






## 9. Flowcharts 4

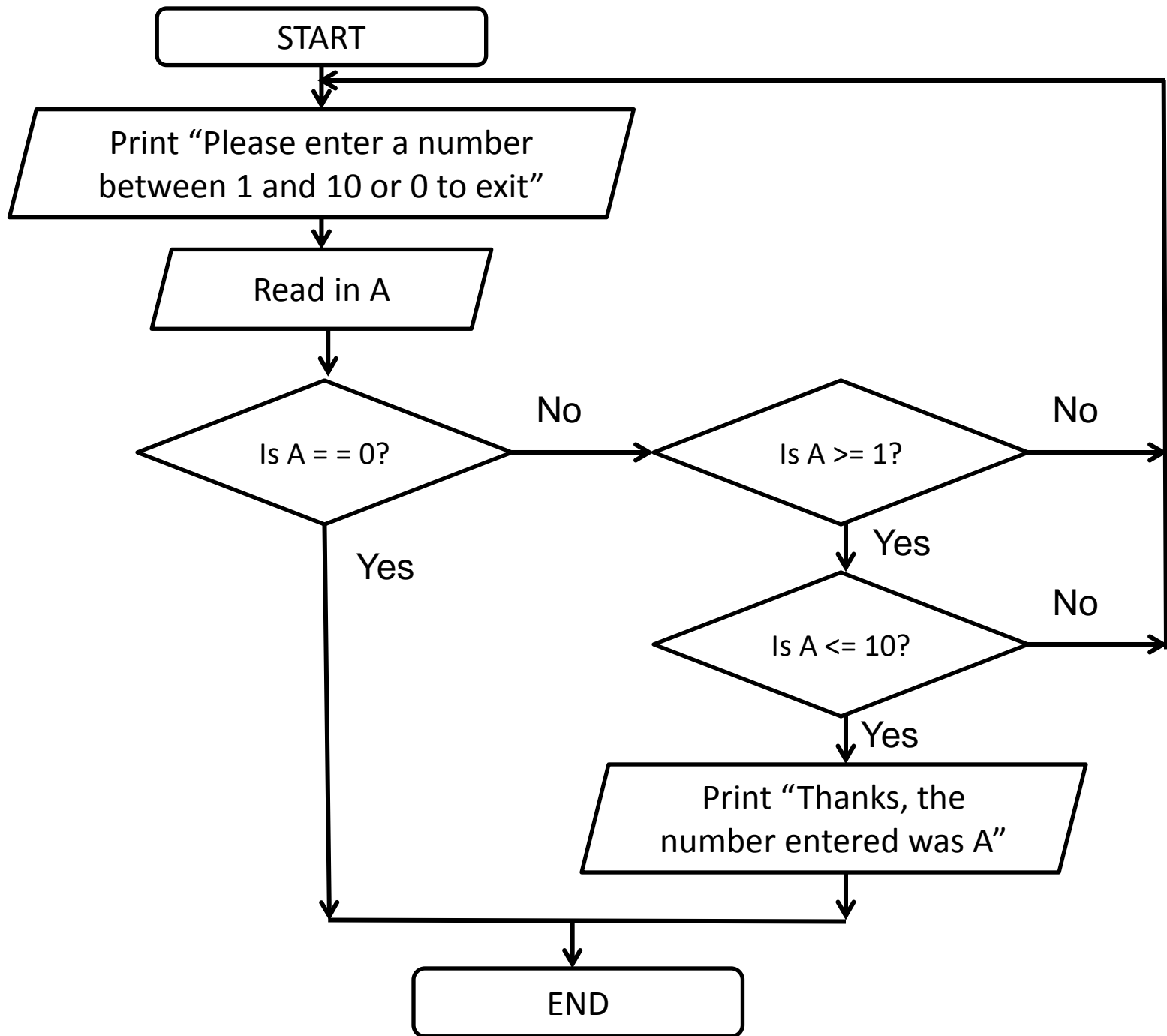
**What did we do last time?**

# Flowcharts – Error Handling (Problem 11)

- Express the following:

*Read in a number, check if the number is between 1 and 10. If not ask again and keep asking. Exit if 0 entered.*






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 output.
	Process	A rectangle represents a process.
	Decision	A diamond indicates a decision.

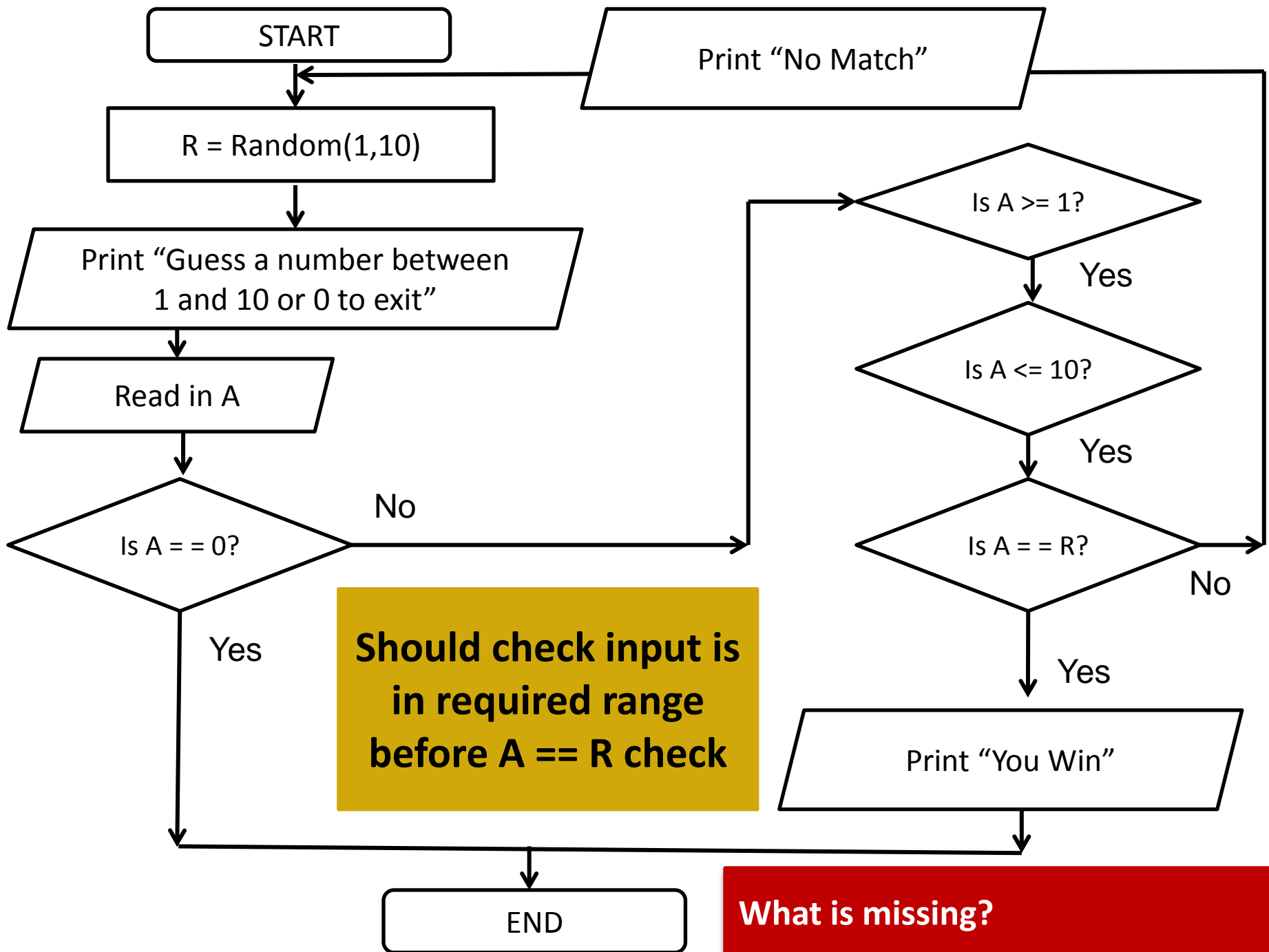


# Flowcharts – Error Handling (Problem 12)

- Express the following:

*Generate a random number between 1 and 10, ask the user to guess the number or enter 0 to exit.*






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 output.
	Process	A rectangle represents a process.
	Decision	A diamond indicates a decision.

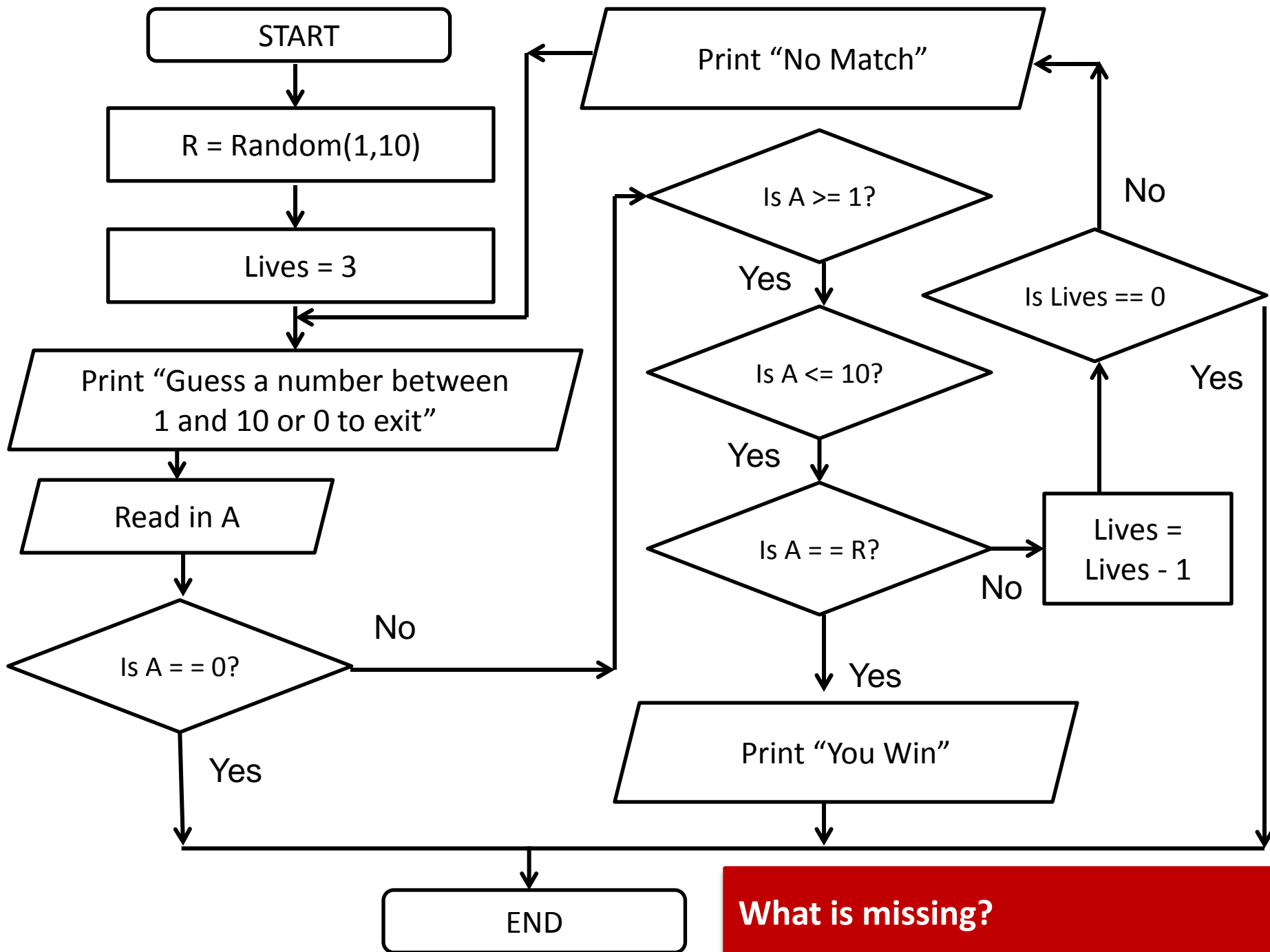


# Flowcharts – Guessing (Problem 13)

- Express the following:

*Generate a random number between 1 and 10, ask the user to guess the number or enter 0 to exit. Let the user have three chances to guess the 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 output.
	Process	A rectangle represents a process.
	Decision	A diamond indicates a decision.








**What is missing?**

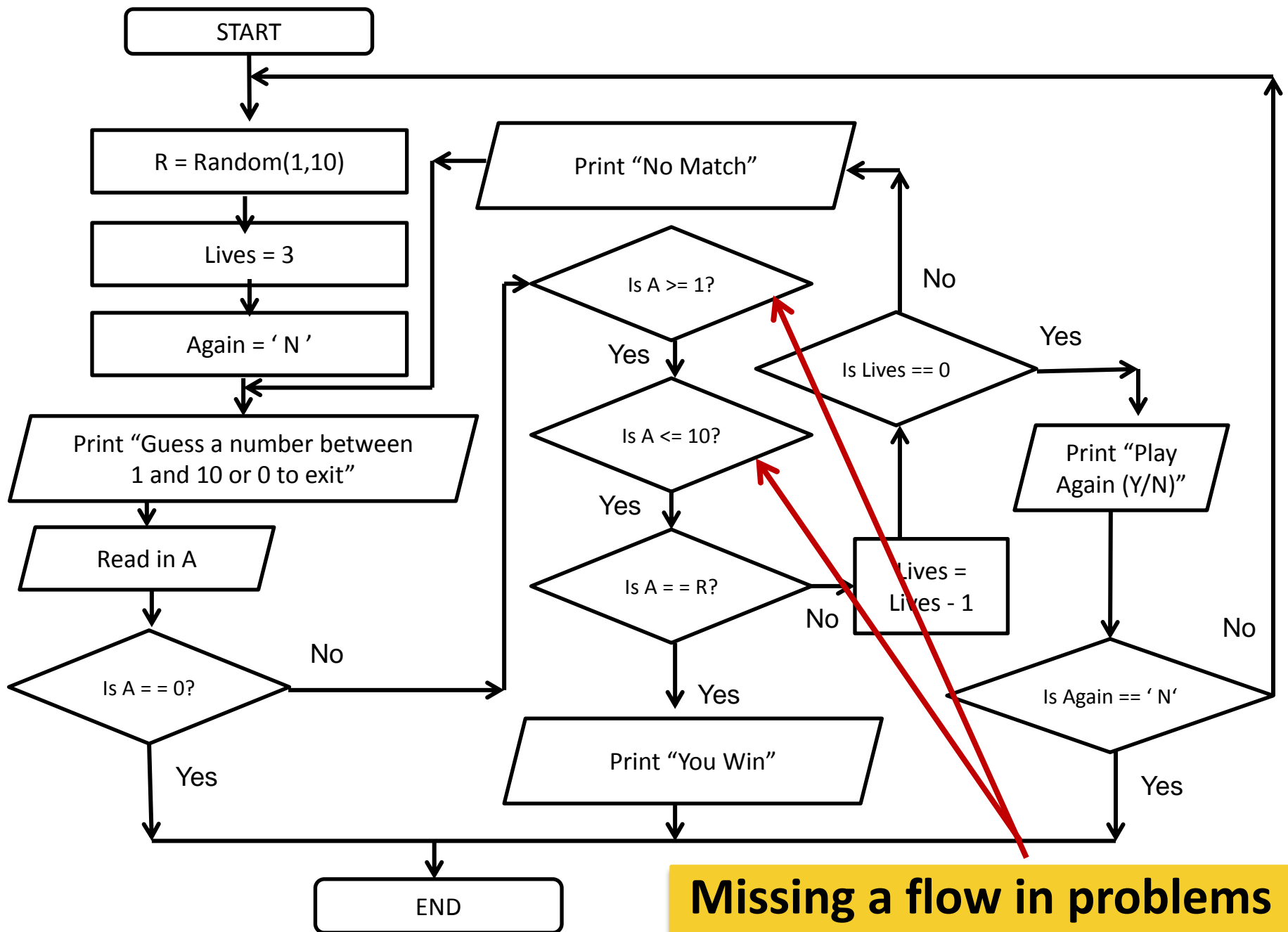


# Flowcharts – Guessing (Problem 14)

- Express the following:

*Generate a random number between 1 and 10, ask the user to guess the number or enter 0 to exit. Let the user have three chances to guess the number. **After 3 chances ask if the user wants to play again rather than just exit.***

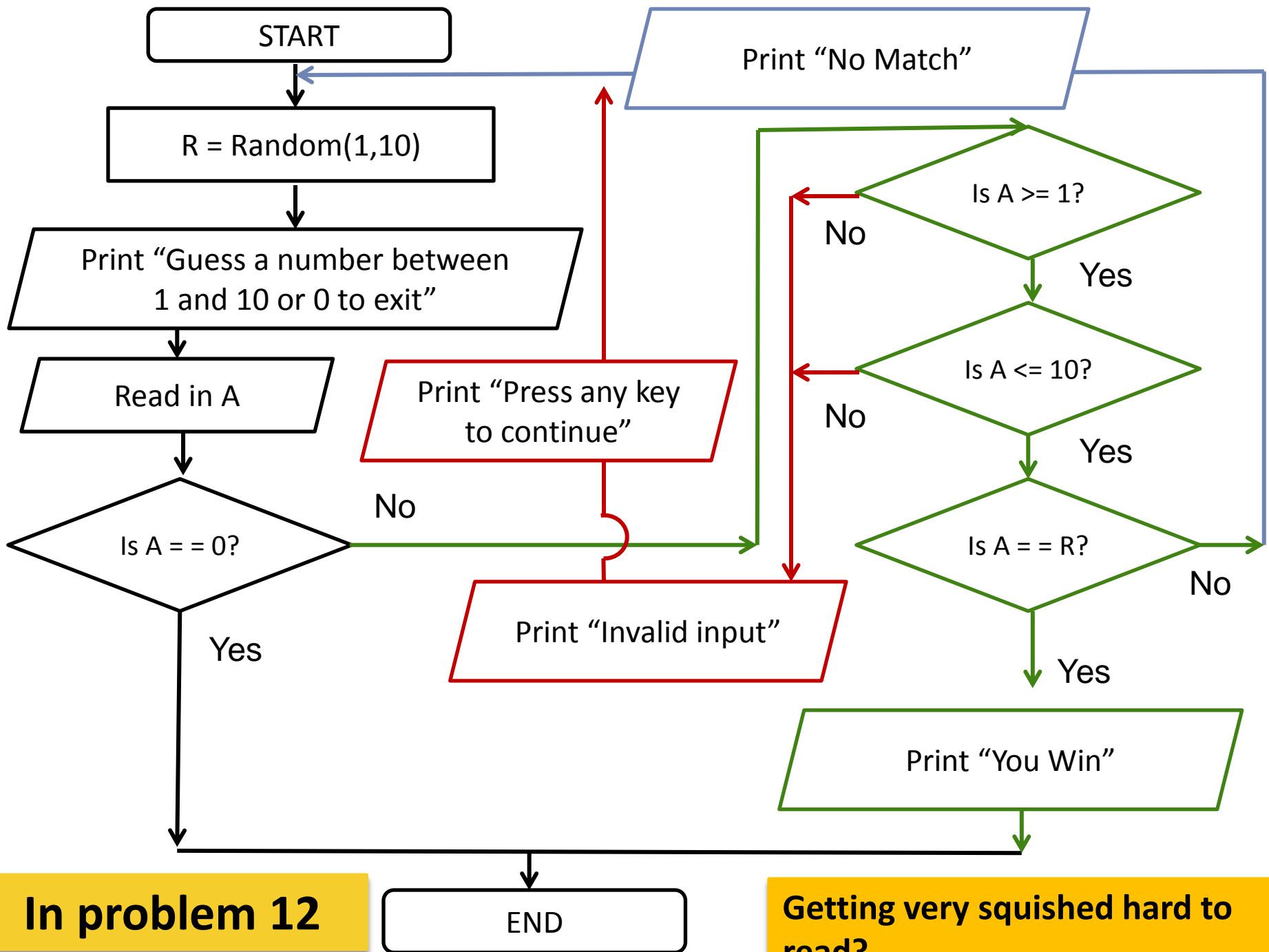
Symbol	Name	Function
	Start/end	An oval represents a start or end point.
	Arrows	A line is a connector that shows relationships between the representative shapes.
	Input/Output	A parallelogram represents input or output.
	Process	A rectangle represents a process.
	Decision	A diamond indicates a decision.



