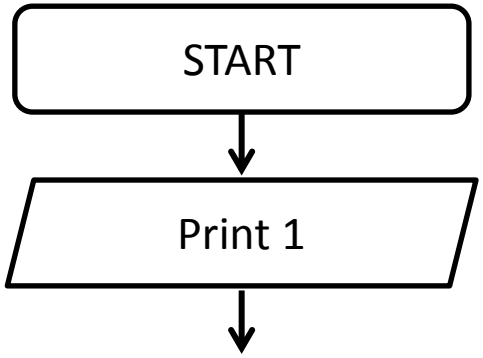
Flowcharts (Problem 8)

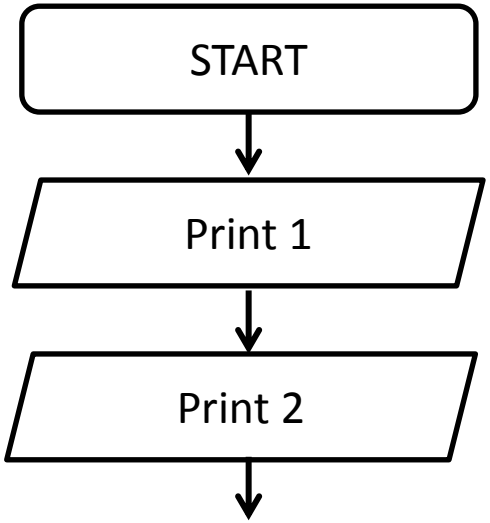
- So let's say we want to express the following algorithm:

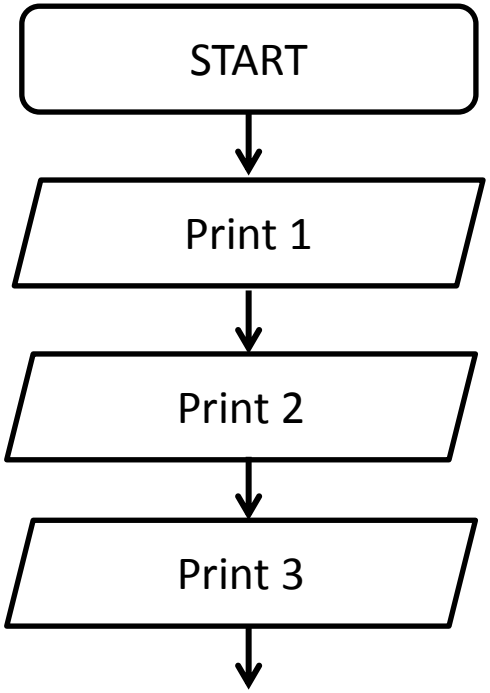
Print out the numbers from 1 to 5

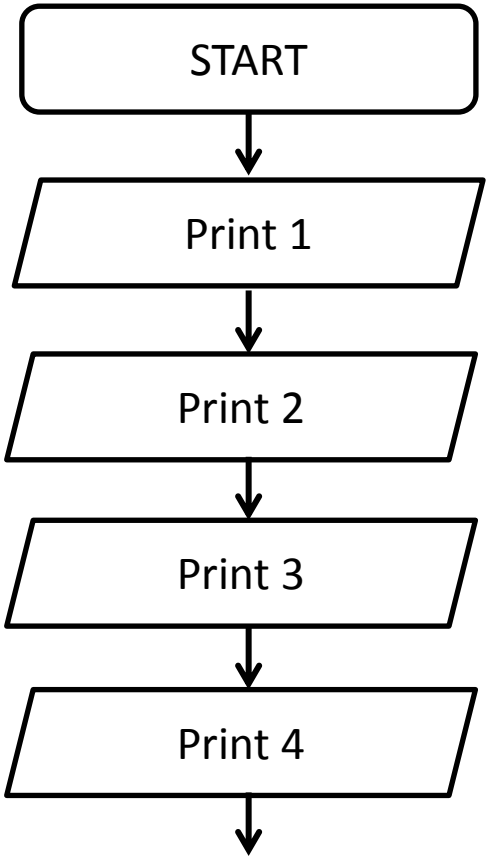
Symbol	Name	Function
	Start/end	An oval represents a start or end point.
	Arrows	A line is a connector that shows relationships between the representative shapes.
	Input/Output	A parallelogram represents input or ouptut.
	Process	A rectangle represents a process.
	Decision	A diamond indicates a decision.

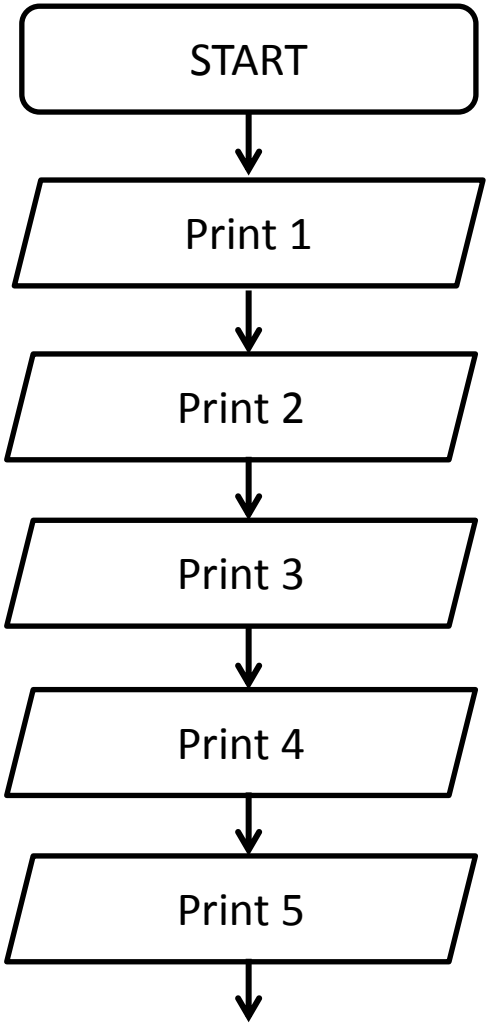

START

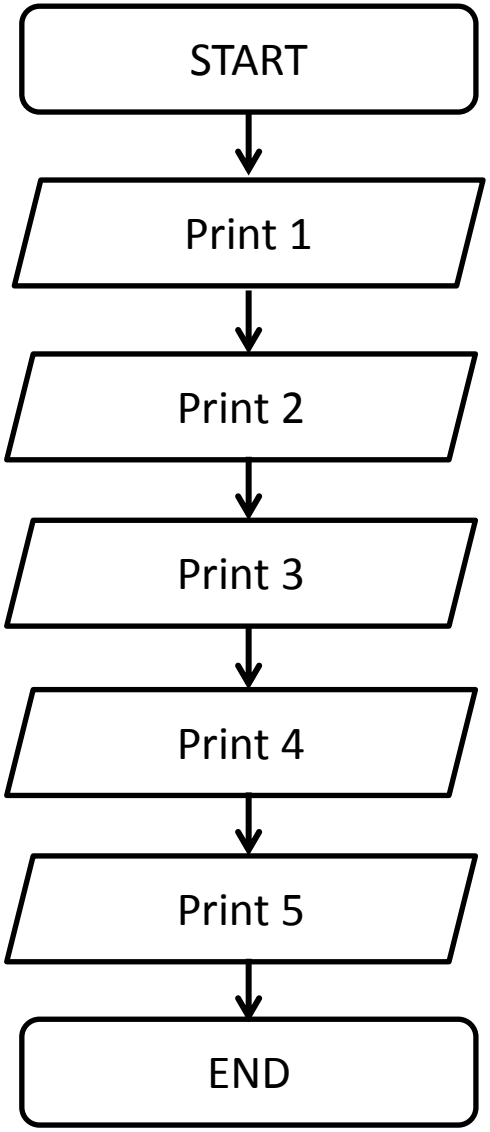

START
Print 1


START
Print 1
Print 2

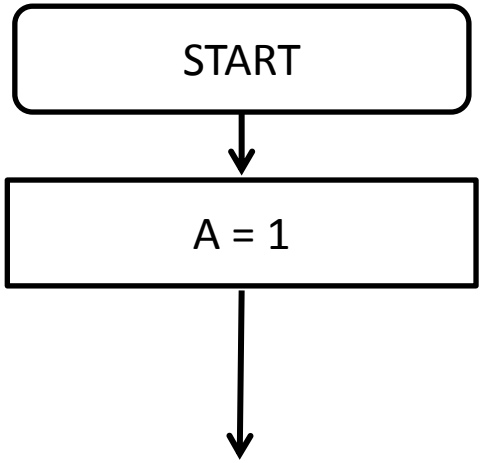

START
Print 1
Print 2
Print 3

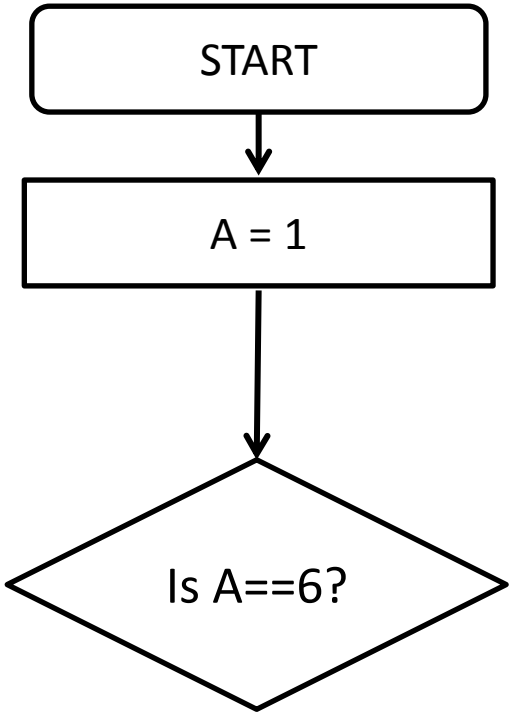

START
Print 1
Print 2
Print 3
Print 4


START
Print 1
Print 2
Print 3
Print 4
Print 5


START
Print 1
Print 2
Print 3
Print 4
Print 5
END

Or alternatively...