**Missing a flow in problems  
12, 13, and 14**

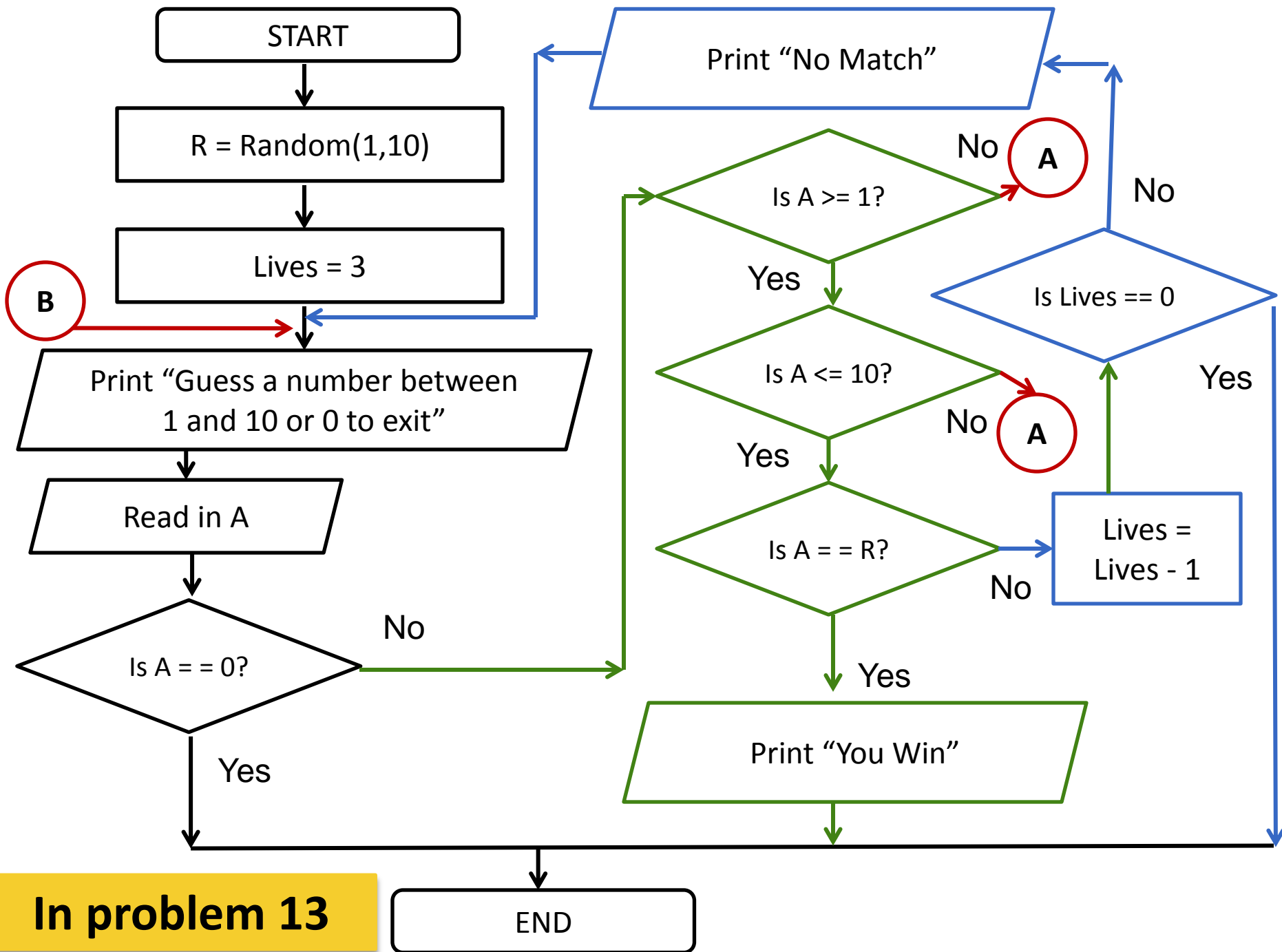
Program Design

# **ERRORS AND CONNECTORS**

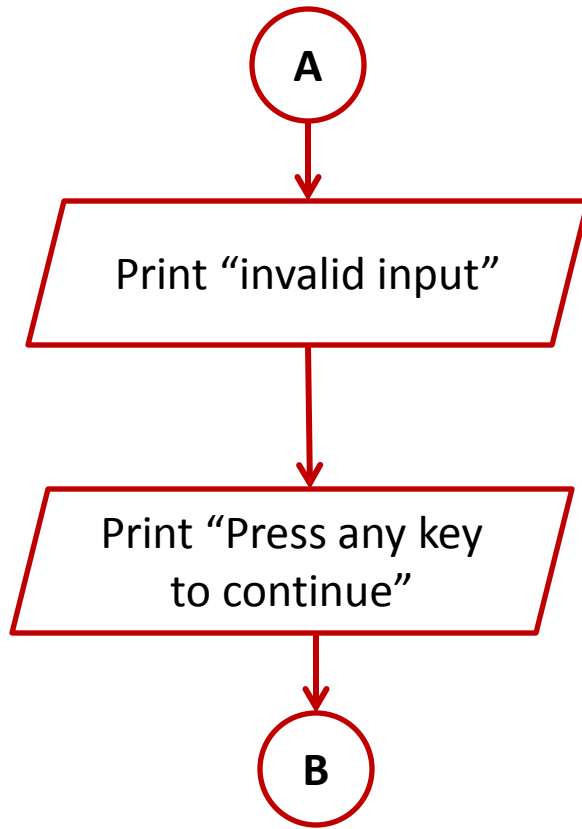


**In problem 12**

**Getting very squished hard to read?**



**In problem 13**



**This is like a module or function**  
**A piece of code that is used over and over again from multiple places in the code**

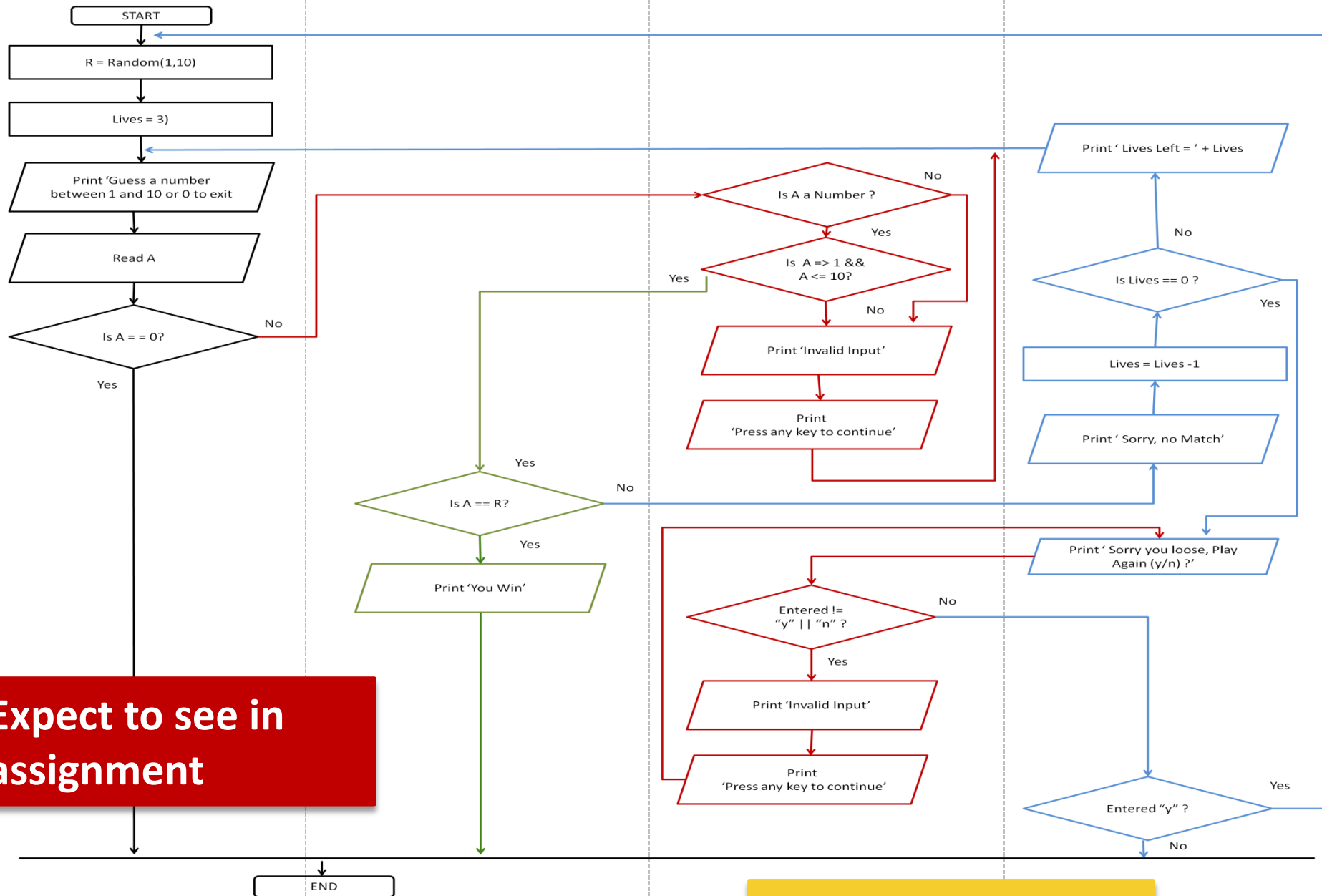
**Also a good way to split things down if you start from a summary flowchart and start expanding**

## Getting Started

## Running

## Checking & Error

## Keep Running ?



**Expect to see in  
assignment**

**In problem 14**

# Today's Lecture



Program Design

# **DATA TYPES, ARRAYS AND SEARCHING FOR A CHARACTER**

# Data Types in Programs (types of variables)

## NUMBERS

### Integers

100, 56, 9000000, 256,

### Decimal \$,£,€

3.14, 6.626068 , 100.01

*int, float, decimal,  
numeric, tinyint,  
smallint, bigint, money,  
real*

## TEXT

### Character

'A', 'C', 'h', 'a', 'r'

### String "a text string"

*char(n), varchar(n),  
text, nchar(n),  
nvarchar(n), ntext*

## DATES

### DateTime

2010-12-01

### TimeStamp

2010-12-01 23:55:10

*datetime, timestamp,  
date, time*

## BINARY

### Bit

1 , 0

### Binary

100011100011101010

*bit,  
binary(n), varbinary(n),  
image*

## BOOLEAN

### Boolean True, false

*boolean*

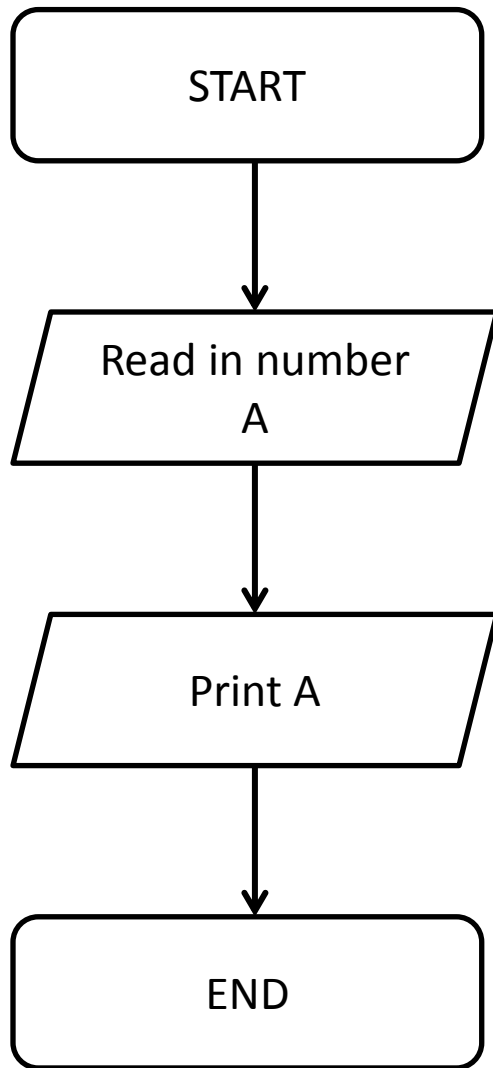
**NULL** and **void** are value  
type in some languages  
= undefined, value unknown

# Everything is stored as binary numbers inside the computer – the ASCII Table

DEC	OCT	HEX	BIN	Symbol	HTML Number	HTML Name	Description
32	040	20	00100000		&#32;		Space
33	041	21	00100001	!	&#33;		Exclamation mark
34	042	22	00100010	"	&#34;	&quot;	Double quotes (or speech marks)
35	043	23	00100011	#	&#35;		Number
36	044	24	00100100	\$	&#36;		Dollar
37	045	25	00100101	%	&#37;		Procenttecken
38	046	26	00100110	&	&#38;	&amp;	Ampersand
39	047	27	00100111	'	&#39;		Single quote

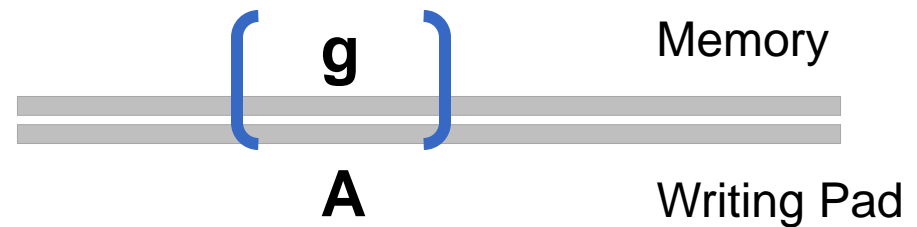
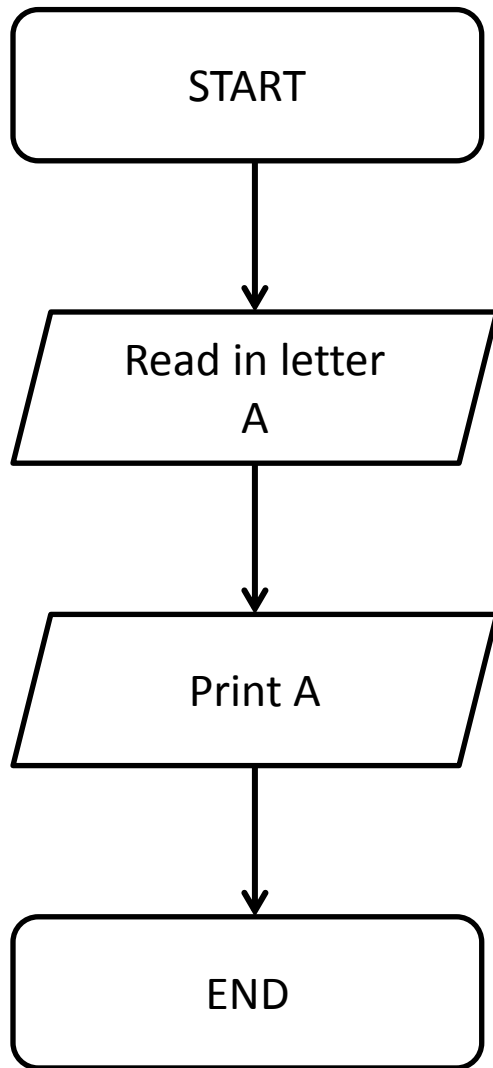
65	101	41	01000001	A	&#65;		Uppercase A
66	102	42	01000010	B	&#66;		Uppercase B
67	103	43	01000011	C	&#67;		Uppercase C
68	104	44	01000100	D	&#68;		Uppercase D
69	105	45	01000101	E	&#69;		Uppercase E
70	106	46	01000110	F	&#70;		Uppercase F
71	107	47	01000111	G	&#71;		Uppercase G
72	110	48	01001000	H	&#72;		Uppercase H

97	141	61	01100001	a	&#97;		Lowercase a
98	142	62	01100010	b	&#98;		Lowercase b
99	143	63	01100011	c	&#99;		Lowercase c
100	144	64	01100100	d	&#100;		Lowercase d
101	145	65	01100101	e	&#101;		Lowercase e
102	146	66	01100110	f	&#102;		Lowercase f



Type : *integer*

A basic variable holds only one value of a particular type



Type : ***character***

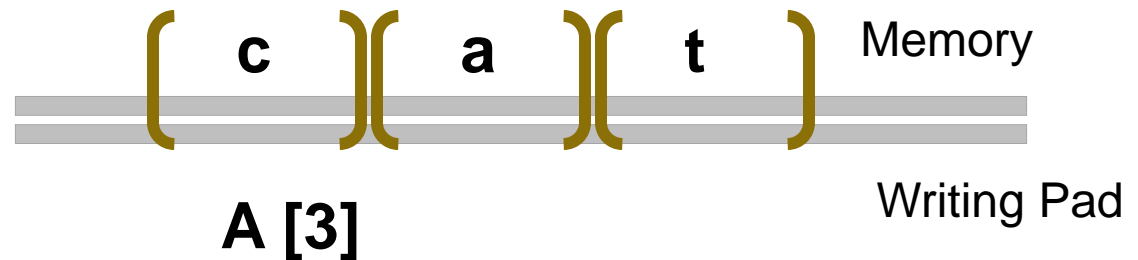
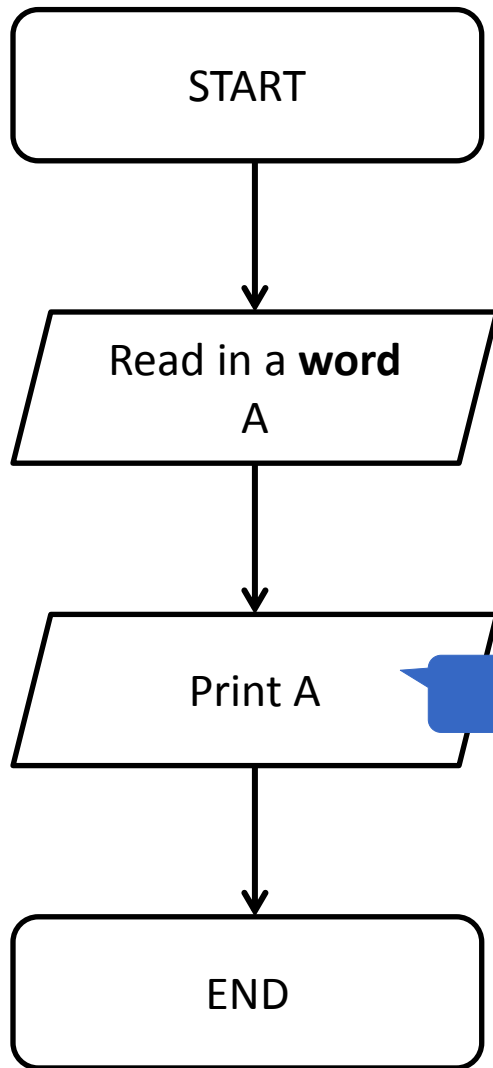
A basic variable holds only one value of a particular type

If we wanted to read in a word  
how should we store it ?

# Arrays

- An array is a **variable** that **can contain more than one value**
- But the values must all be of the **same type**
- An egg box in memory, with a name
- Or a **list** in memory





How?