START
A = 1



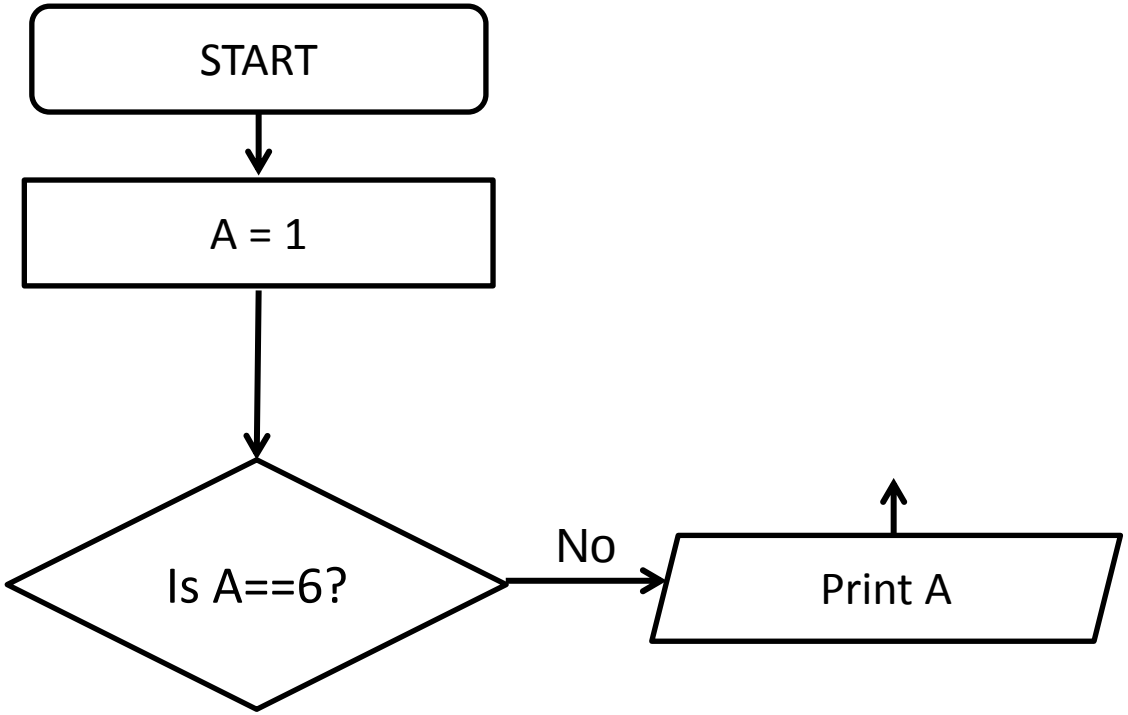
TO CHECK IF EQUAL TO A VALUE WE SAY

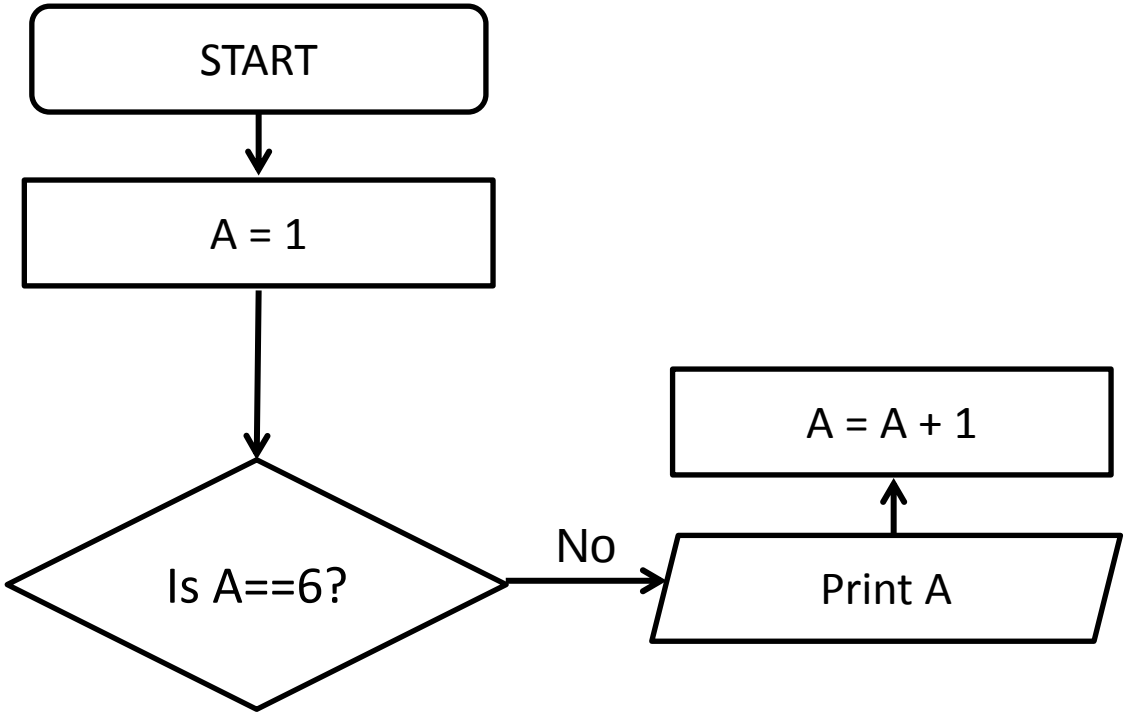
==

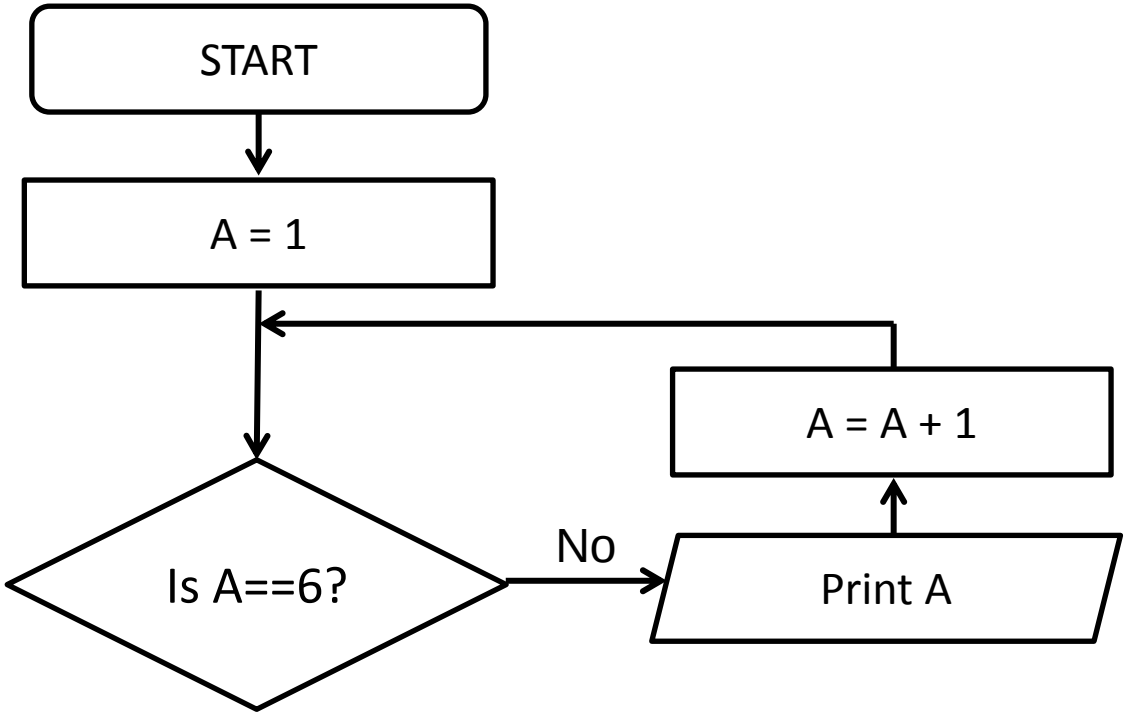
AS WE HAVE USED

=

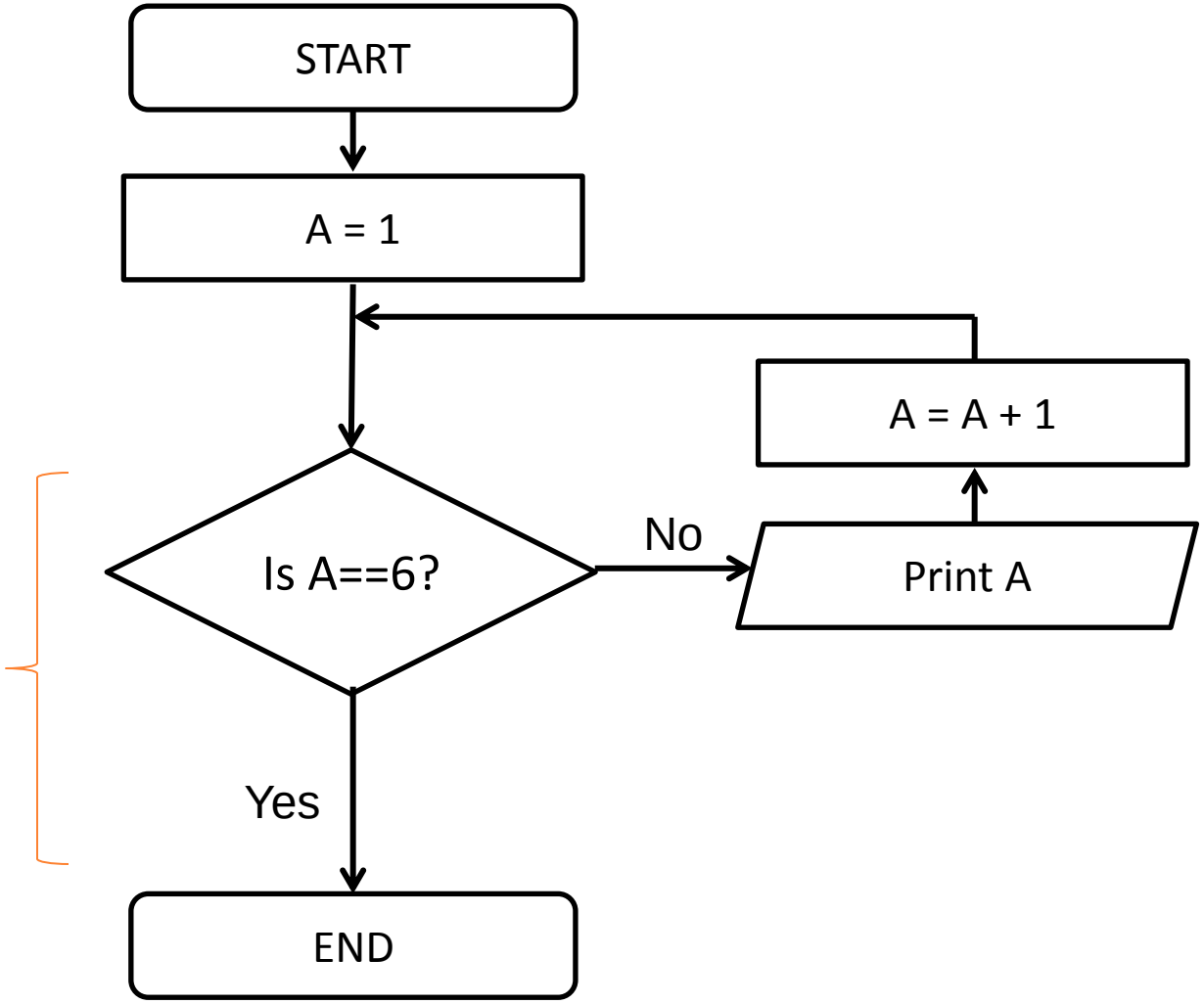
TO MEAN ASSIGN THE VALUE AFTER = TO THE NAMED MEMORY LOCATION


START
A = 1
Is A==6?
No
Print A

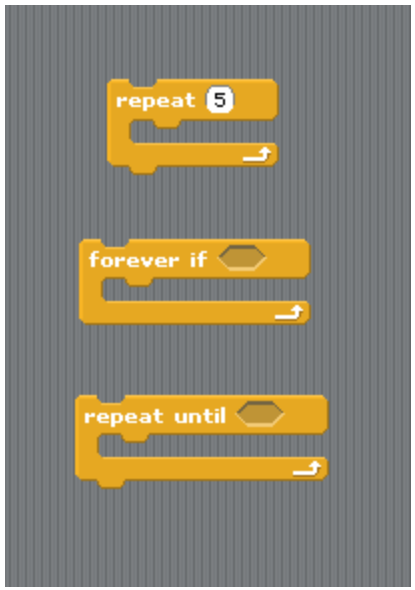

START
A = 1
Is A==6?
No
Print A
A = A + 1


START
A = 1
A = A + 1
Is A==6?
No
Print A

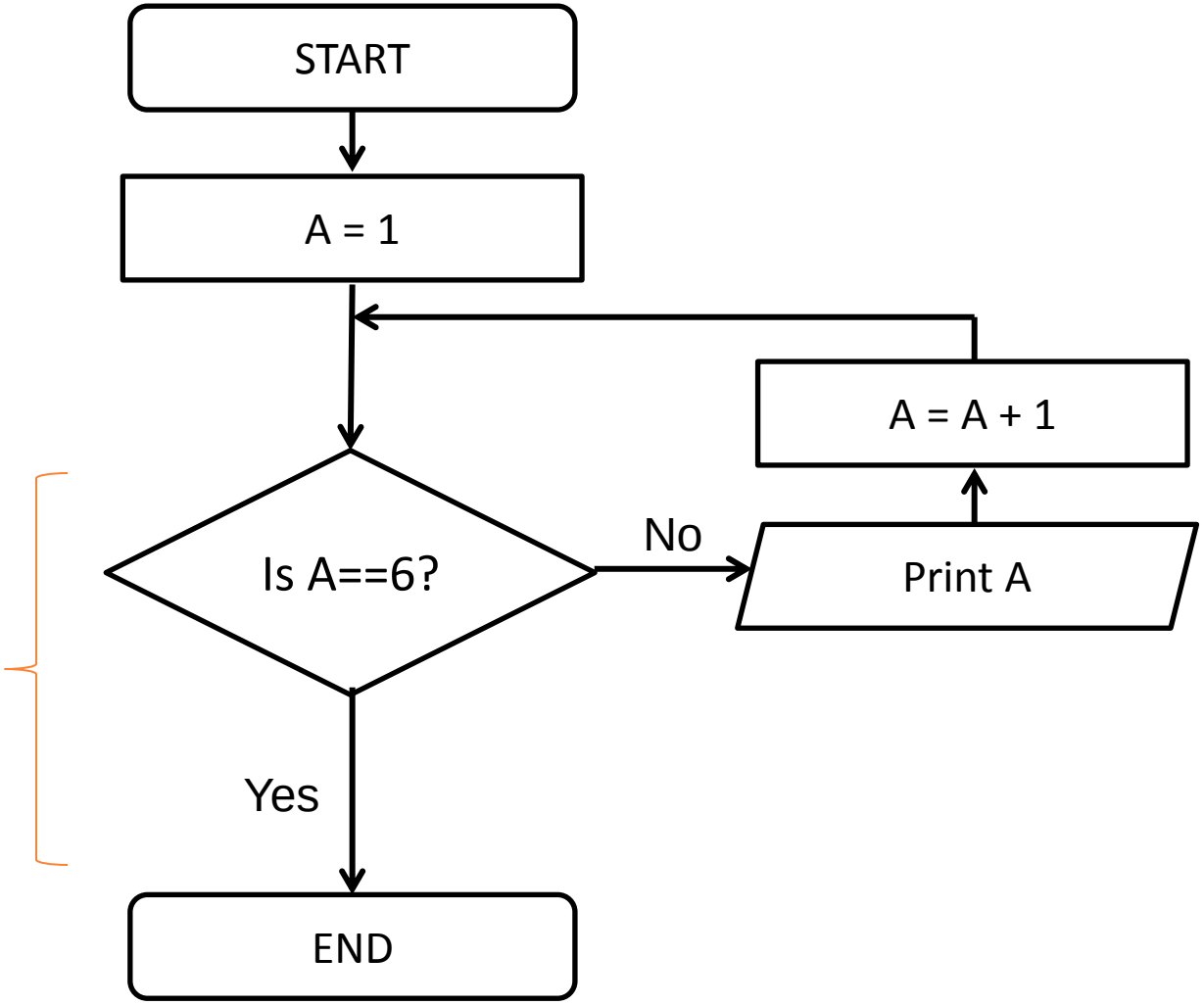
A LOOP



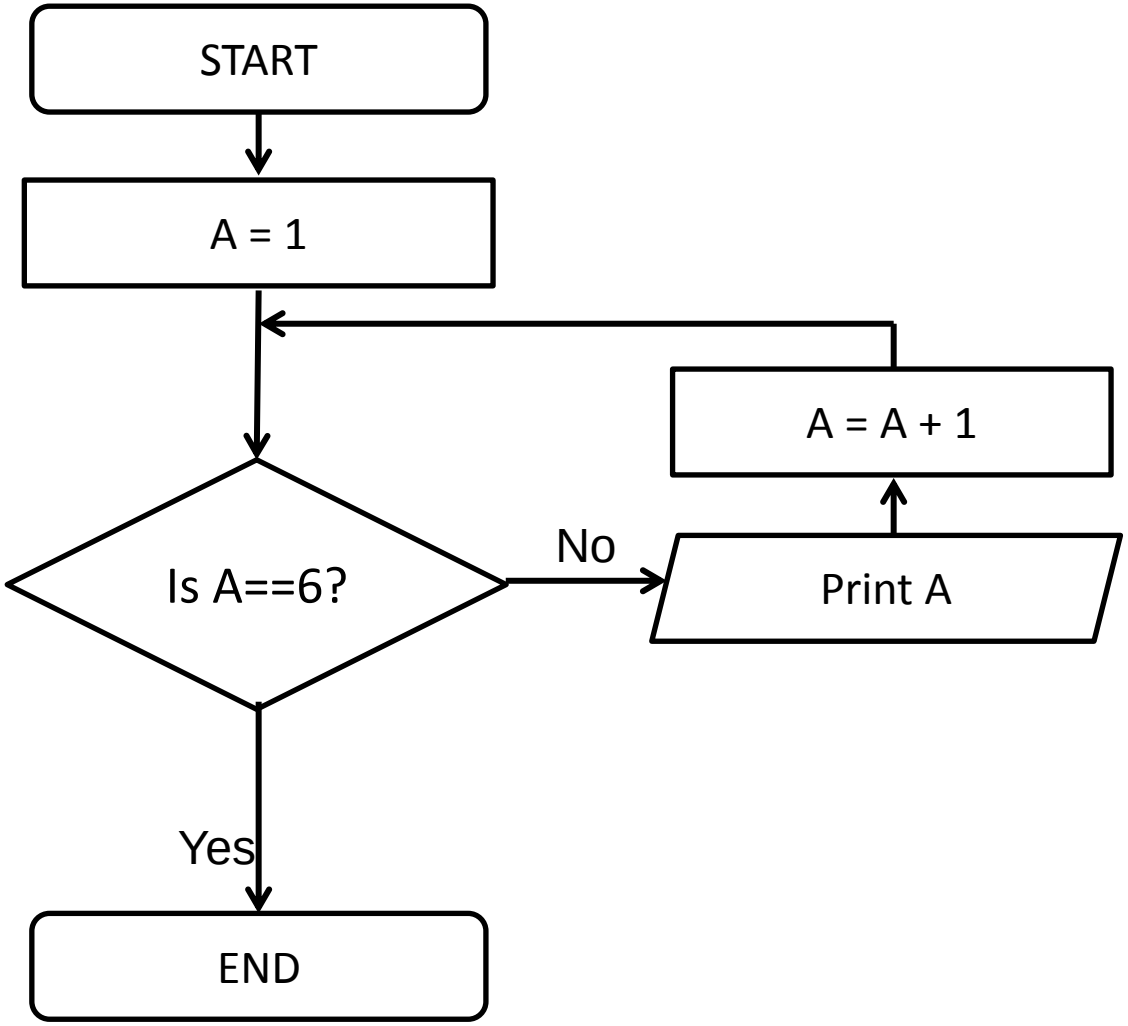
A LOOP



In scratch



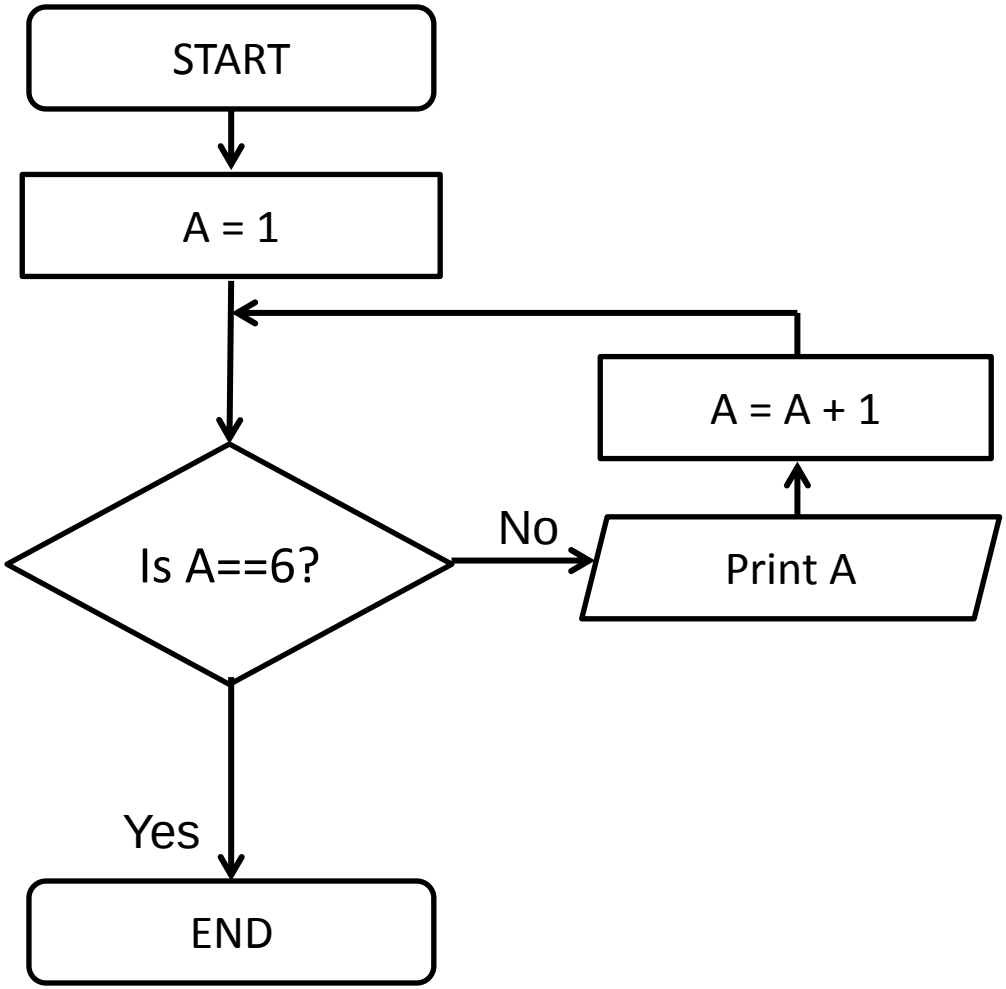
Remember the River crossing Solution model?



Before you put any other code in a loop you should write just this to make sure your loop is iterating as many times as you think it should – not one too many or one less – that's a common loop bug

Remember the River crossing Solution model?

A = 1	A = 1	Is A == 6	NO
A = A +1	A = 2	Is A == 6	NO
A = A +1	A = 3	Is A == 6	NO