Type : **character array** = **string**

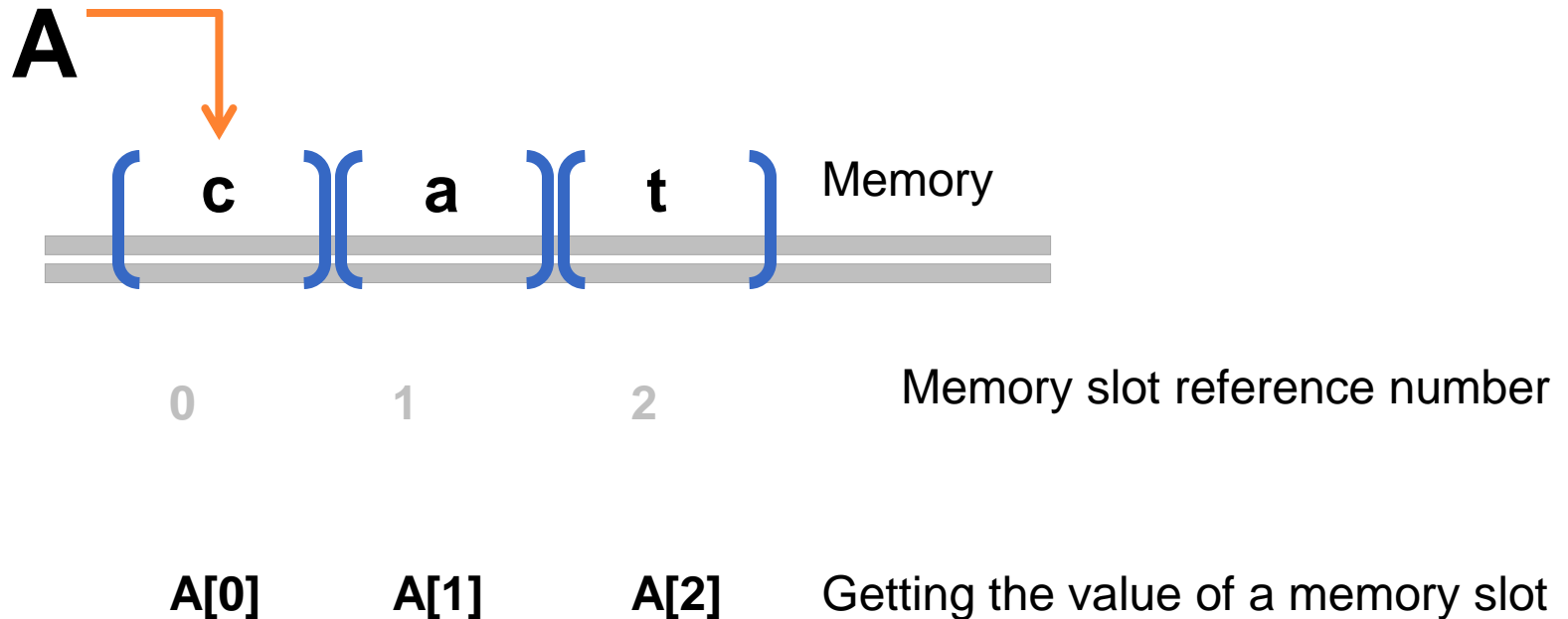
An array variable holds multiple value of the same type.

A name chain of values



# Elements in an array

*The string variable*








Type : **character array = string**

# Flowcharts (Problem 15)

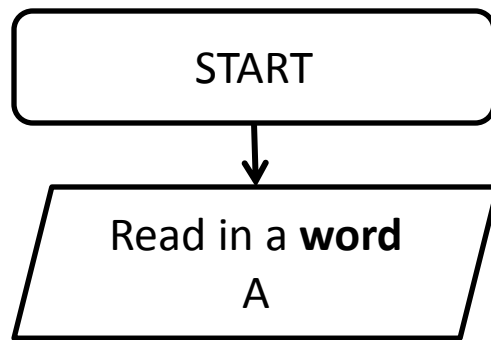
- So let's say we want to express the following algorithm:

*Print out each character in the word "cat".*

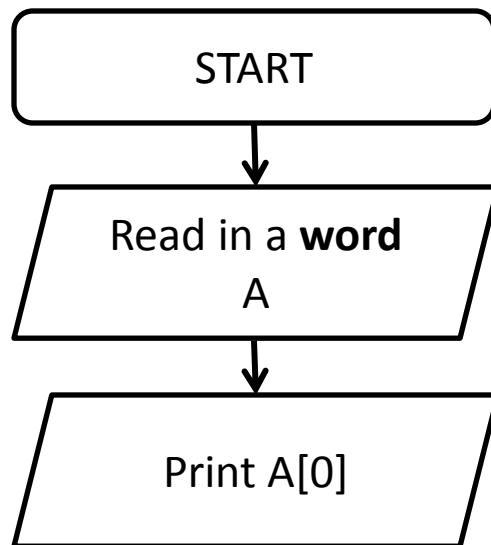
Symbol	Name	Function
	Start/end	An oval represents a start or end point.
	Arrows	A line is a connector that shows relationships between the representative shapes.
	Input/Output	A parallelogram represents input or output.
	Process	A rectangle represents a process.
	Decision	A diamond indicates a decision.

START

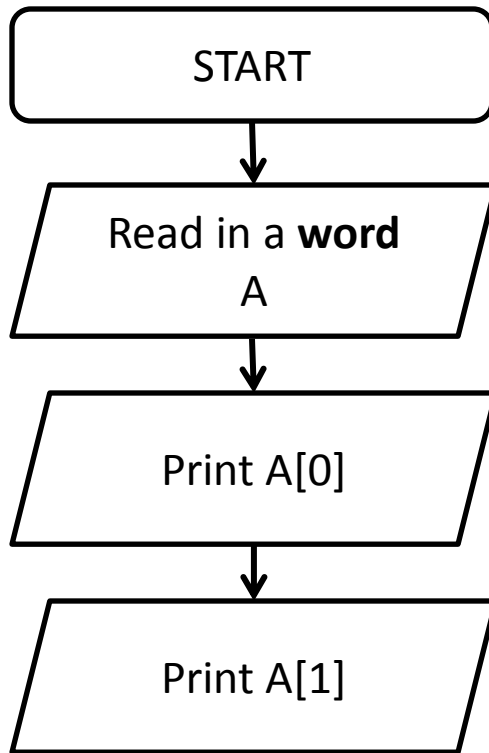




0 1 2  
( c )( a )( t )



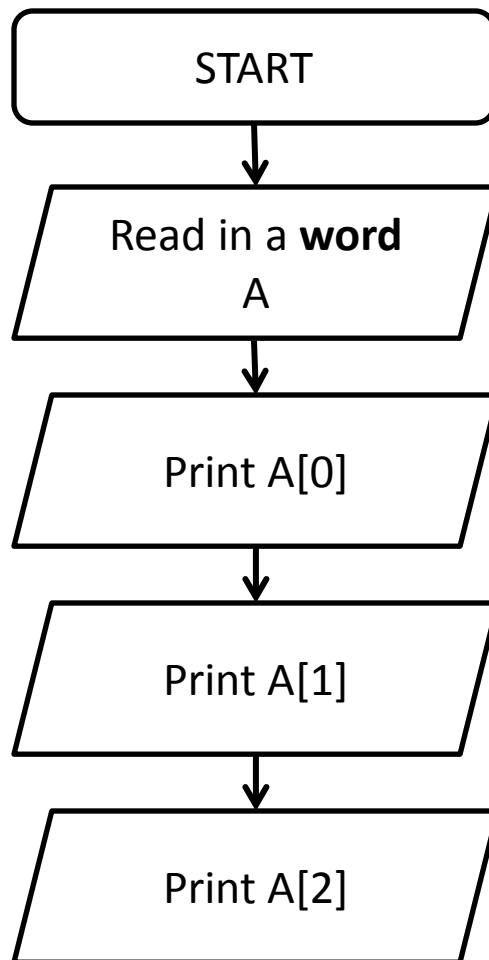
0 1 2  
( c ) ( a ) ( t )  
c



0 1 2  
( c ) ( a ) ( t )

c

a

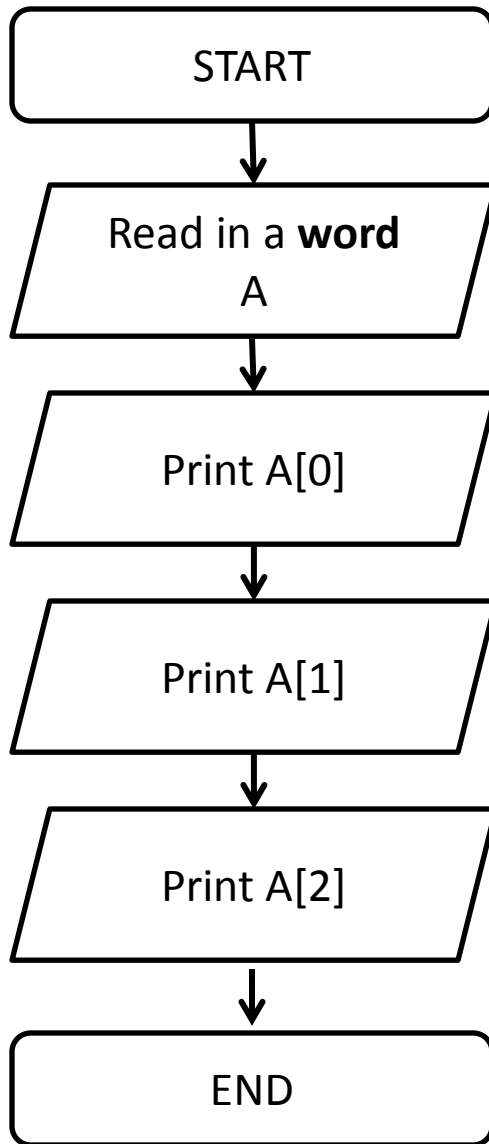


0 1 2  
( c ) ( a ) ( t )

c

a

t



0 1 2  
( c ) ( a ) ( t )

c

a






t

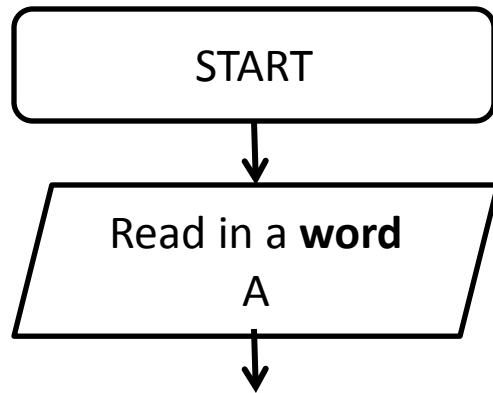


# Flowcharts (Problem 16)

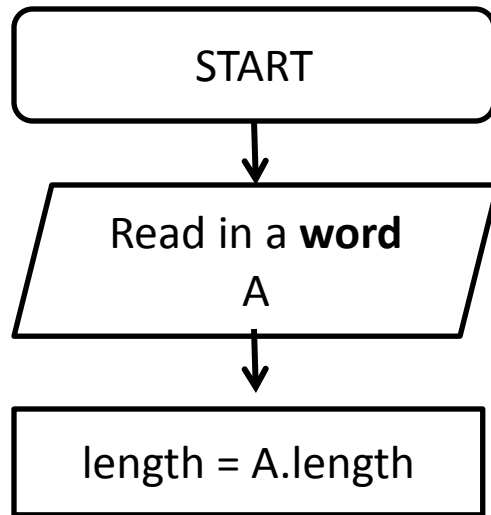
- So let's say we want to express the following algorithm:

*Print out each character in any word that is entered.*

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 output.
	Process	A rectangle represents a process.
	Decision	A diamond indicates a decision.

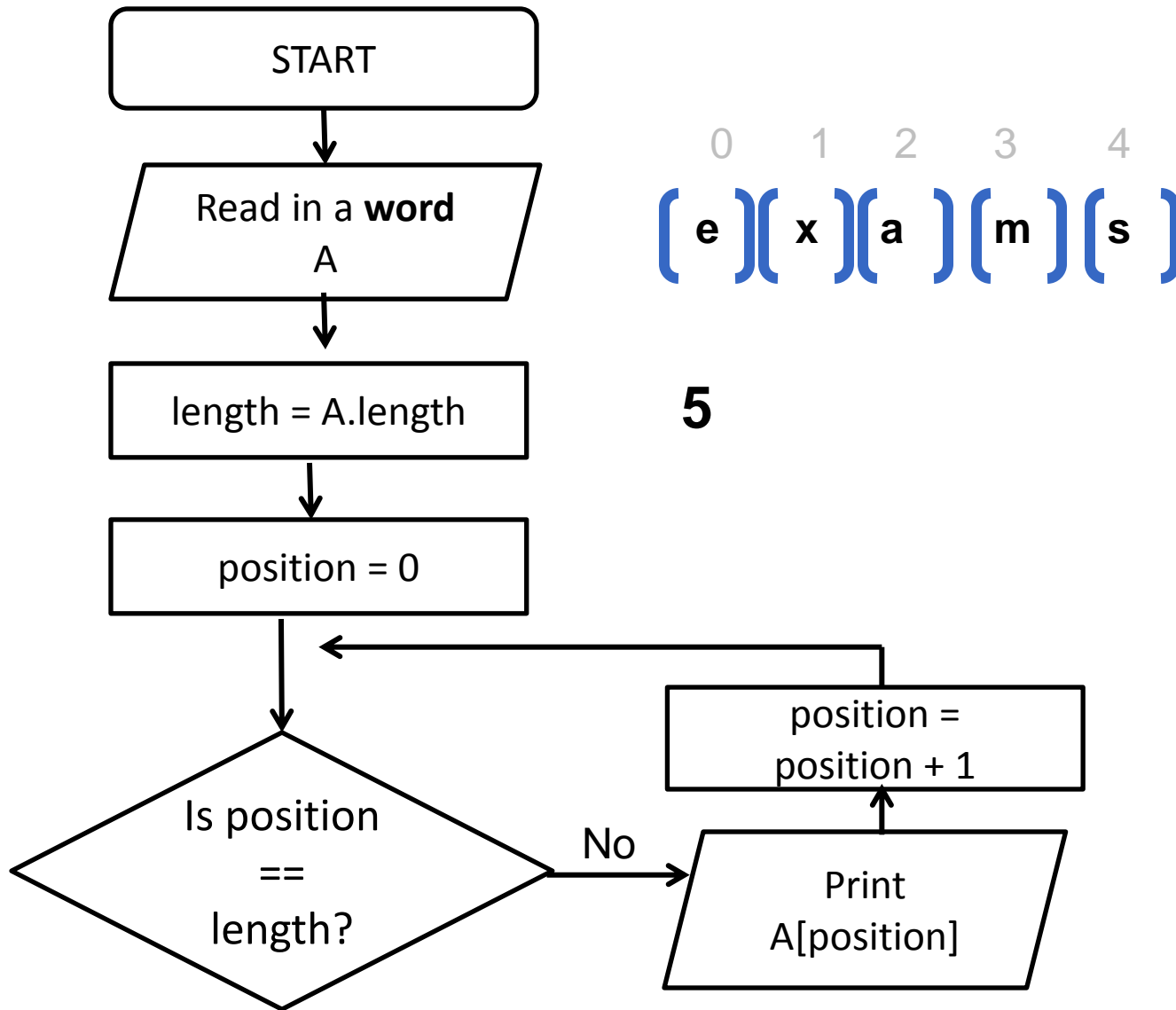


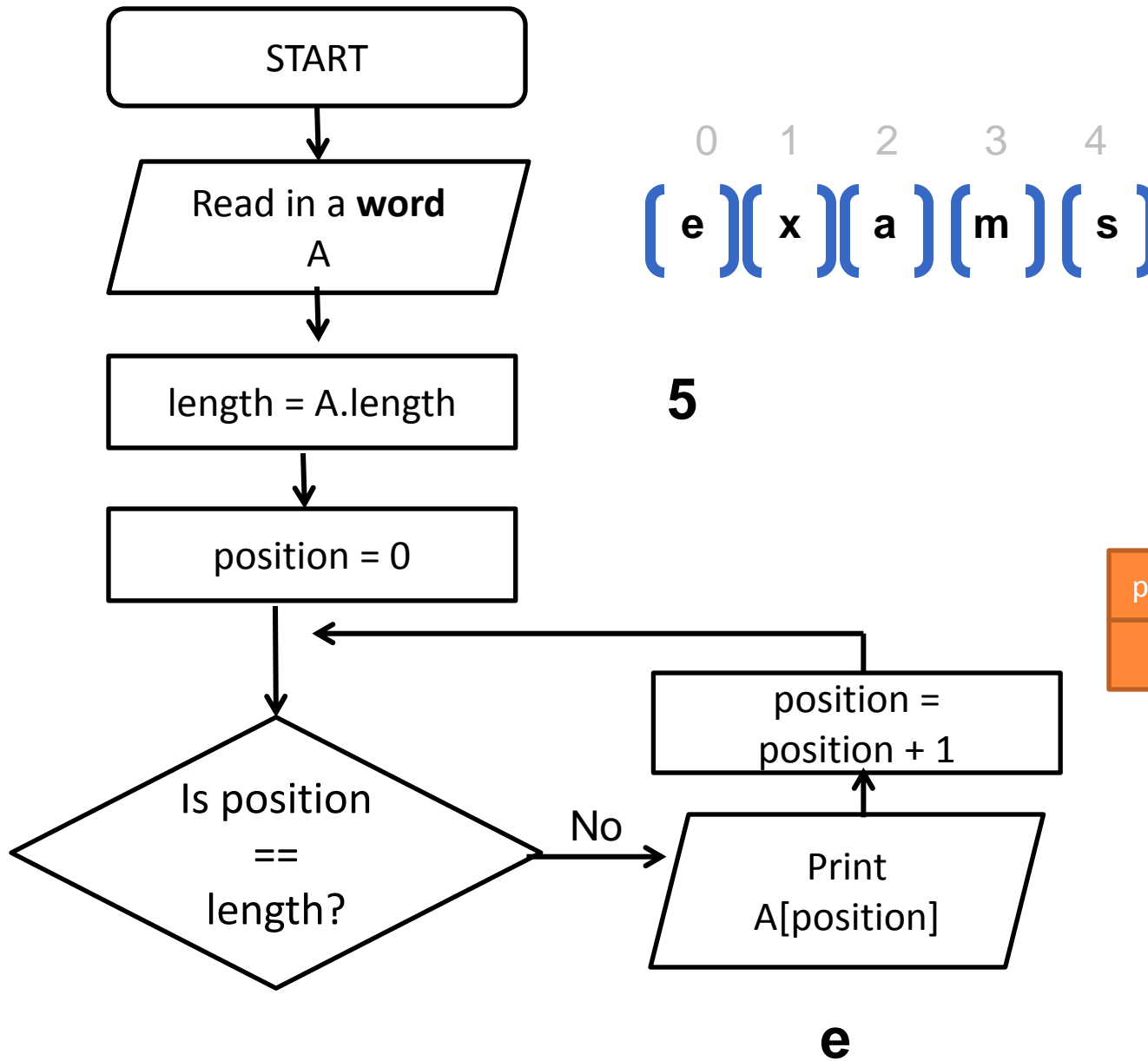
0 1 2 3 4  
( e )( x )( a )( m )( s )



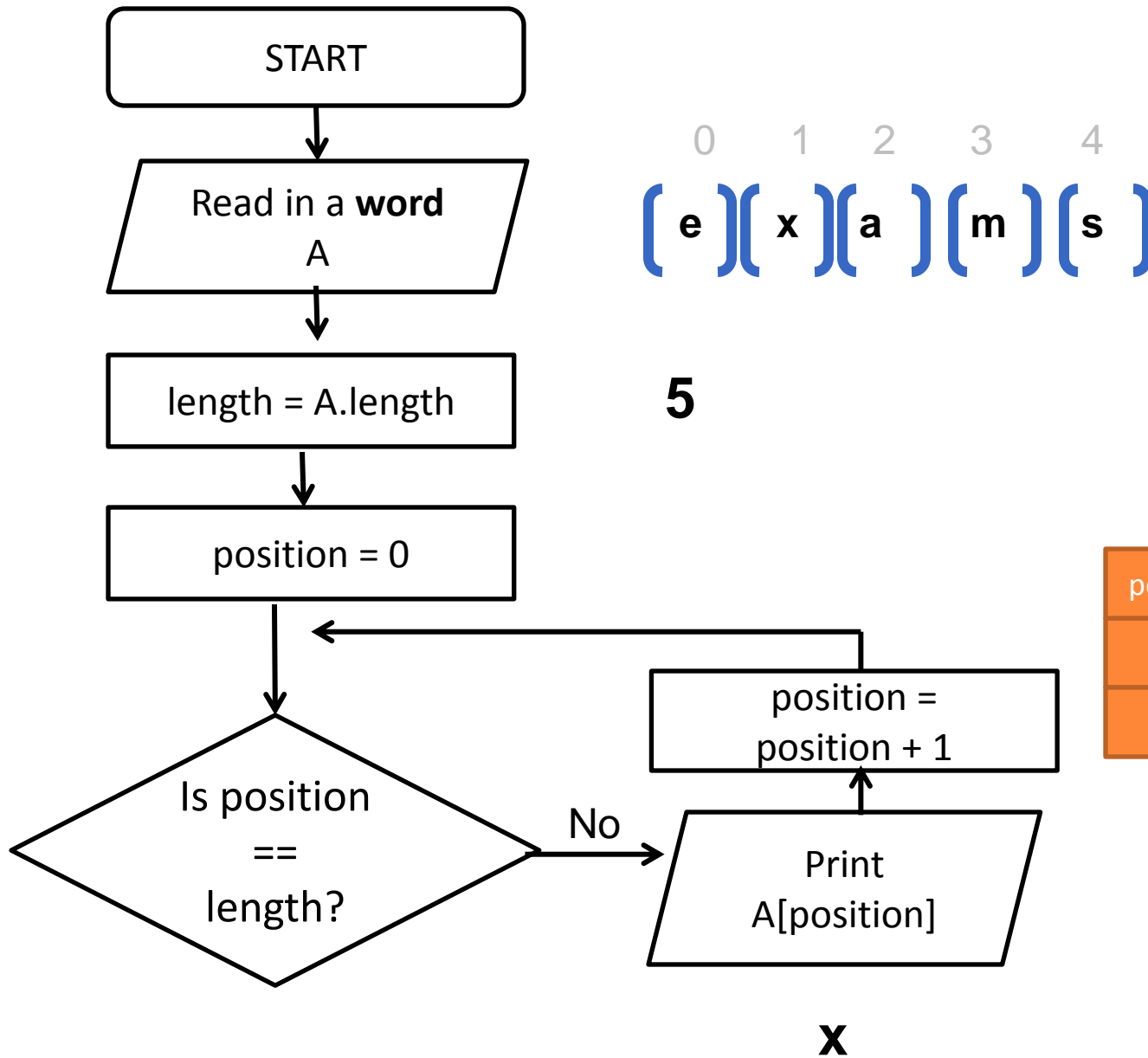
0 1 2 3 4  
( e )( x )( a )( m )( s )

5

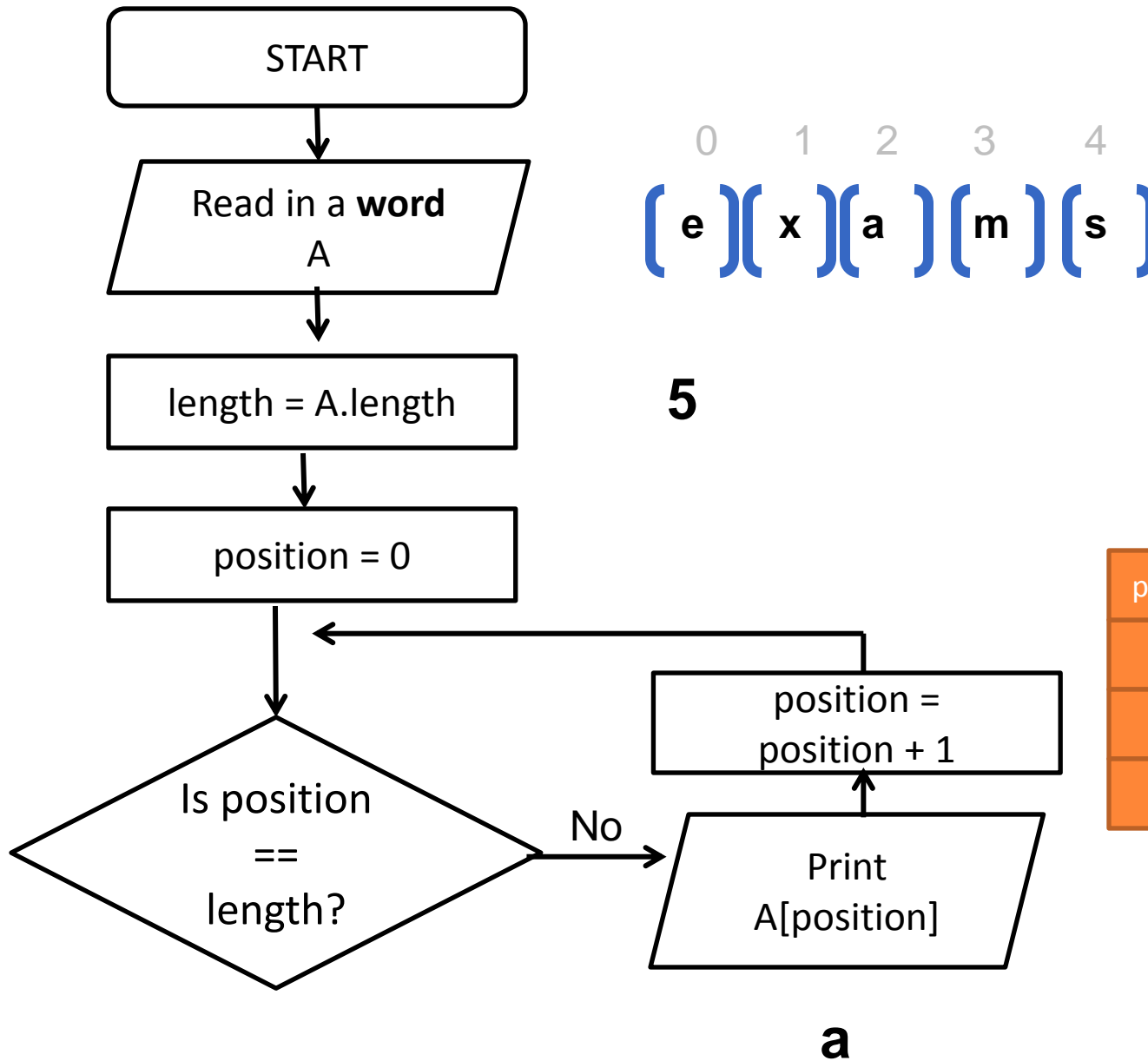




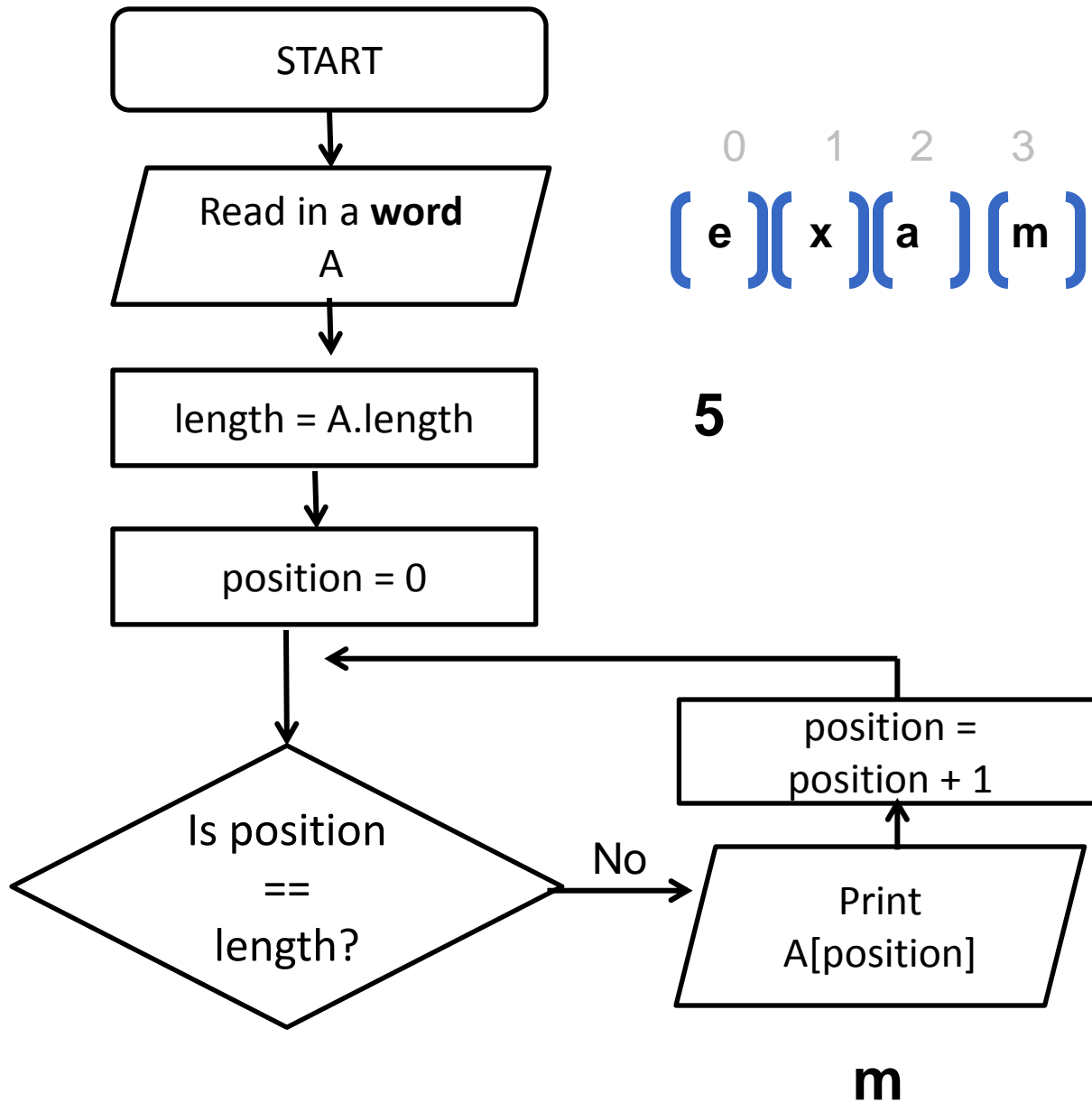
position	Array ref	value
0	A[0]	e



position	Array ref	value
0	A[0]	e
1	A[1]	x



position	Array ref	value
0	A[0]	e
1	A[1]	x
2	A[2]	a

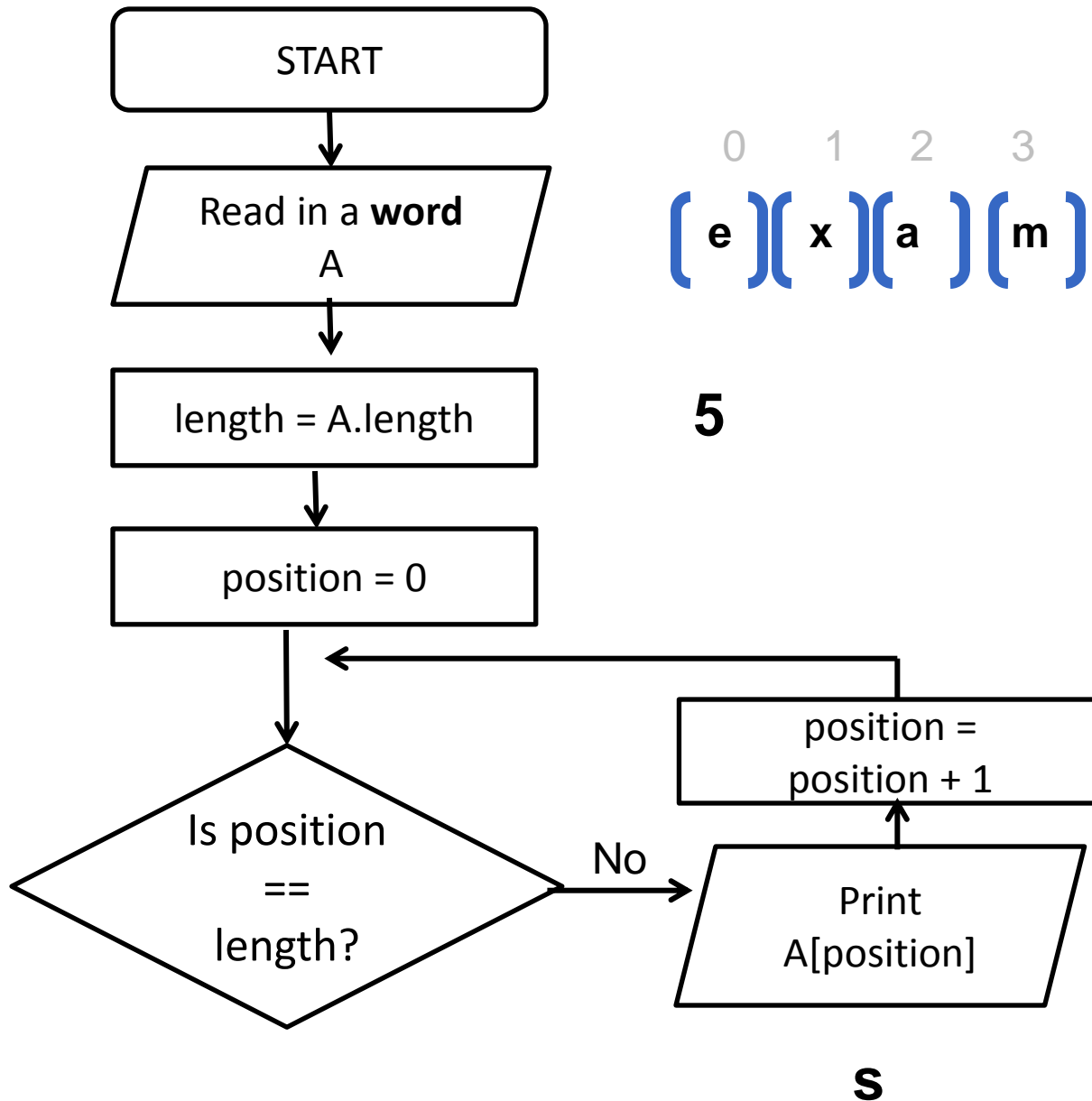


0 1 2 3 4  
( e )( x )( a )( m )( s )

5

position	Array ref	value
0	A[0]	e
1	A[1]	x
2	A[2]	a
3	A[3]	m



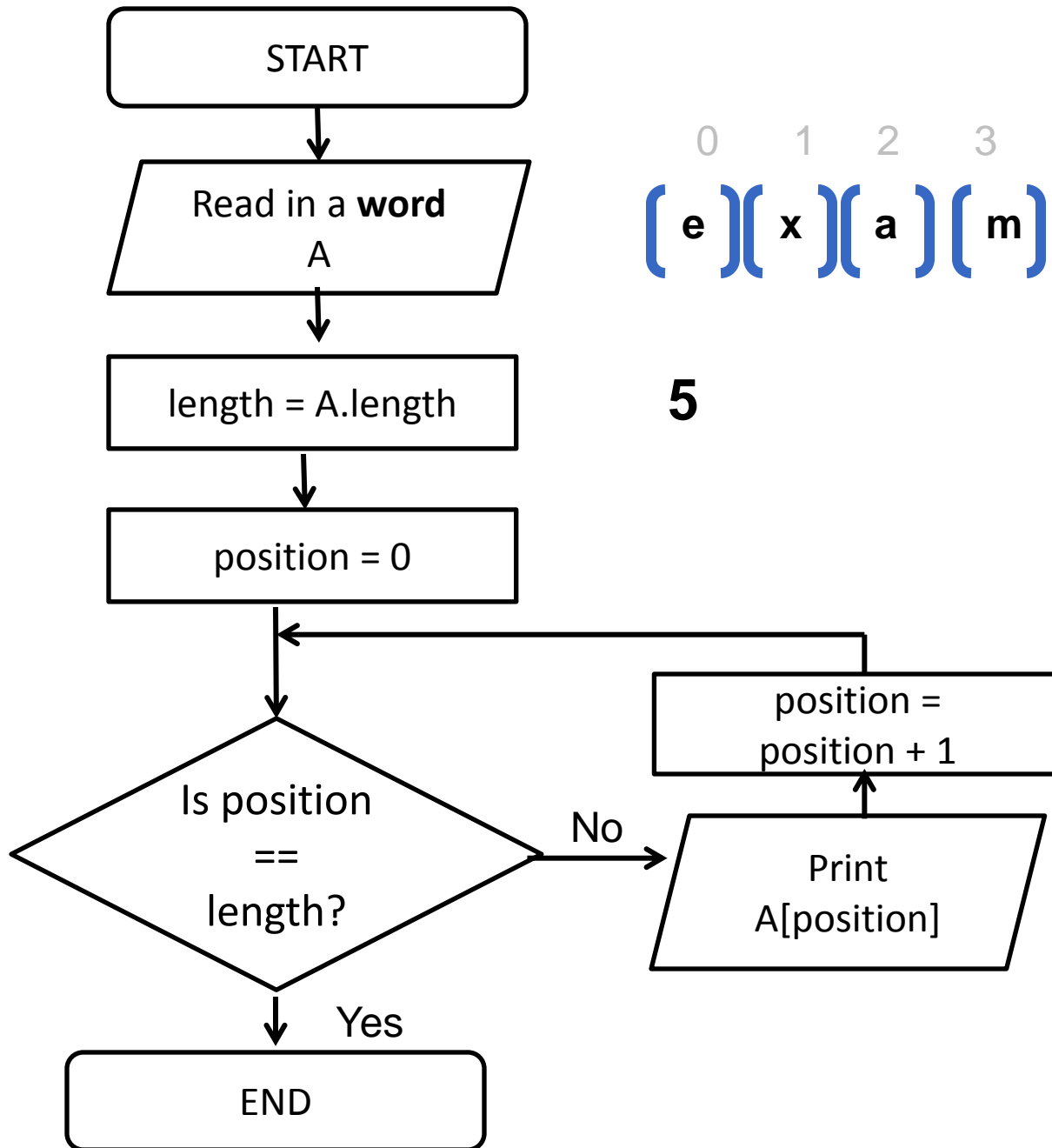


0 1 2 3 4  
( e )( x )( a )( m )( s )

5

position	Array ref	value
0	A[0]	e
1	A[1]	x
2	A[2]	a
3	A[3]	m
4	A[4]	s

s



0 1 2 3 4  
( e )( x )( a )( m )( s )






5

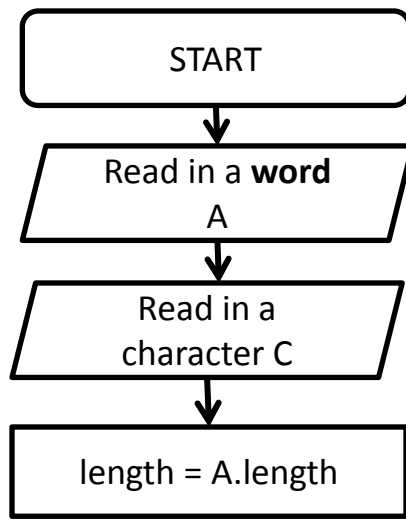
position	Array ref	value
0	A[0]	e
1	A[1]	x
2	A[2]	a
3	A[3]	m
4	A[4]	s

# Flowcharts (Problem 17)

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

*Given a word and a character from that word print out the number of times the character is in the word.*

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 output.
	Process	A rectangle represents a process.
	Decision	A diamond indicates a decision.