A = A +1	A = 4	Is A == 6	NO
A = A +1	A = 5	Is A == 6	NO
A = A +1	A = 6	Is A == 6	YES

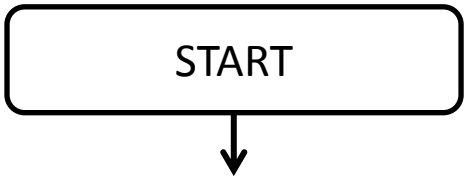


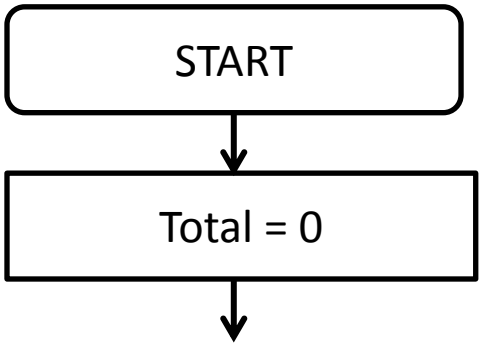
Flowcharts (Problem 9)

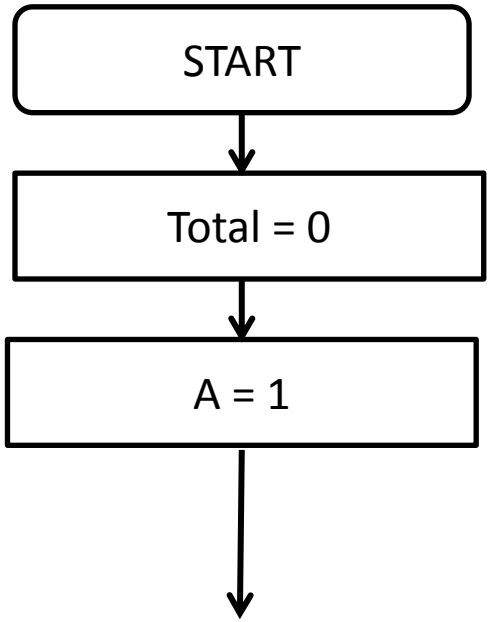
- So let's say we want to express the following algorithm:

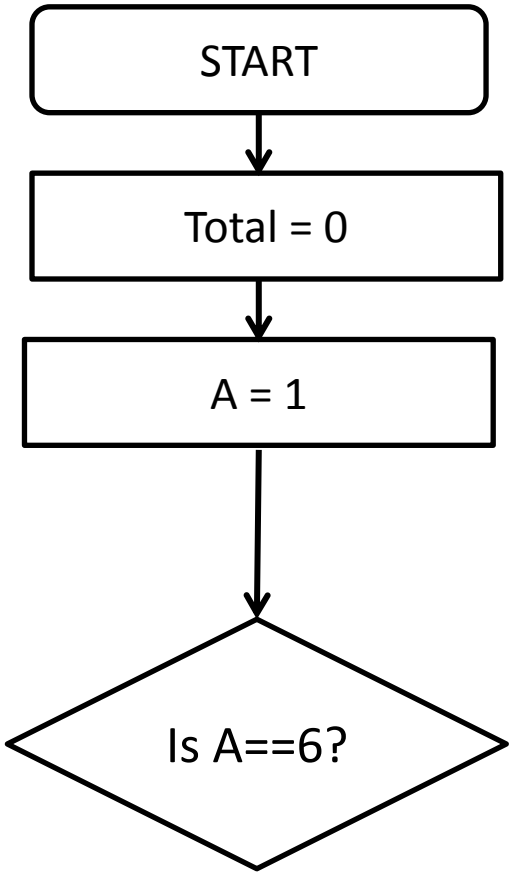
Add up the numbers 1 to 5

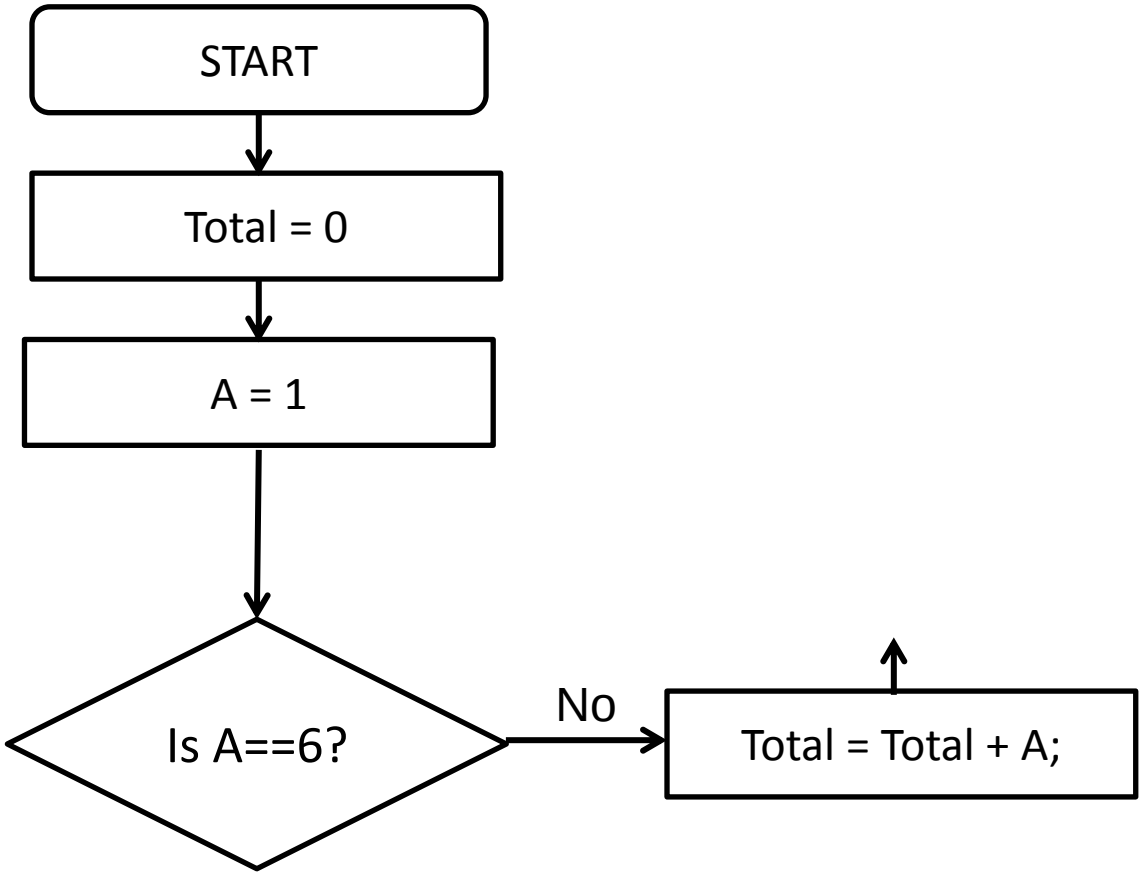
Symbol	Name	Function
	Start/end	An oval represents a start or end point.
	Arrows	A line is a connector that shows relationships between the representative shapes.
	Input/Output	A parallelogram represents input or ouptut.
	Process	A rectangle represents a process.
	Decision	A diamond indicates a decision.