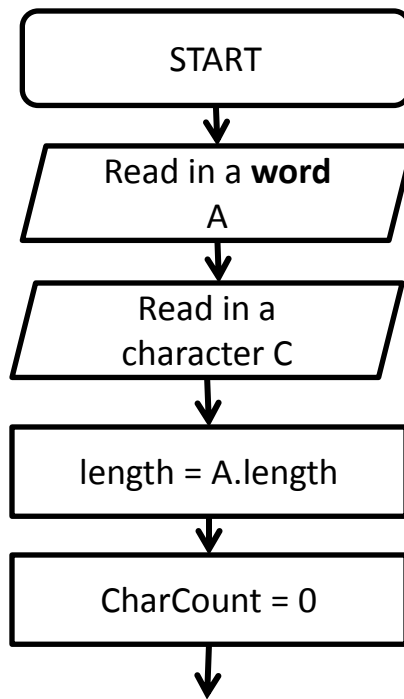


0 1 2 3 4 5

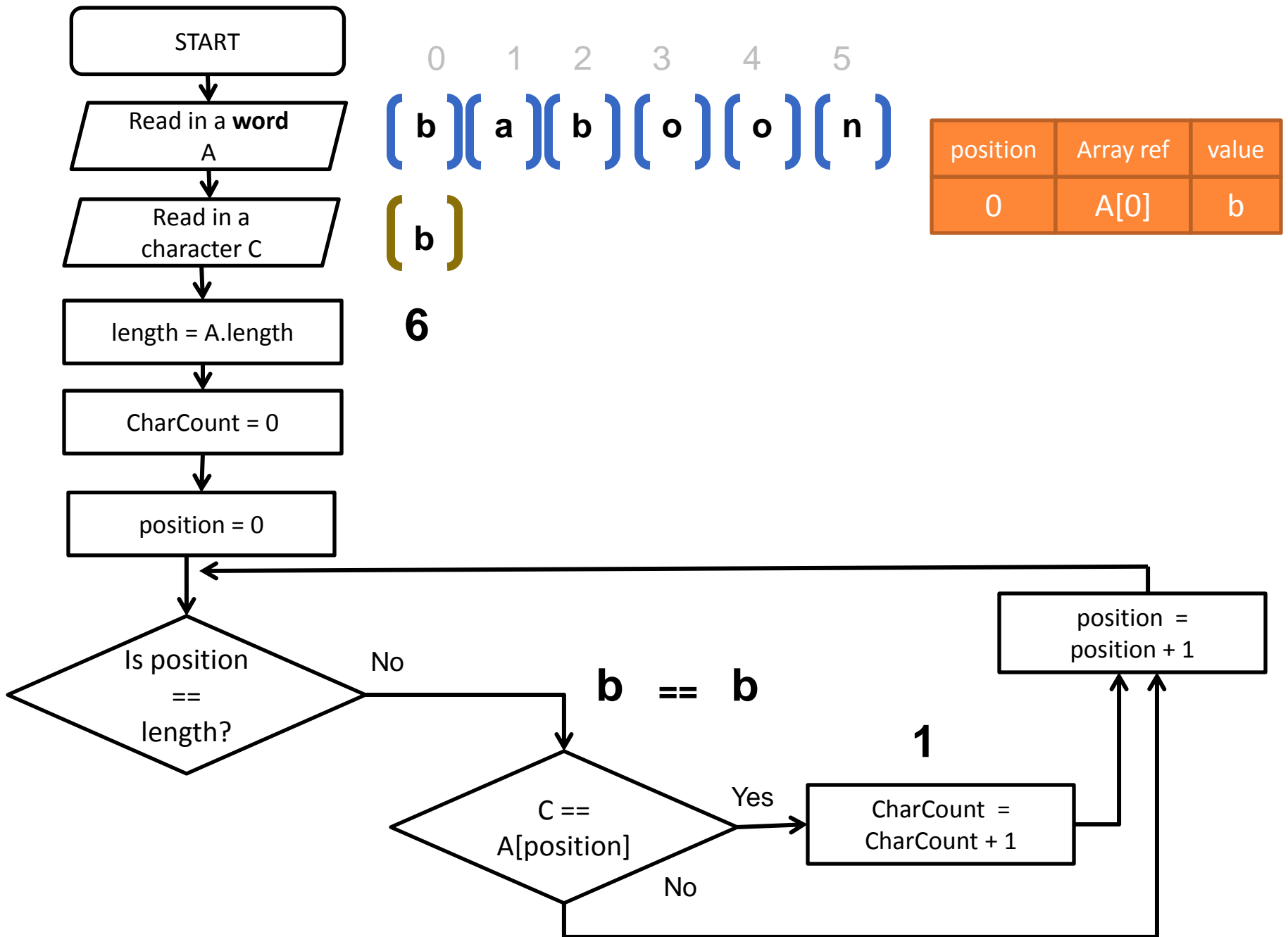
( b ) ( a ) ( b ) ( o ) ( o ) ( n )

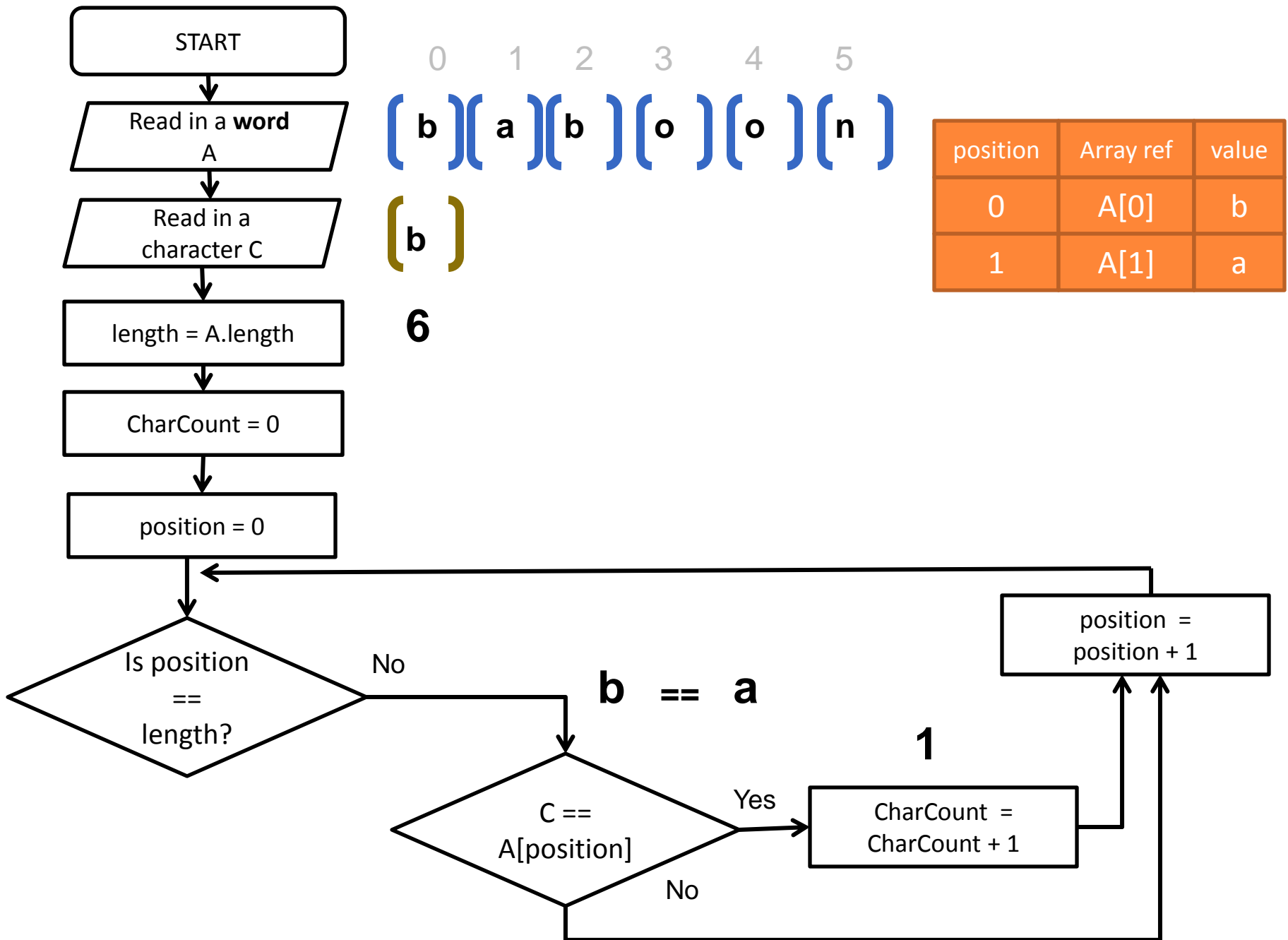
( b )

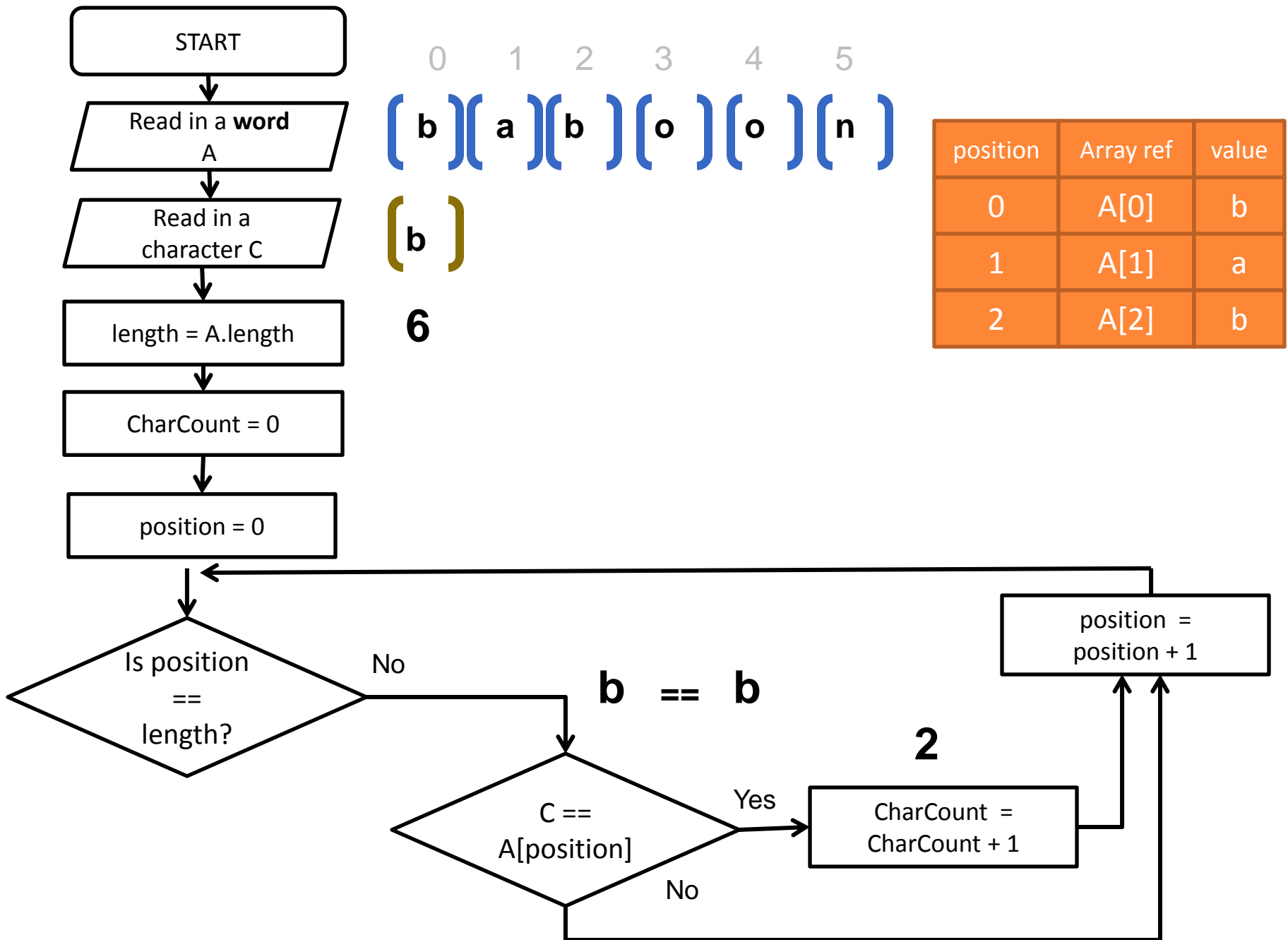
6



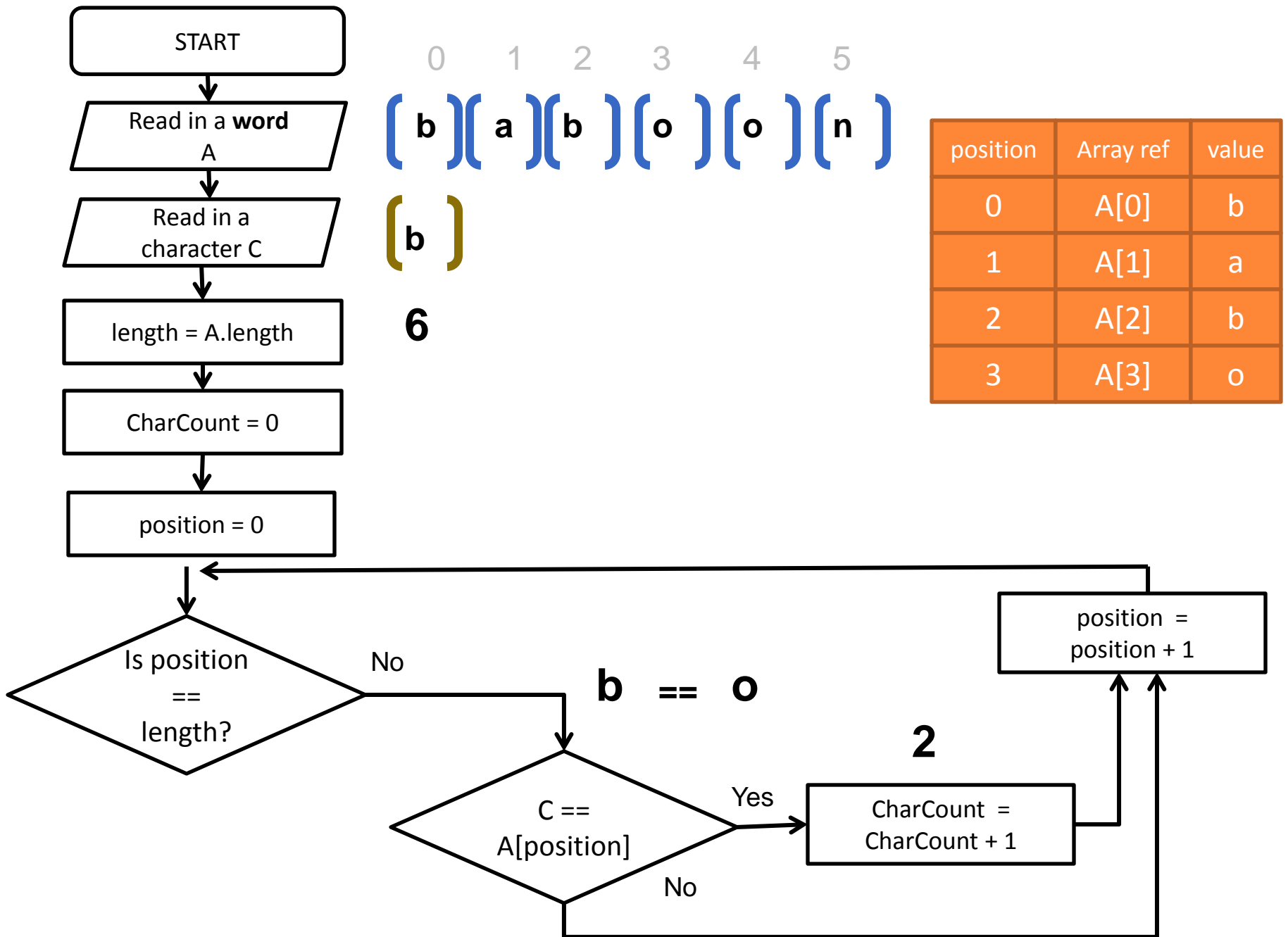
0 1 2 3 4 5  
( b ) ( a ) ( b ) ( o ) ( o ) ( n )  
( b )  
6