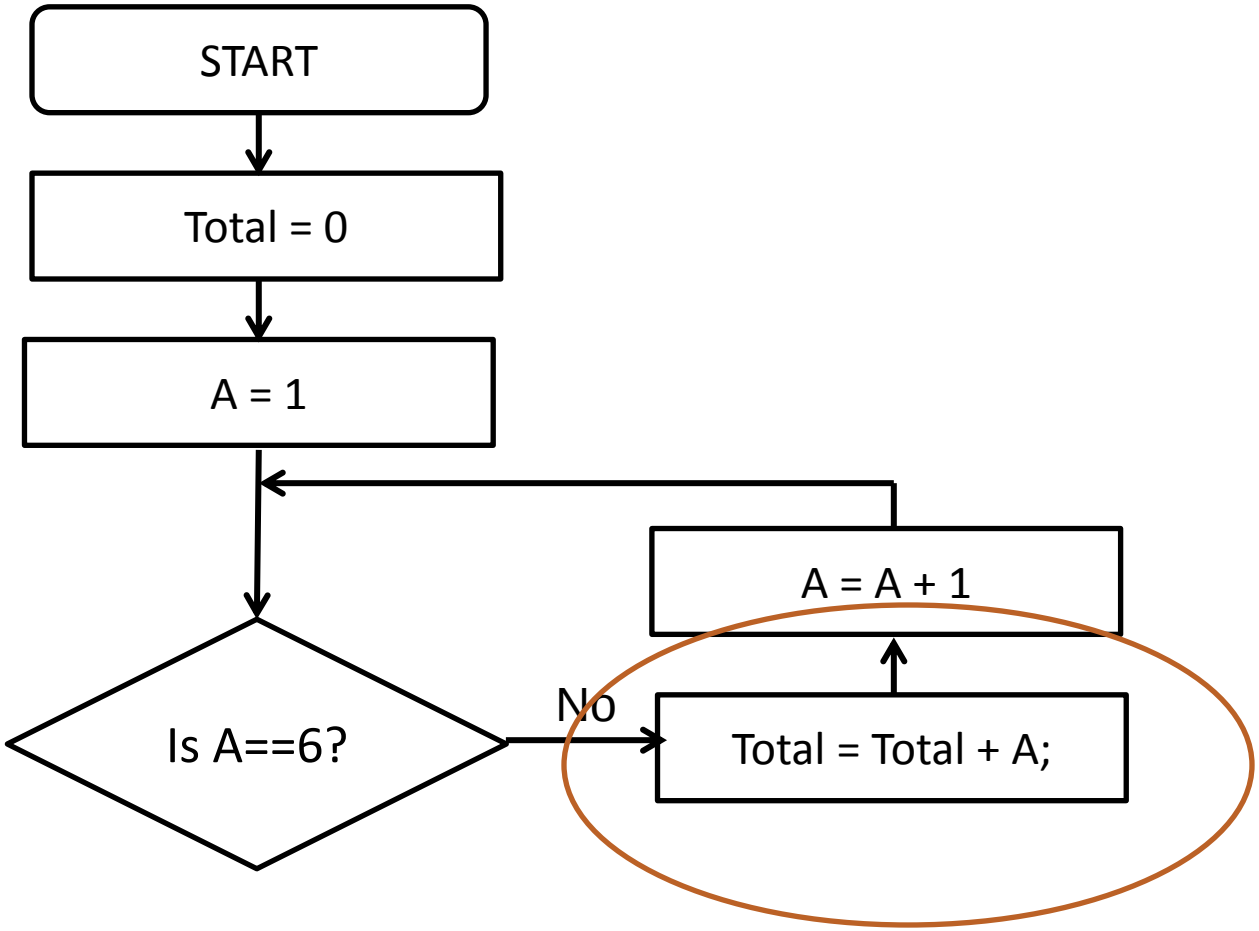

START


START
Total = 0

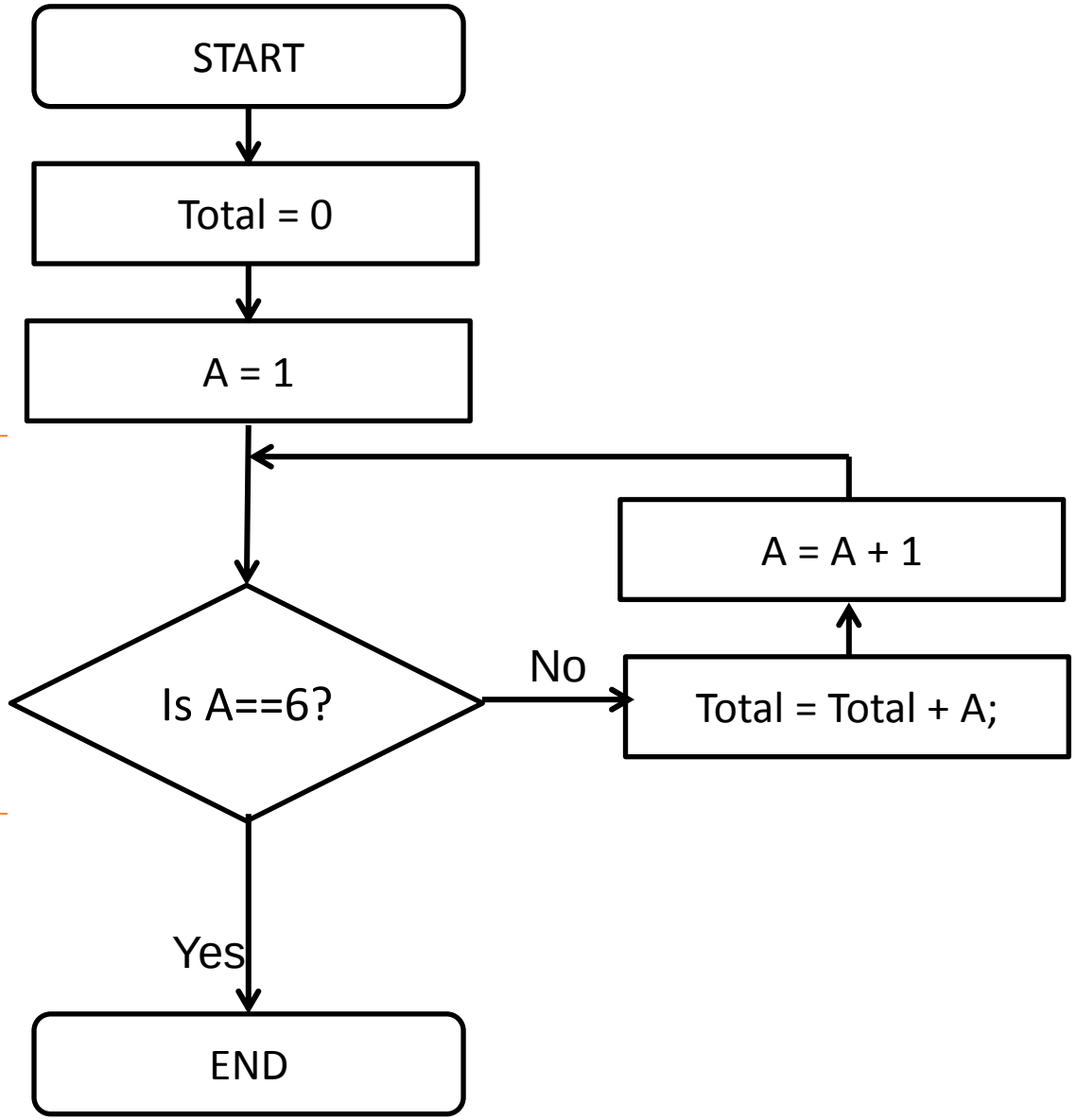

START
Total = 0
A = 1

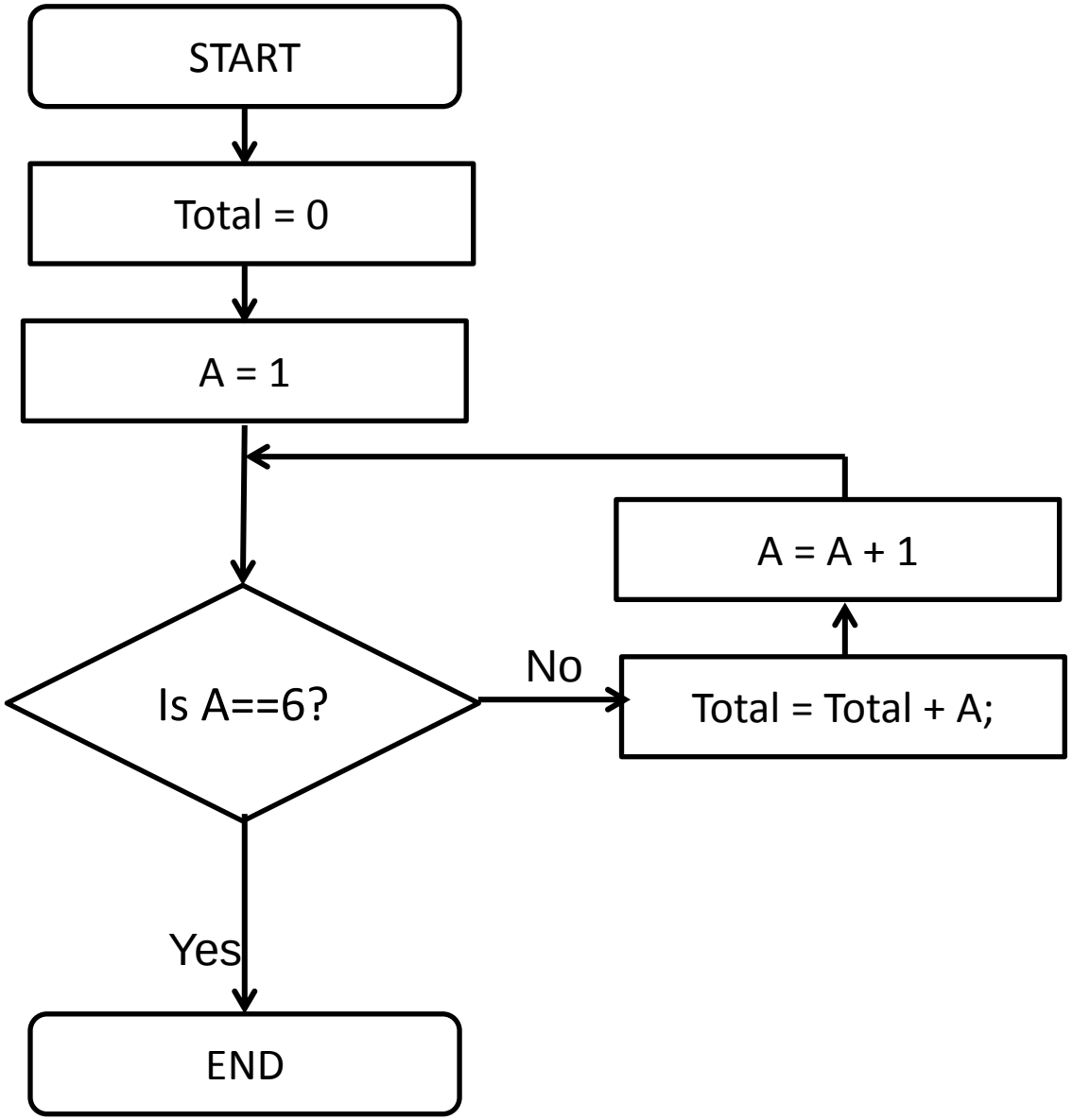

START
Total = 0
A = 1
Is A==6?


START
Total = 0
A = 1
Is A==6?
No
Total = Total + A;


START
Total = 0
A = 1
A = A + 1
Is A==6?
No
Total = Total + A;

A LOOP




START
Total = 0
A = 1
A = A + 1
Is A==6?
No
Total = Total + A;
Yes
END

Flowcharts (Problem 10)

- So let's say we want to express the following algorithm:

Read in a number and check if it's a prime number.

Symbol	Name	Function
	Start/end	An oval represents a start or end point.
	Arrows	A line is a connector that shows relationships between the representative shapes.
	Input/Output	A parallelogram represents input or ouptut.
	Process	A rectangle represents a process.
	Decision	A diamond indicates a decision.

Flowcharts

What's a prime number?

PROBLEM DEFINITION

Flowcharts

- *A prime number is a number that's only divisible by itself and 1, e.g. 1, 2, 3, 5, 7…*

- *Or to put it another way, every number other than itself and 1 gives a remainder, e.g. For 7, if 6, 5, 4, 3, and 2 give a remainder then 7 is prime.*

Flowcharts

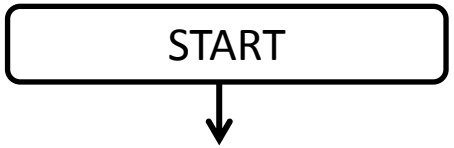
So all we need to do is divide 7 by all numbers less than it but greater than 1, and if all of them have a remainder, we know it is a prime.

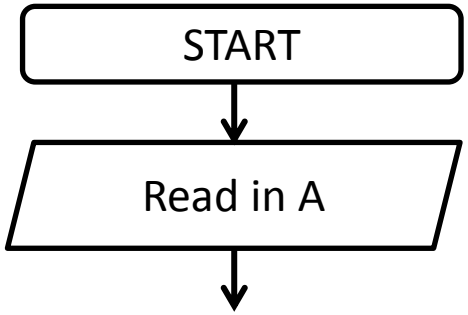
Flowcharts

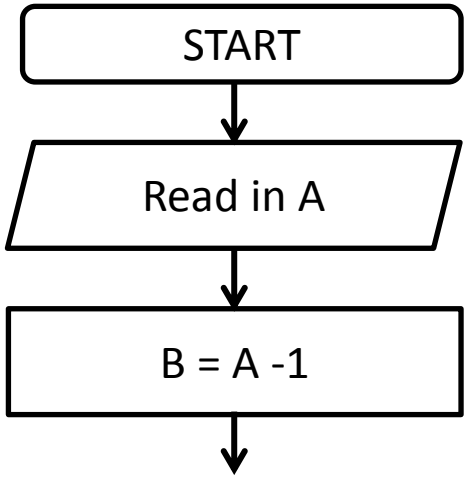
- So,
- If the number is 7, as long as 6, 5, 4, 3, and 2 give a remainder, 7 is a prime.
- If the number is 9, we know that 8, 7, 6, 5, and 4, all give remainders, but 3 does not give a remainder, it goes evenly into 9 so we can say 9 is not prime.

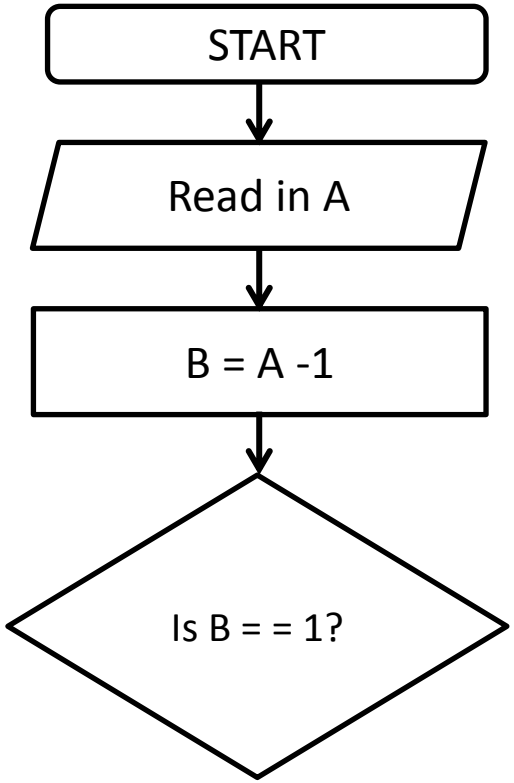
Flowcharts

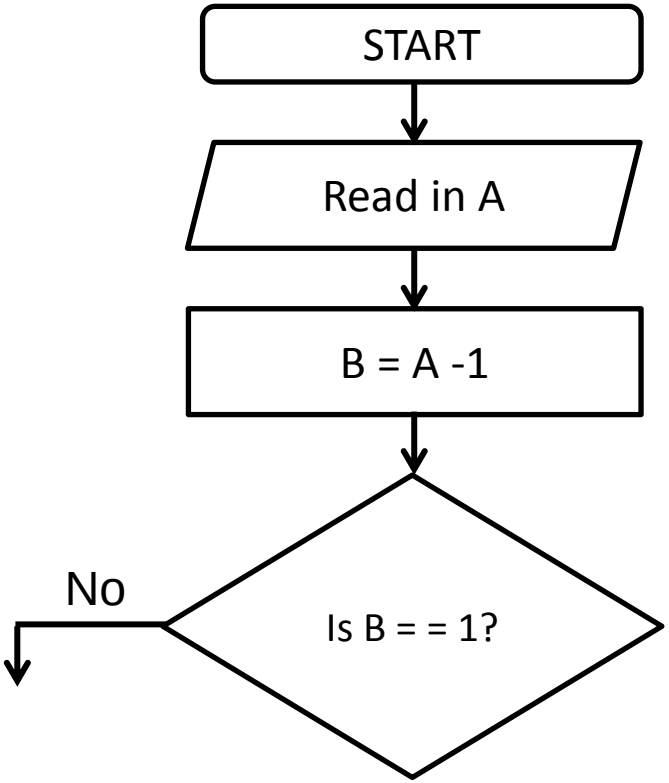
- So in general,
  - if the number is A, as long as A-1, A-2, A-3, A-4, ... 2 give a remainder, A is a prime.