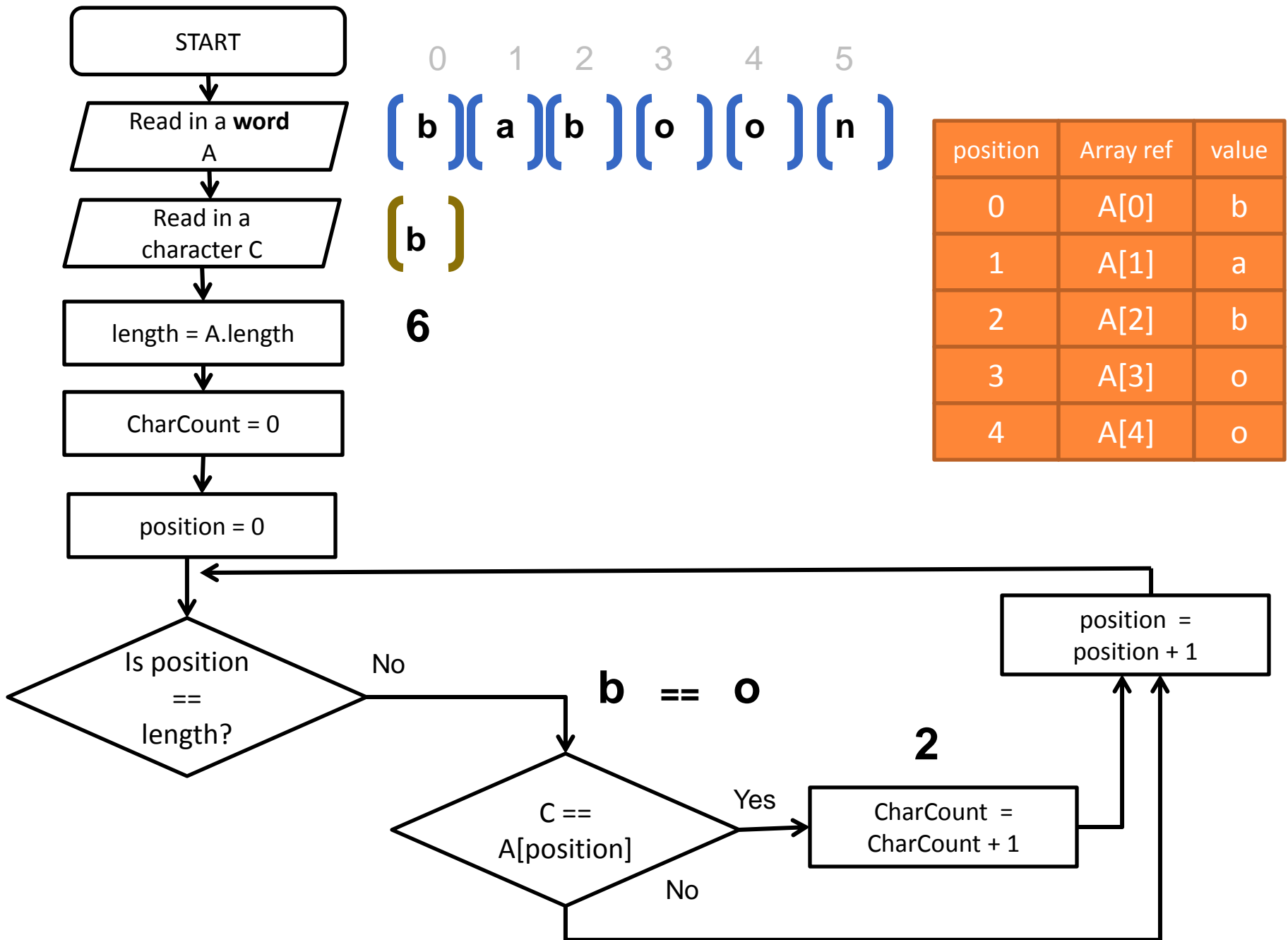


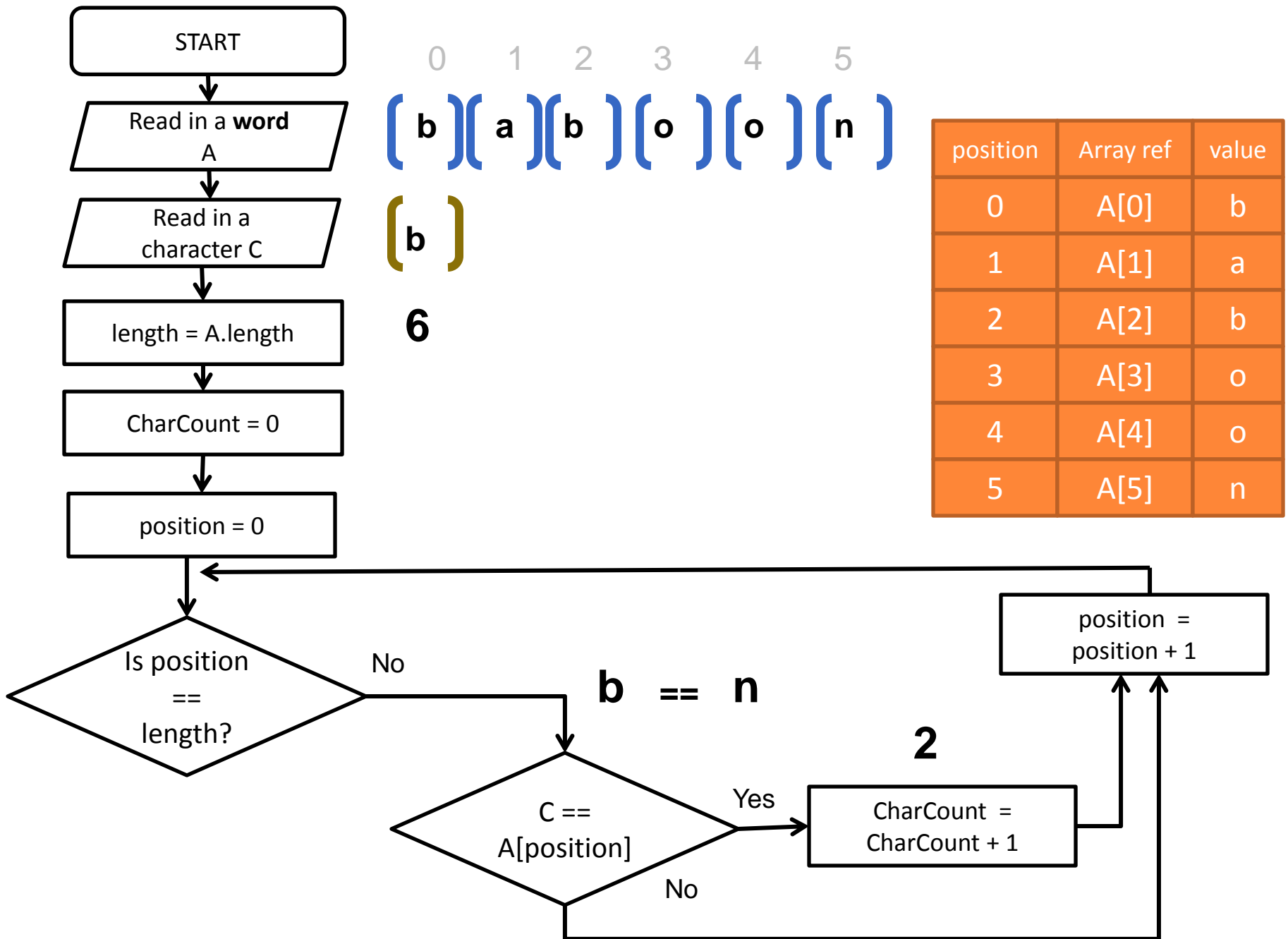


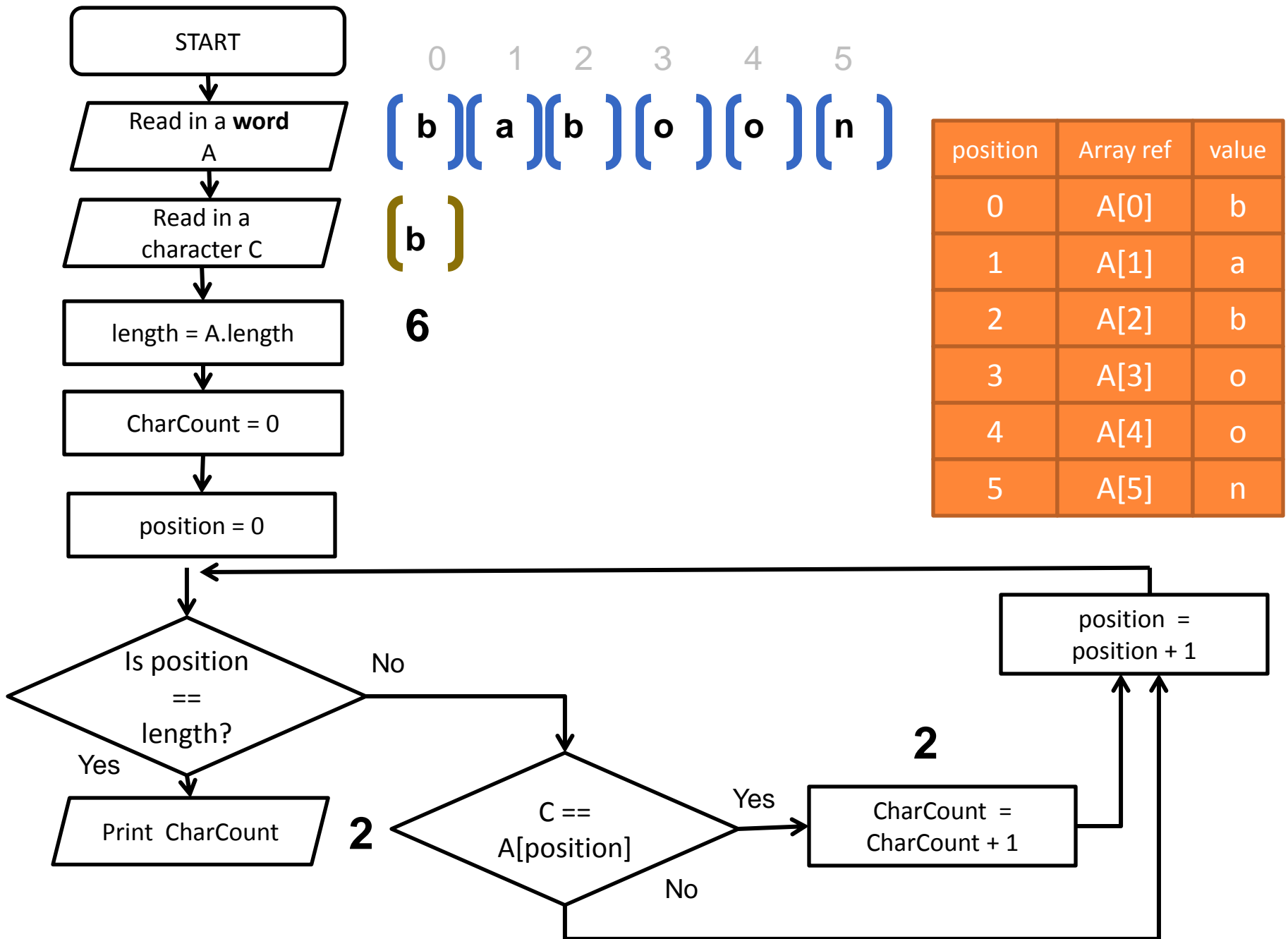


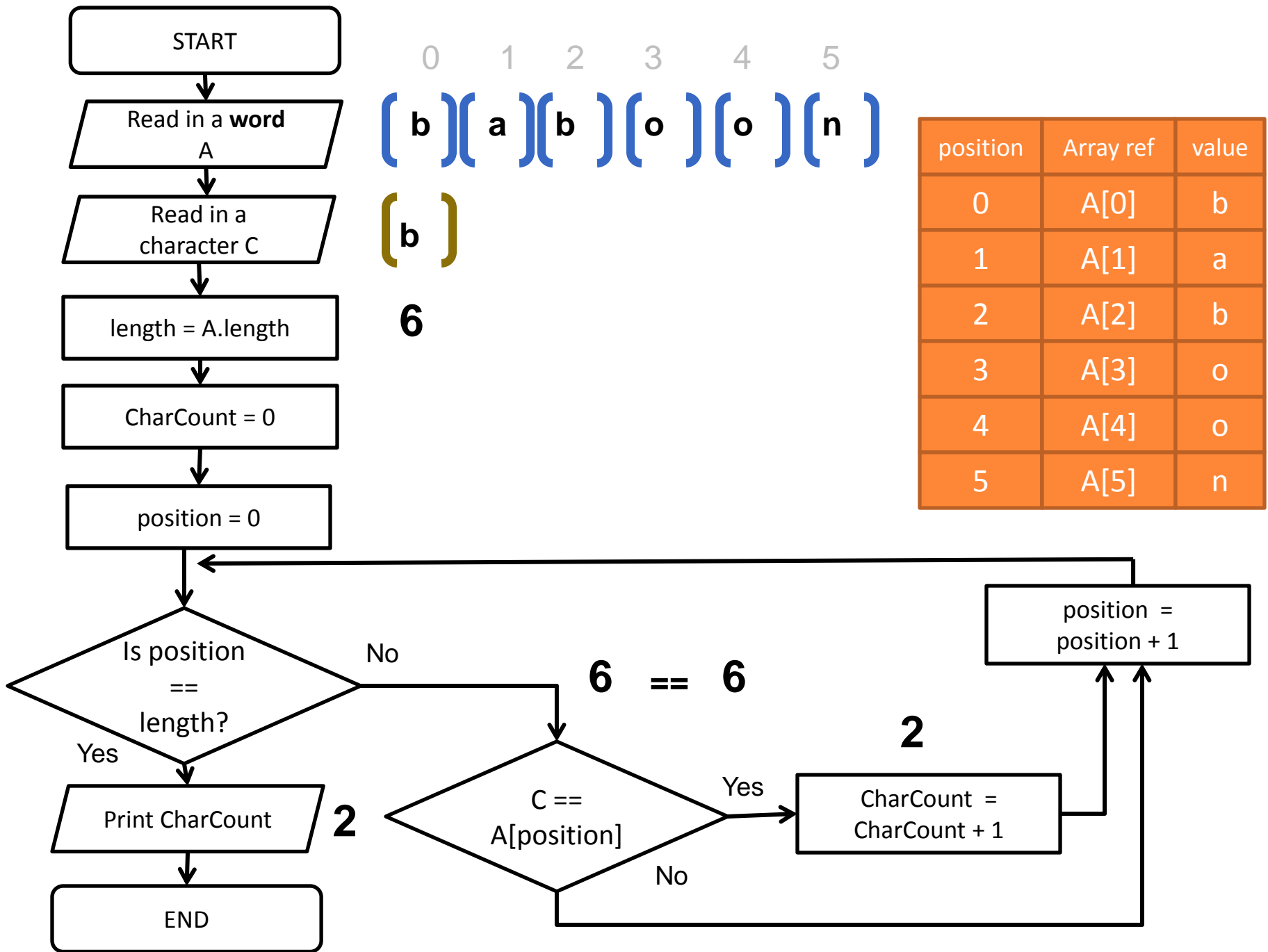












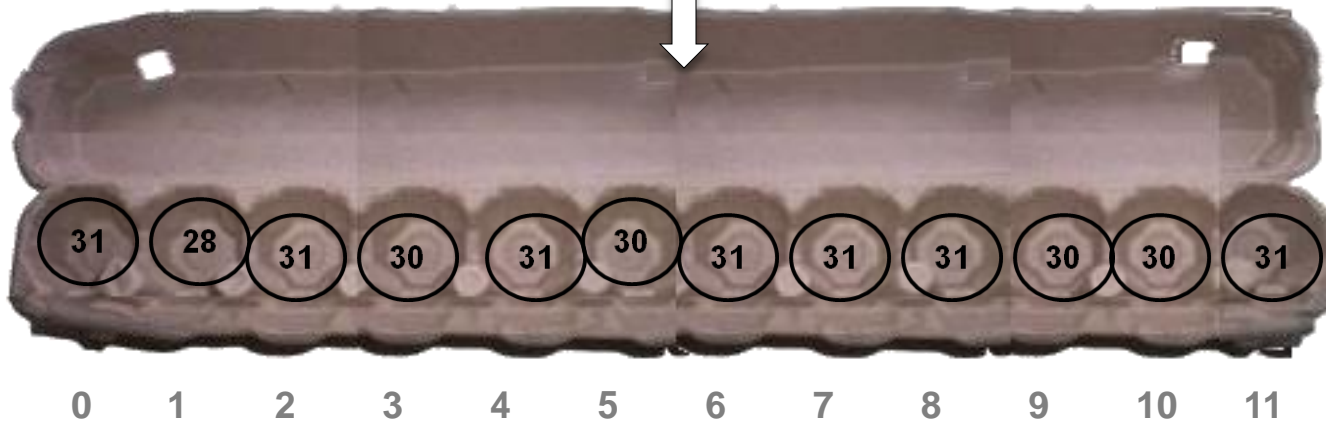
# Not just for letters in a word

We can store lists of other data types  
in an array

# What is it used for

- Example: Used to store lists of things that you may want to reference  
e.g. to store the number of days in each month of the year, or a list of words of a wordlist

**daysInEachMonthArray**



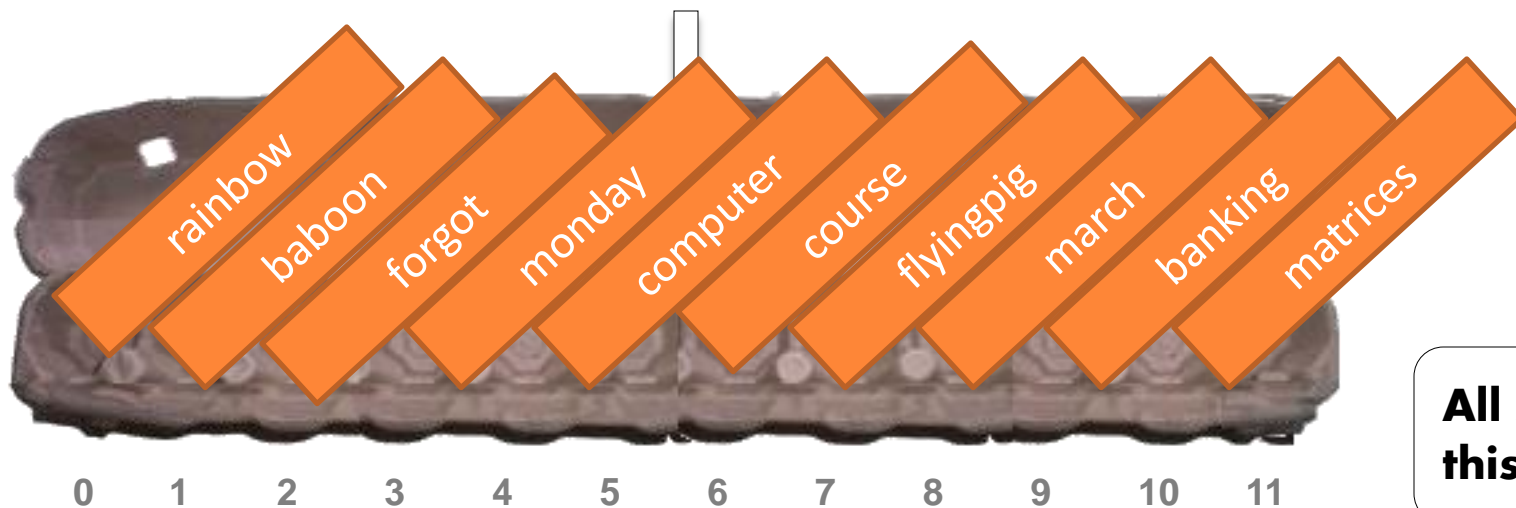
**All Integers in  
this case**

# What is it used for

- Example: Used to store lists of things that you may want to reference

e.g. to store the number of days in each month of the year, or a list of words of a wordlist

**WordListArray**



**All strings in this case**



# Using the Array (in general)

```
int daysInEachMonthArray [12];           //declare array size
```

```
daysInEachMonthArray = {31,28,31,30,31,30,31,31,31,30,30,31};
```

```
// assign one value
```

```
daysInEachMonthArray[1] = 28;           //February
```

```
// print a value
```

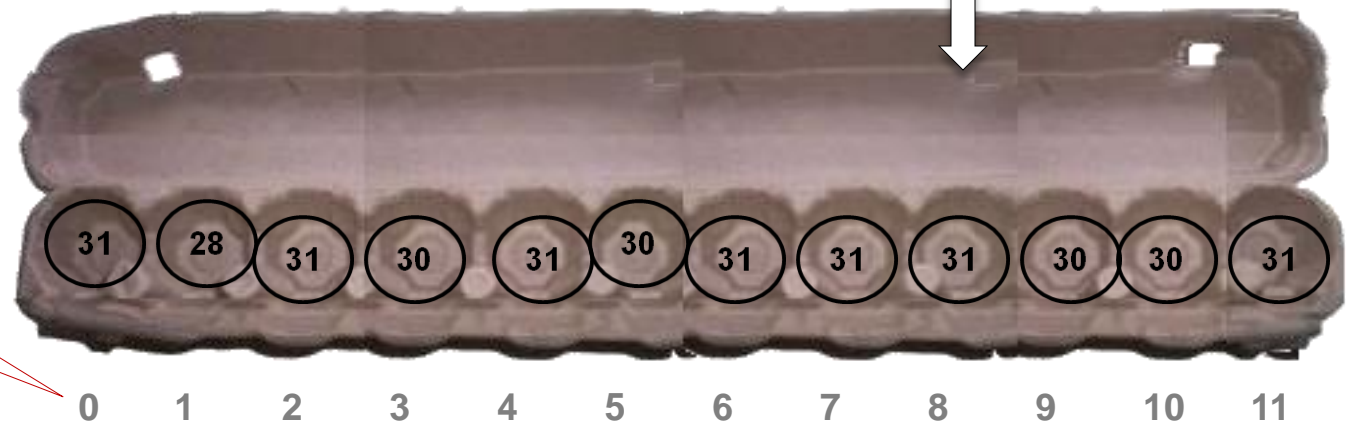
```
Print daysInEachMonthArray[7]           //August
```

variable name

31

**daysInEachMonthArray**






**The position  
counter starts  
at zero**



# Flowcharts (Problem 18)

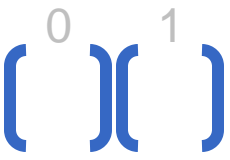
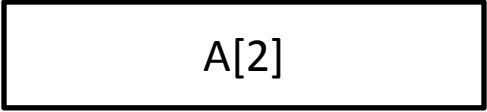
- So let's say we want to express the following algorithm:

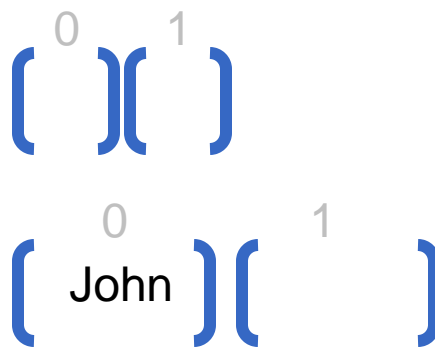
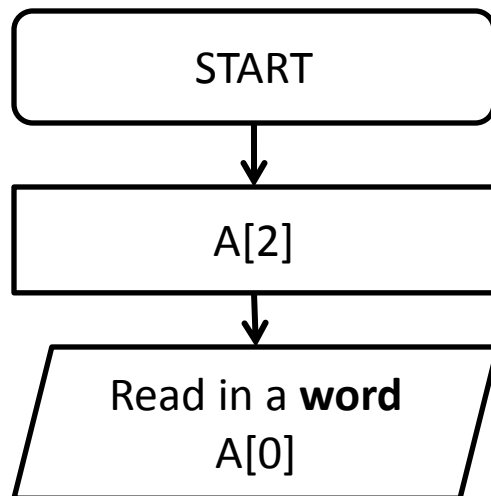
*Read in two words and print them out (We want to use an array to store the words).*

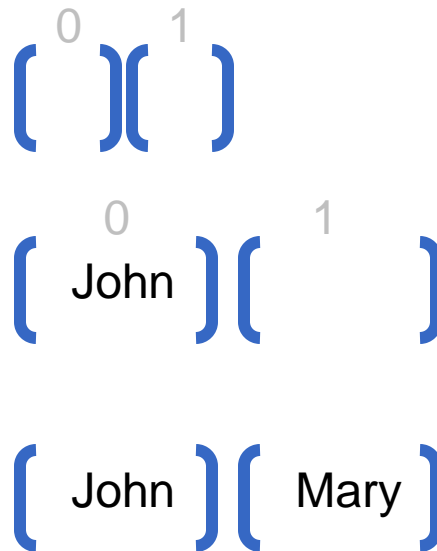
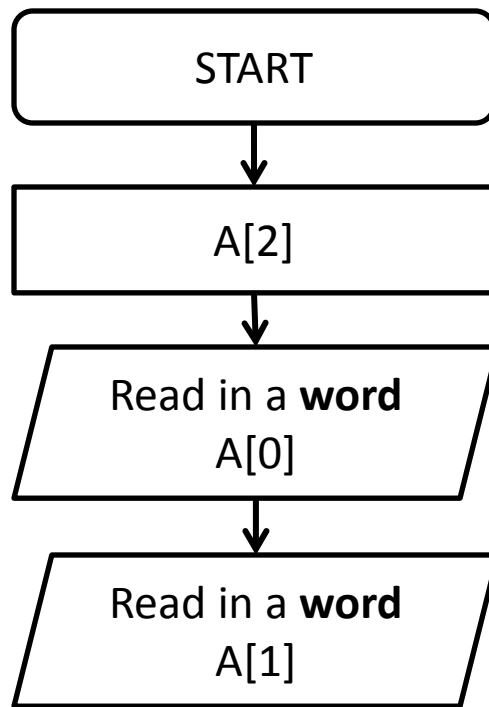
Symbol	Name	Function
	Start/end	An oval represents a start or end point.
	Arrows	A line is a connector that shows relationships between the representative shapes.
	Input/Output	A parallelogram represents input or output.
	Process	A rectangle represents a process.
	Decision	A diamond indicates a decision.

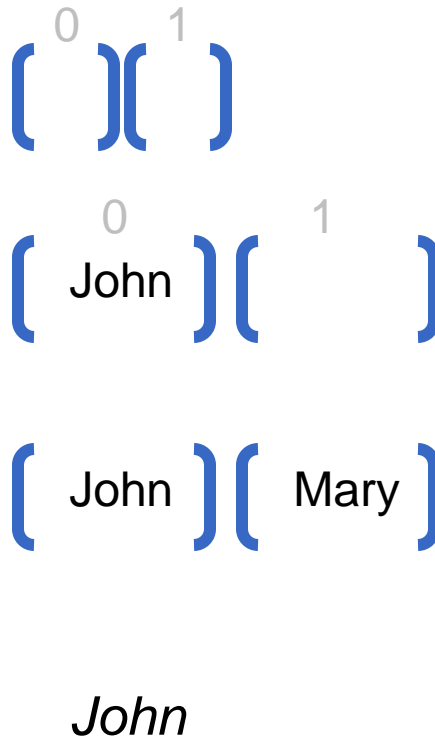
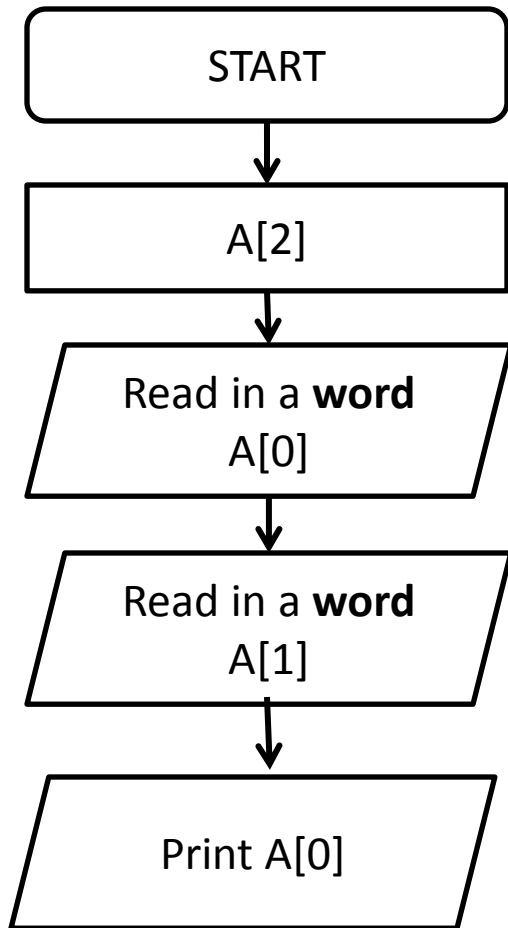
START











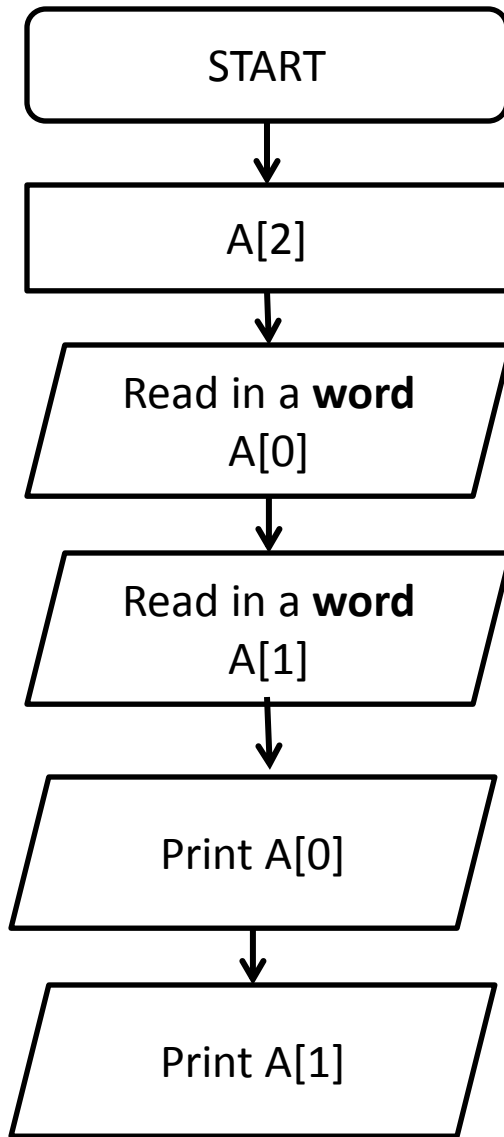


Diagram illustrating the state of the array A:

Initial state (empty slots):

0	1

After reading "John" into A[0]:

0	1
John	

After reading "Mary" into A[1]:

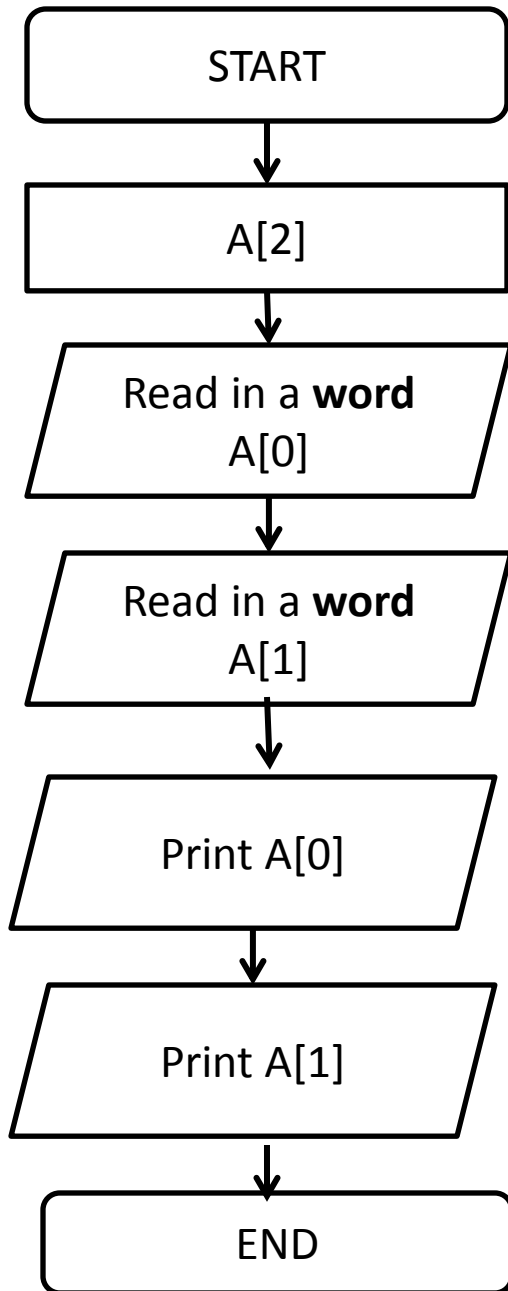
0	1
John	Mary

Output of the program:

*John*

*Mary*





<sup>0</sup> <sup>1</sup>  
( ) ( )

<sup>0</sup> <sup>1</sup>  
( John ) ( )

( John ) ( Mary )






*John*

*Mary*

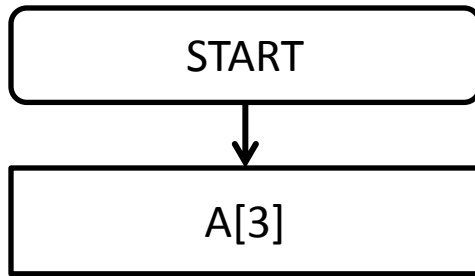
# Flowcharts (Problem 19)

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

*Place 3 words into memory and print them out (We want to use an array to store the words).*

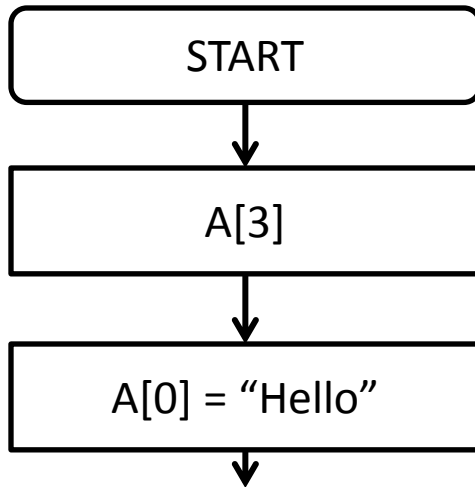
Symbol	Name	Function
	Start/end	An oval represents a start or end point.
	Arrows	A line is a connector that shows relationships between the representative shapes.
	Input/Output	A parallelogram represents input or output.
	Process	A rectangle represents a process.
	Decision	A diamond indicates a decision.





Create a variable, called A  
A is an array.

<sup>0</sup> <sup>1</sup> <sup>2</sup>  
( ) ( ) ( )

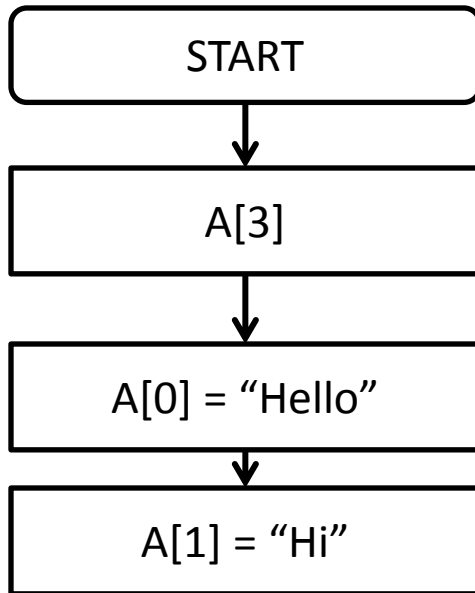


Create a variable, called A  
A is an array.

<sup>0</sup> <sup>1</sup> <sup>2</sup>  
( ) ( ) ( )

The memory slots owned by the variable A can be  
referenced using an index number that starts as  
zero

<sup>0</sup> <sup>1</sup> <sup>2</sup>  
( Hello ) ( ) ( )

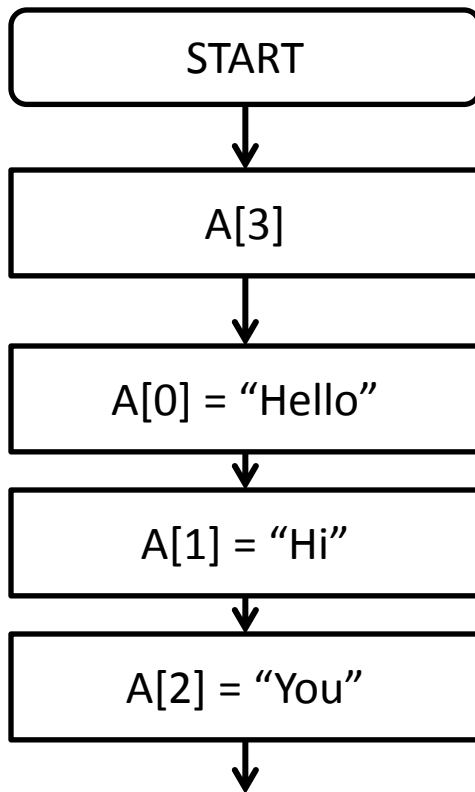


Create a variable, called A  
A is an array.

<sup>0</sup> <sup>1</sup> <sup>2</sup>  
( ) ( ) ( )

The memory slots owned by the variable A can be  
referenced using an index number that starts as  
zero

<sup>0</sup> <sup>1</sup> <sup>2</sup>  
( Hello ) ( Hi ) ( )

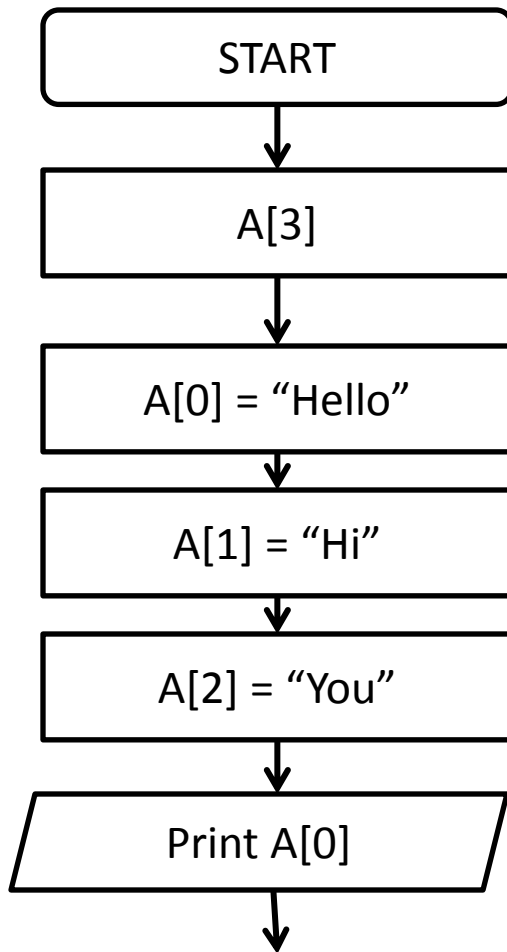


Create a variable, called A  
A is an array.

<sup>0</sup> <sup>1</sup> <sup>2</sup>  
( ) ( ) ( )

The memory slots owned by the variable A can be  
referenced using an index number that starts as  
zero

<sup>0</sup> <sup>1</sup> <sup>2</sup>  
( Hello ) ( Hi ) ( You )



Create a variable, called A  
A is an array.

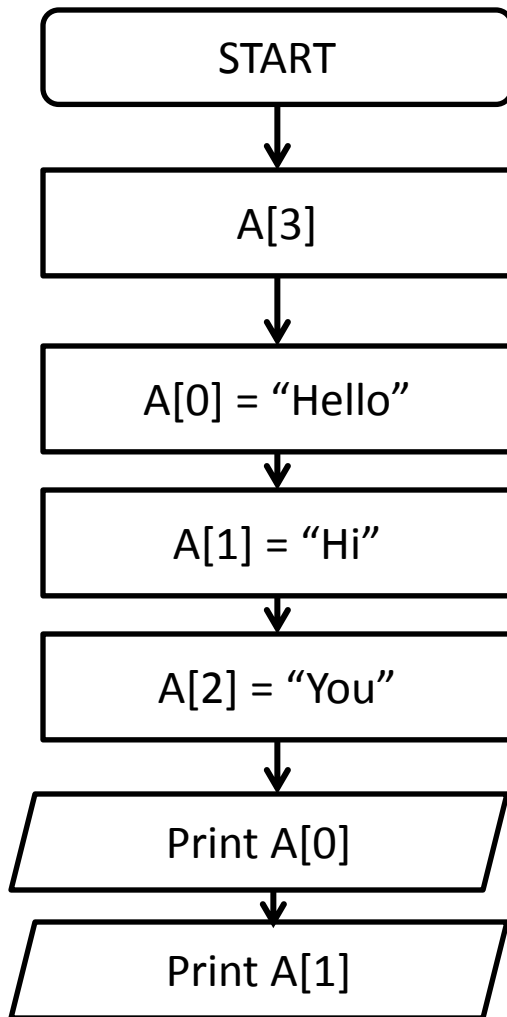
<sup>0</sup> <sup>1</sup> <sup>2</sup>  
( ) ( ) ( )

The memory slots owned by the variable A can be referenced using an index number that starts as zero

<sup>0</sup> <sup>1</sup> <sup>2</sup>  
( Hello ) ( Hi ) ( You )

*Hello*





Create a variable, called A  
A is an array.

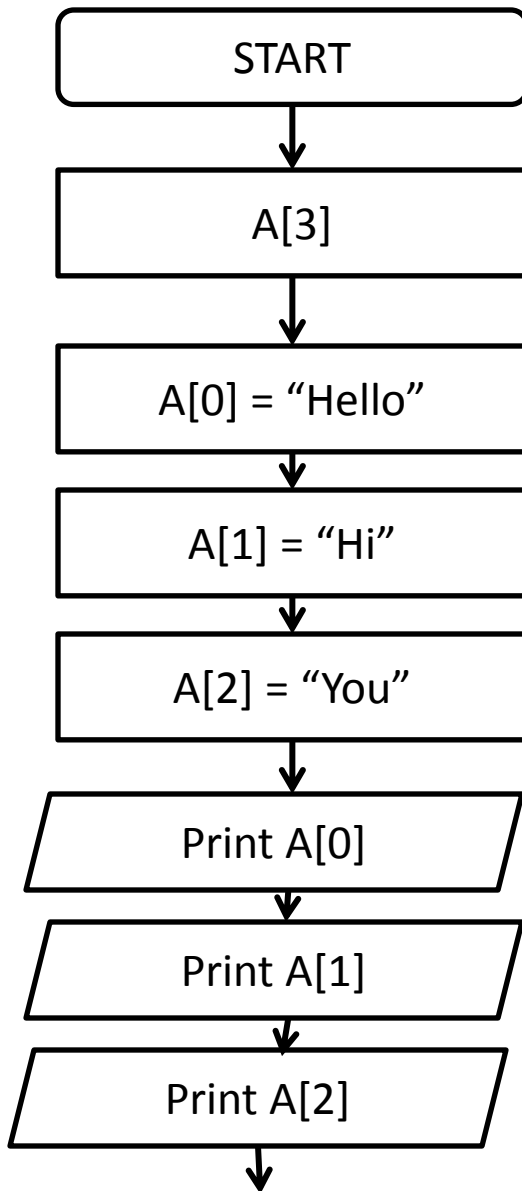
<sup>0</sup> <sup>1</sup> <sup>2</sup>  
( ) ( ) ( )

The memory slots owned by the variable A can be referenced using an index number that starts as zero

<sup>0</sup> <sup>1</sup> <sup>2</sup>  
( Hello ) ( Hi ) ( You )

*Hello*

*Hi*



Create a variable, called A  
A is an array.