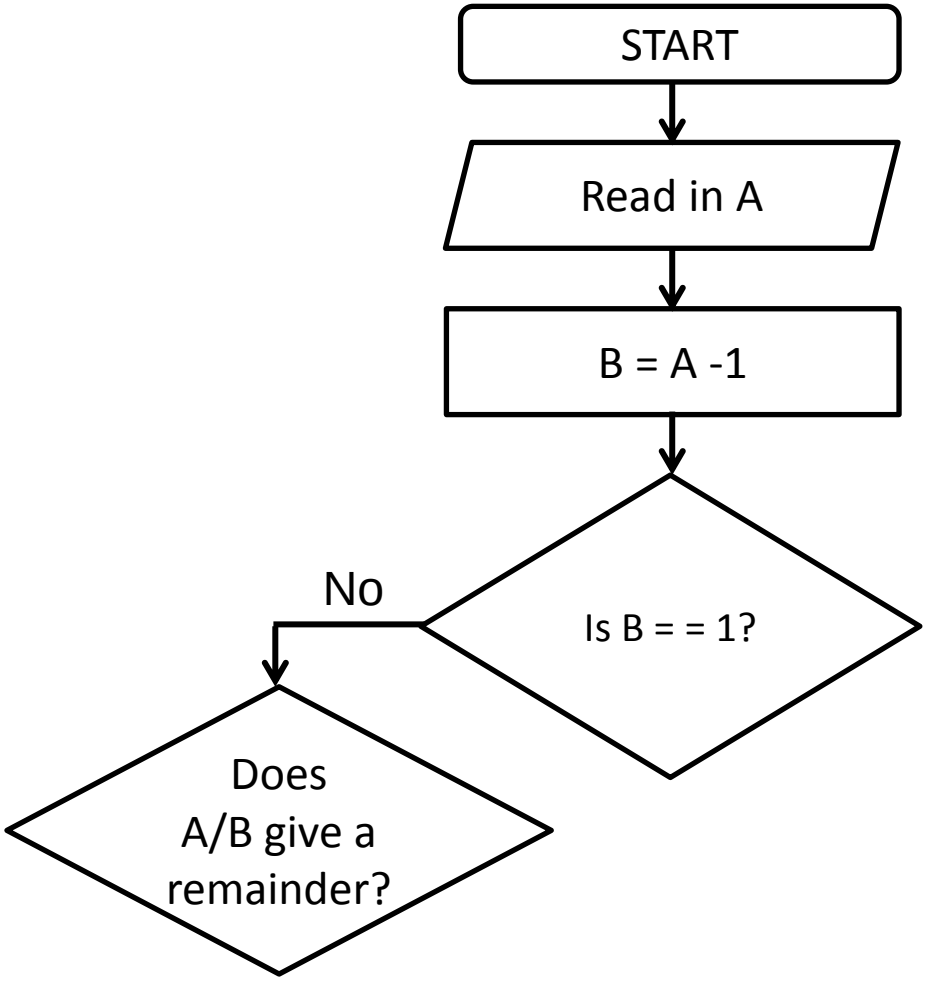

START

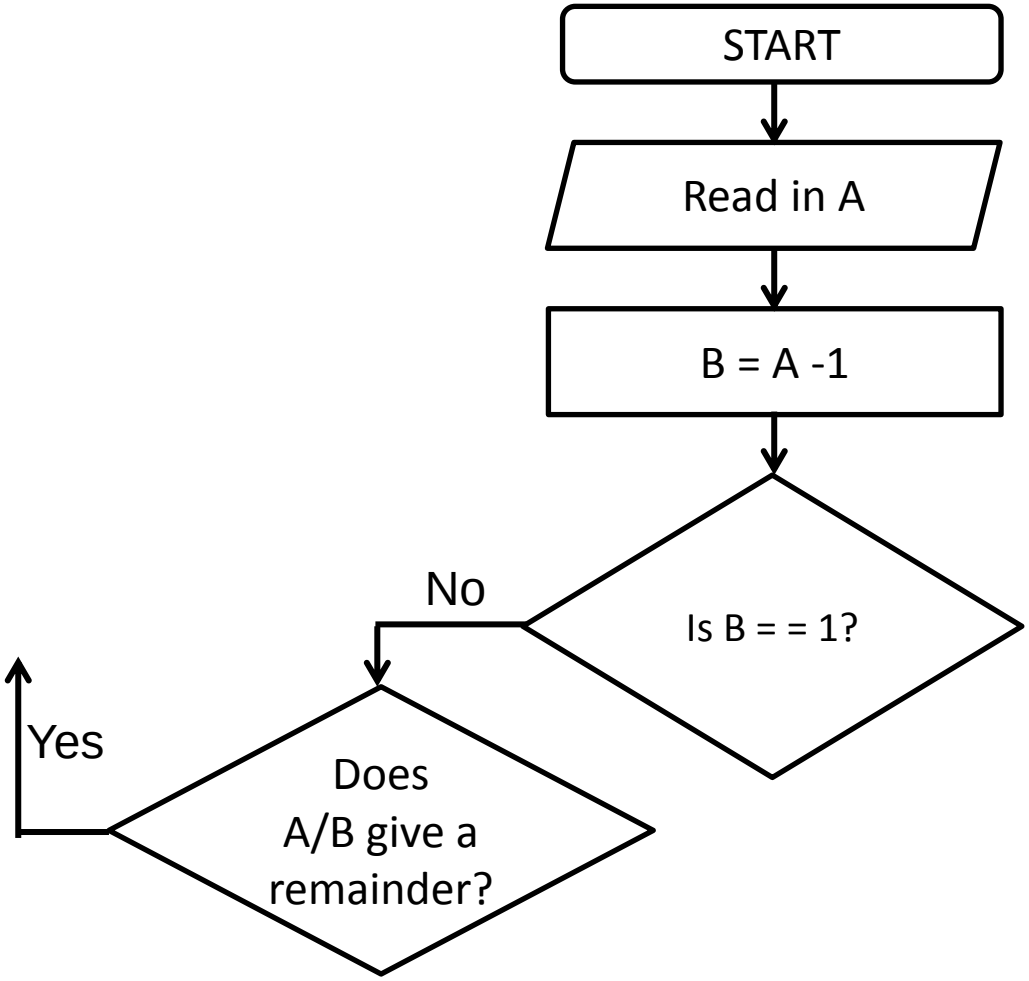

START
Read in A

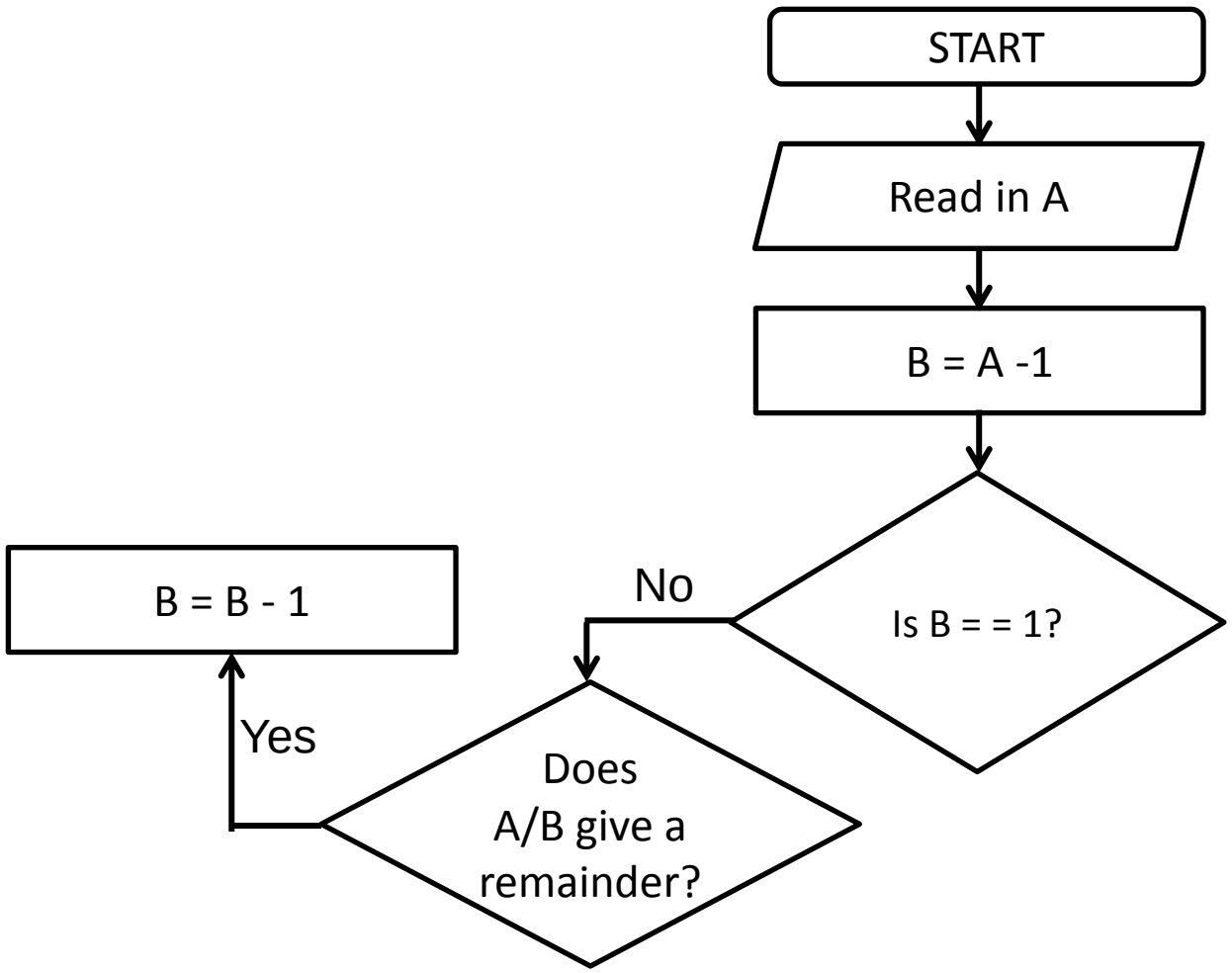

START
Read in A
B = A -1

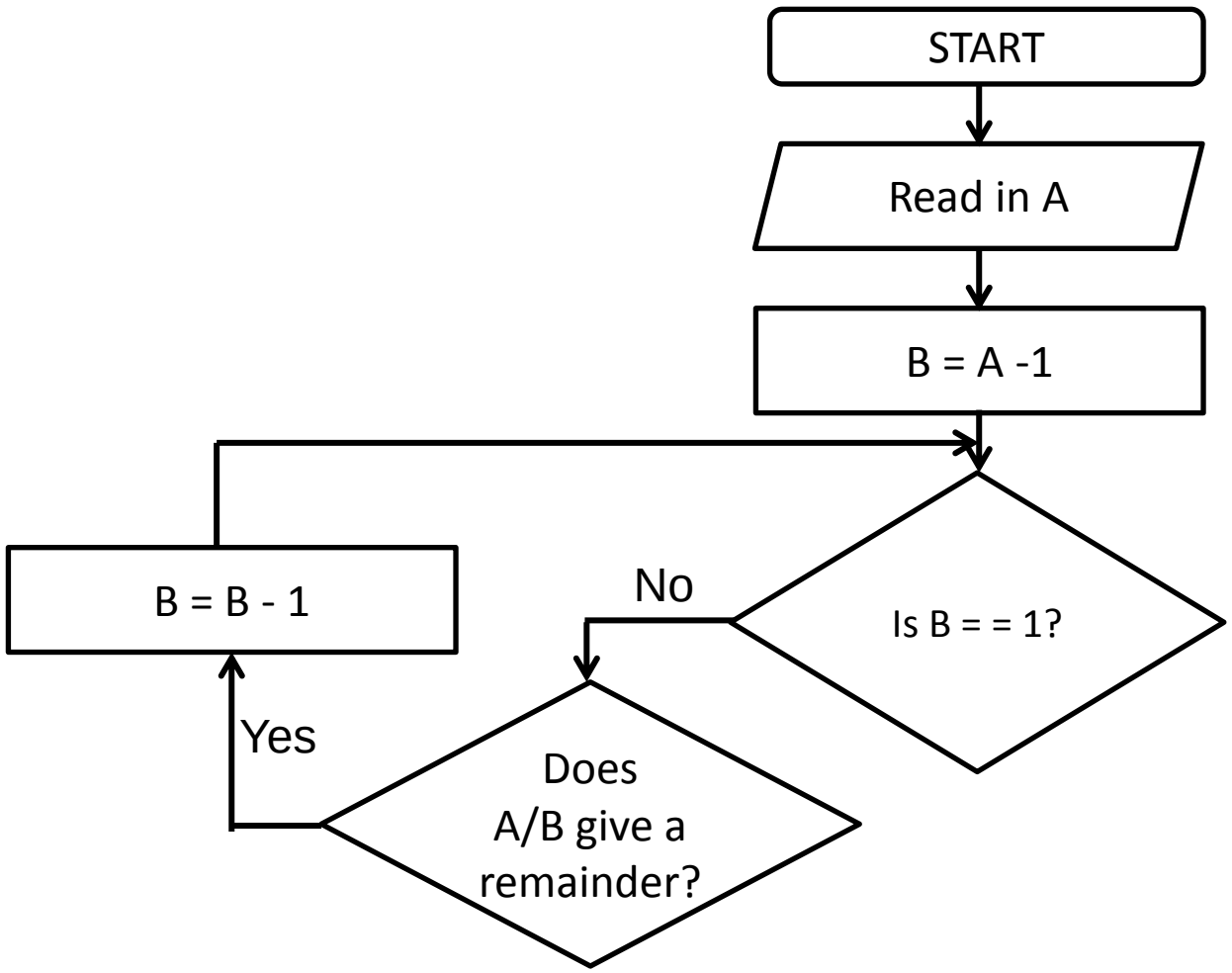

START
Read in A
B = A -1
Is B = = 1?

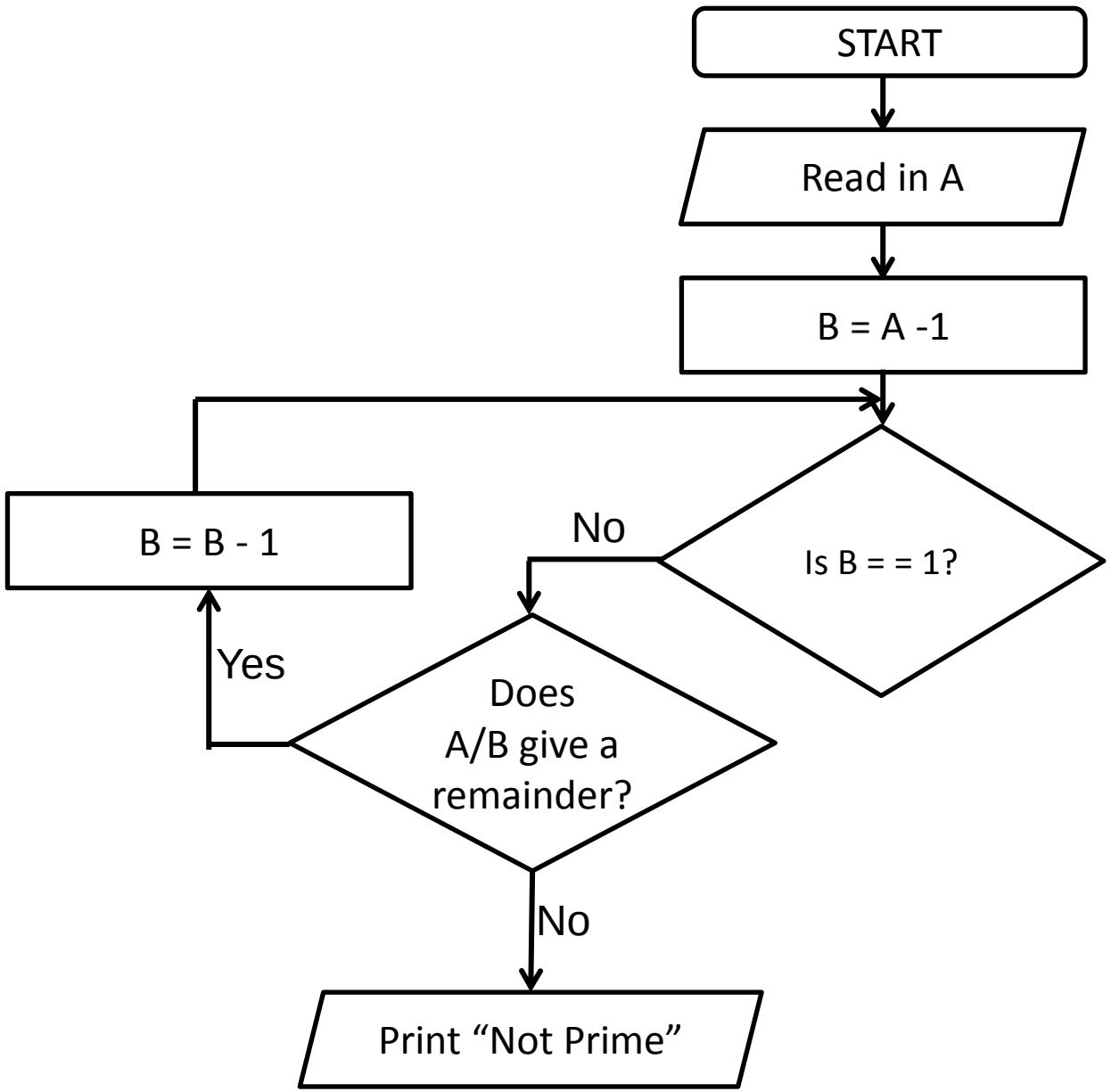

START
Read in A
B = A -1
Is B = = 1?
No

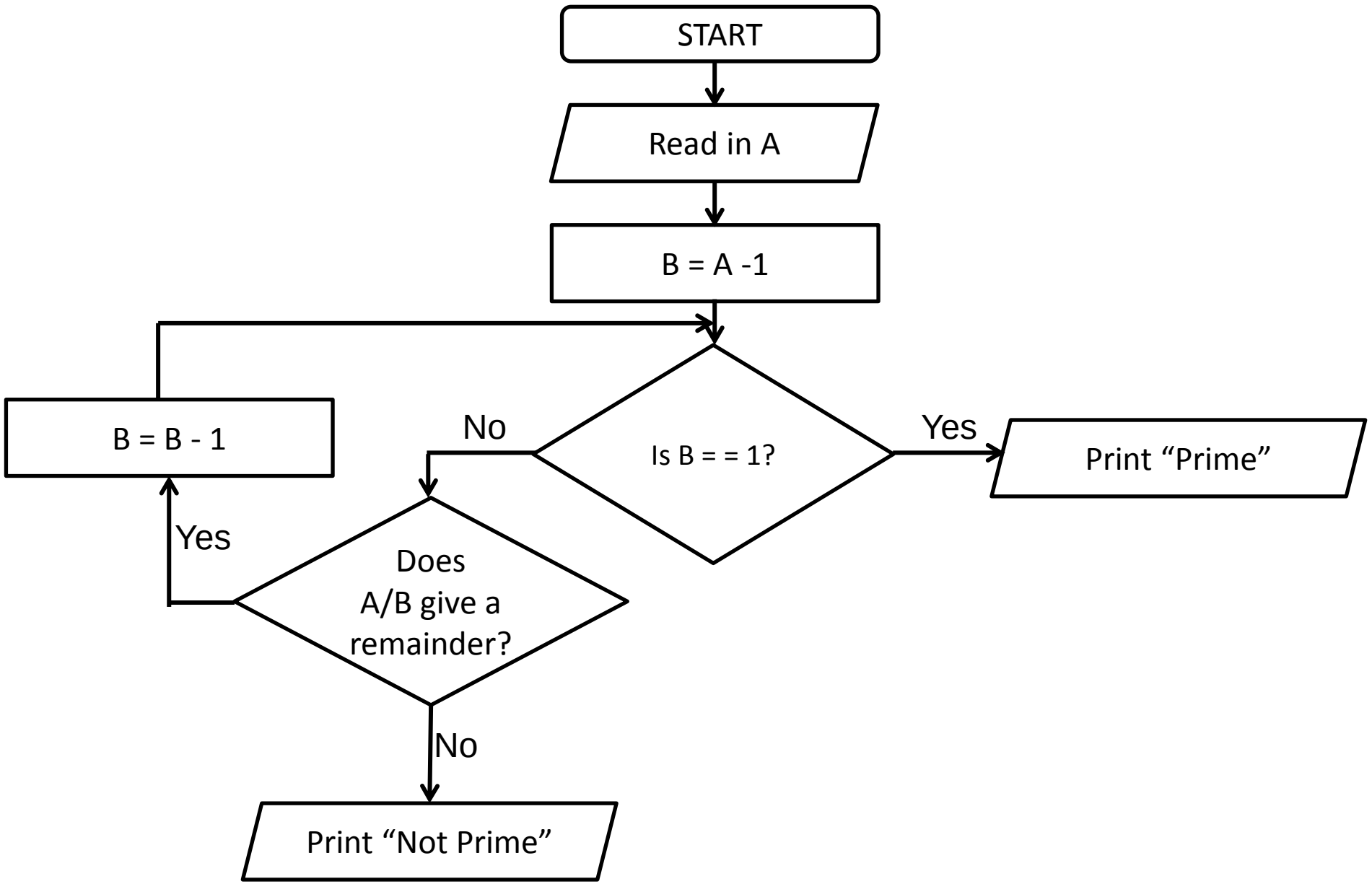

START
Read in A
B = A -1
Is B = = 1?
No
Does
A/B give a
remainder?

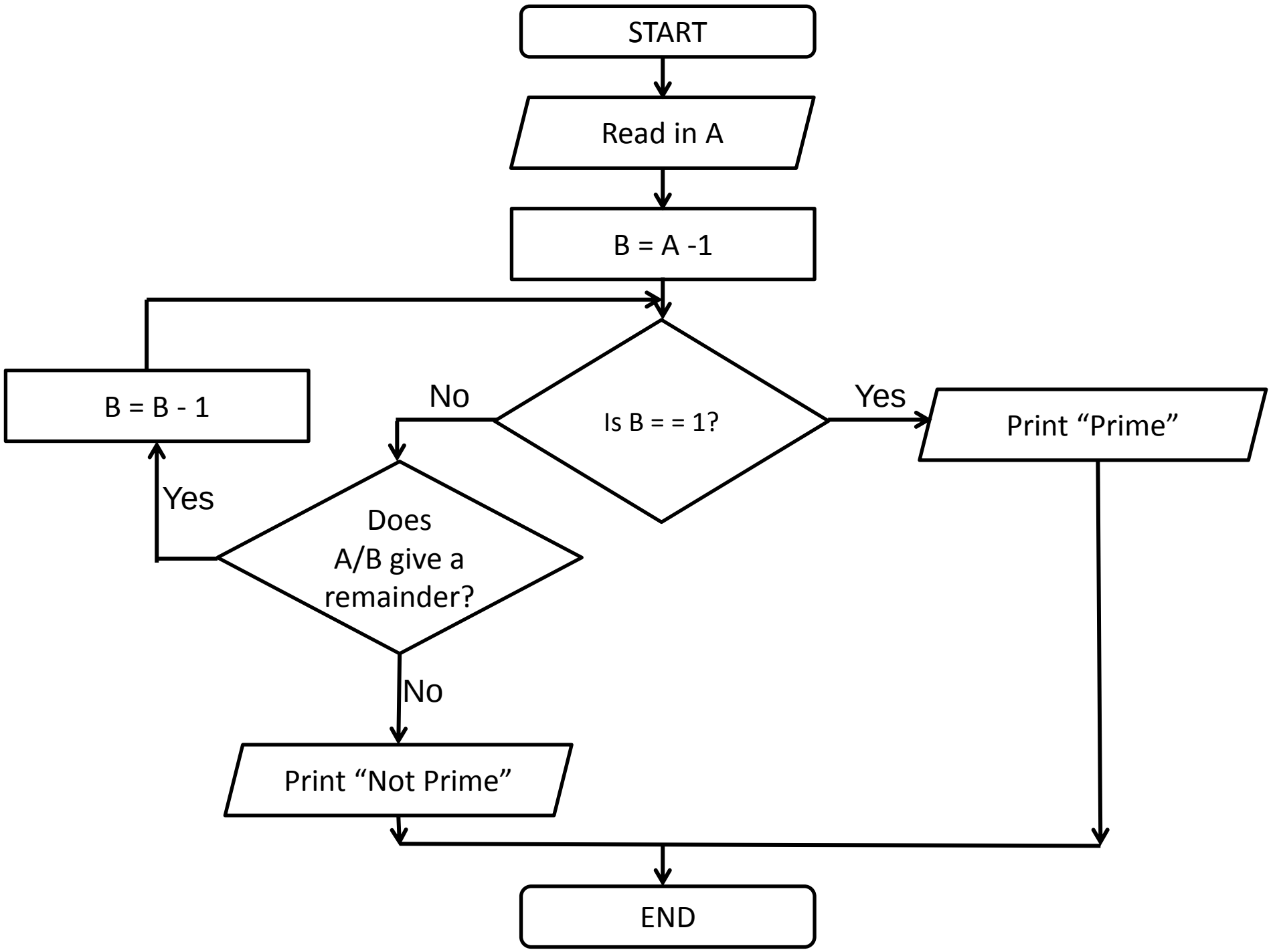

START
Read in A
B = A -1
Is B = = 1?
No
Does
A/B give a
remainder?
Yes


START
Read in A
B = A -1
Is B = = 1?
No
Does
A/B give a
remainder?
Yes
B = B - 1


START
Read in A
B = A -1
Is B = = 1?
No
Does
A/B give a
remainder?
Yes
B = B - 1


START
Read in A
B = A -1
Is B = = 1?
No
B = B - 1
Yes
Does
A/B give a
remainder?
No
Print “Not Prime”


START
Read in A
B = A -1
Is B = = 1?
Yes
Print “Prime”
No
Does
A/B give a
remainder?
Yes
B = B - 1
No
Print “Not Prime”


START
Read in A
B = A -1
Is B = = 1?
No
Yes
Print “Prime”
B = B - 1
Yes
Does
A/B give a
remainder?
No
Print “Not Prime”
END

References

- 2009, Barry, Paul and Griffiths, David; Head First Programming, O'Reilly Media Inc.
- 2009, Pine, Chris ; Learn to Program, 2nd Edition, The Pragmatic Programmers