<sup>0</sup> <sup>1</sup> <sup>2</sup>  
( ) ( ) ( )

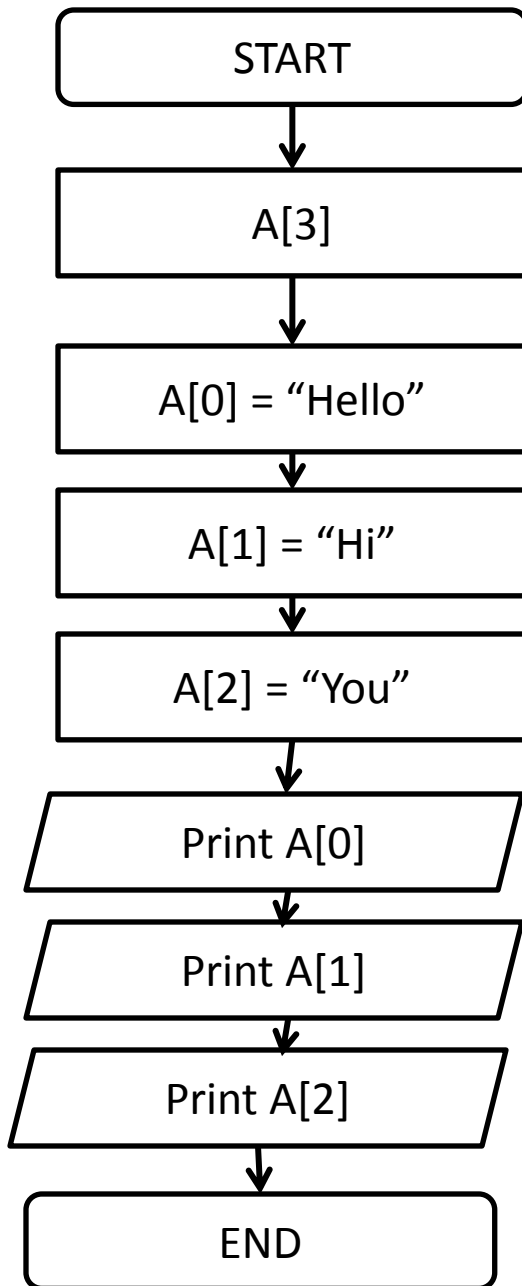
The memory slots owned by the variable A can be referenced using an index number that starts as zero

<sup>0</sup> <sup>1</sup> <sup>2</sup>  
( Hello ) ( Hi ) ( You )

*Hello*

*Hi*

*You*



Create a variable, called A  
A is an array.

<sup>0</sup> <sup>1</sup> <sup>2</sup>  
( ) ( ) ( )

The memory slots owned by the variable A can be referenced using an index number that starts as zero

<sup>0</sup> <sup>1</sup> <sup>2</sup>  
( Hello ) ( Hi ) ( You )

*Hello*






*Hi*

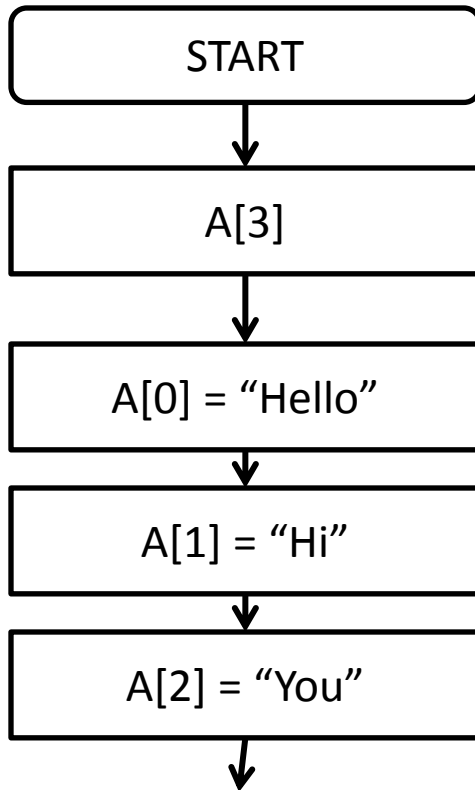
*You*

# Flowcharts (Problem 20)

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

*Place 3 words into memory, generate a random number between 1 and 3, print out the word chosen by the random number generator.*

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 output.
	Process	A rectangle represents a process.
	Decision	A diamond indicates a decision.

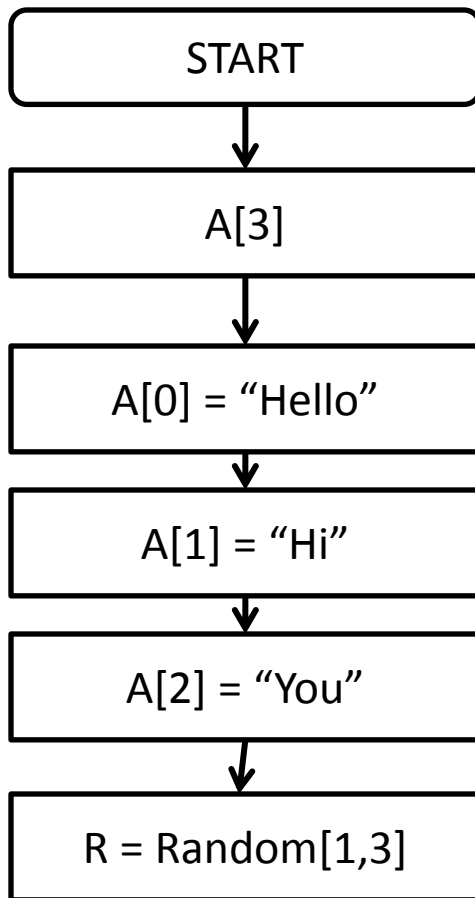


Create a variable, called A  
A is an array.

<sup>0</sup> <sup>1</sup> <sup>2</sup>  
( ) ( ) ( )

The memory slots owned by the variable A can be  
referenced using an index number that starts as  
zero

<sup>0</sup> <sup>1</sup> <sup>2</sup>  
( Hello ) ( Hi ) ( You )



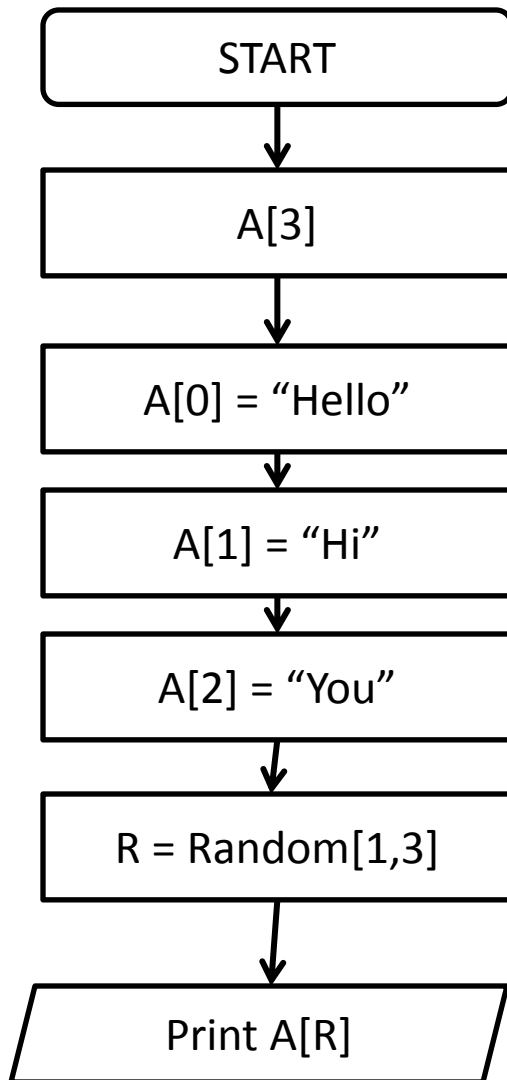
Create a variable, called A  
A is an array.

<sup>0</sup> <sup>1</sup> <sup>2</sup>  
( ) ( ) ( )

The memory slots owned by the variable A can be  
referenced using an index number that starts as  
zero

<sup>0</sup> <sup>1</sup> <sup>2</sup>  
( Hello ) ( Hi ) ( You )

$R = 2$



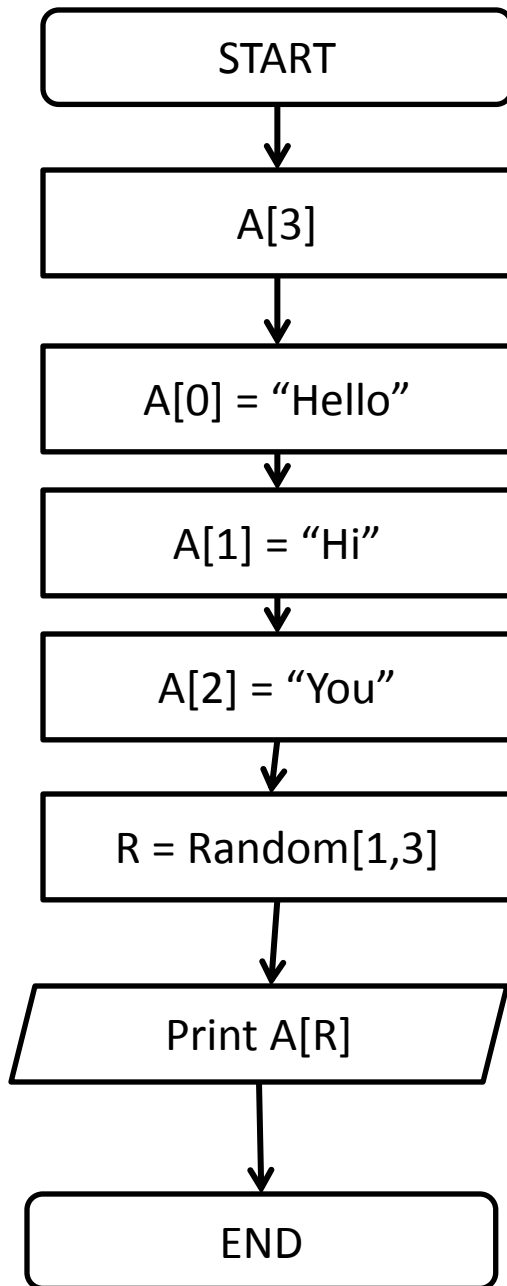
Create a variable, called A  
A is an array.

<sup>0</sup> <sup>1</sup> <sup>2</sup>  
( ) ( ) ( )

The memory slots owned by the variable A can be referenced using an index number that starts as zero

<sup>0</sup> <sup>1</sup> <sup>2</sup>  
( Hello ) ( Hi ) ( You )

$R = 2$



Create a variable, called A  
A is an array.

<sup>0</sup> <sup>1</sup> <sup>2</sup>  
( ) ( ) ( )

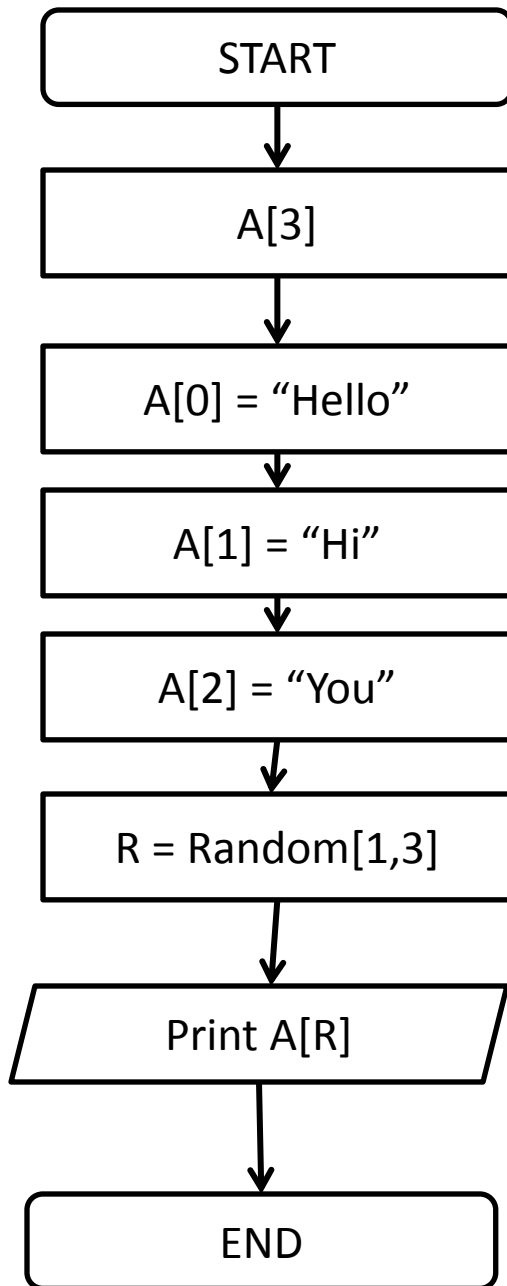
The memory slots owned by the variable A can be referenced using an index number that starts as zero

<sup>0</sup> <sup>1</sup> <sup>2</sup>  
( Hello ) ( Hi ) ( You )

$R = 2$

**There is a problem here with the use of the variable R to access a position in an array**





Create a variable, called A  
A is an array.

<sup>0</sup> <sup>1</sup> <sup>2</sup>  
( ) ( ) ( )

The memory slots owned by the variable A can be referenced using an index number that starts as zero

<sup>0</sup> <sup>1</sup> <sup>2</sup>  
( Hello ) ( Hi ) ( You )

$R = 2$

What are R's possible values?

0 1 2 Problem? 1 2 3

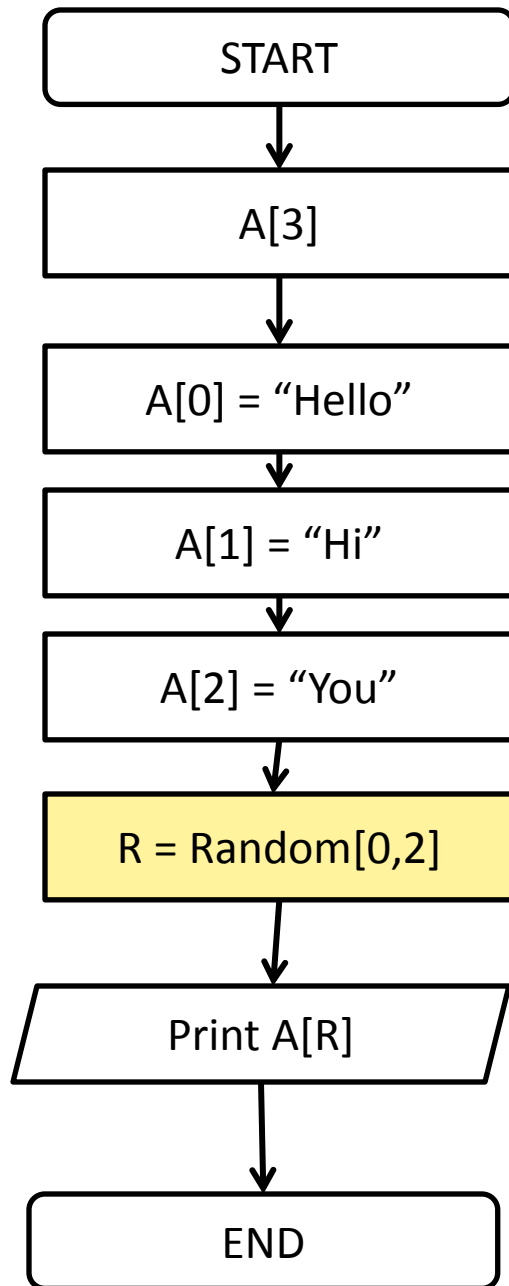
What are the index values on the array?

0                      1                      2  
( Hello ) ( Hi ) ( You )

The 0 word will never be picked

If position 3 is used there will be  
an error as there is no position 3

$R = \text{Random}[1,3]$



## OPTION 1

Create a variable, called A  
A is an array.

<sup>0</sup> <sup>1</sup> <sup>2</sup>  
( ) ( ) ( )

The memory slots owned by the variable A can be referenced using an index number that starts as zero

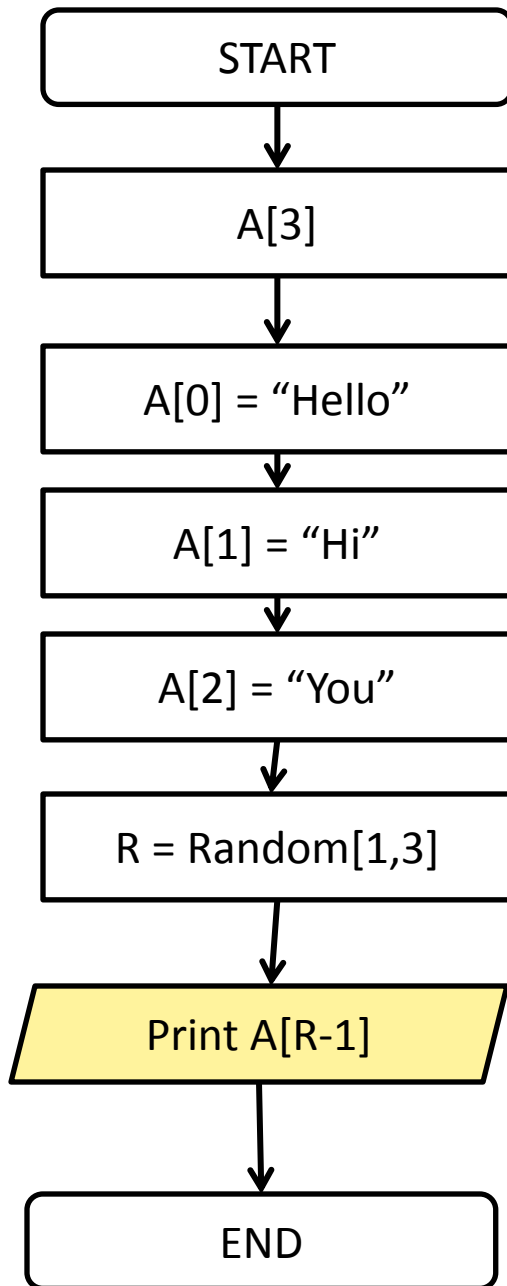
<sup>0</sup> <sup>1</sup> <sup>2</sup>  
( Hello ) ( Hi ) ( You )

$R = 2$

*You*

**Option 1:**

**Change the random numbers to be integers between 0 and 2**



## OPTION 2

Create a variable, called A  
A is an array.

<sup>0</sup> <sup>1</sup> <sup>2</sup>  
( ) ( ) ( )

The memory slots owned by the variable A can be referenced using an index number that starts as zero

<sup>0</sup> <sup>1</sup> <sup>2</sup>  
( Hello ) ( Hi ) ( You )

$R = 2$

Hi

$A[1]$

**Option 2:**

**R - 1 will ensure we start from 0  
and never use 3**

# References

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