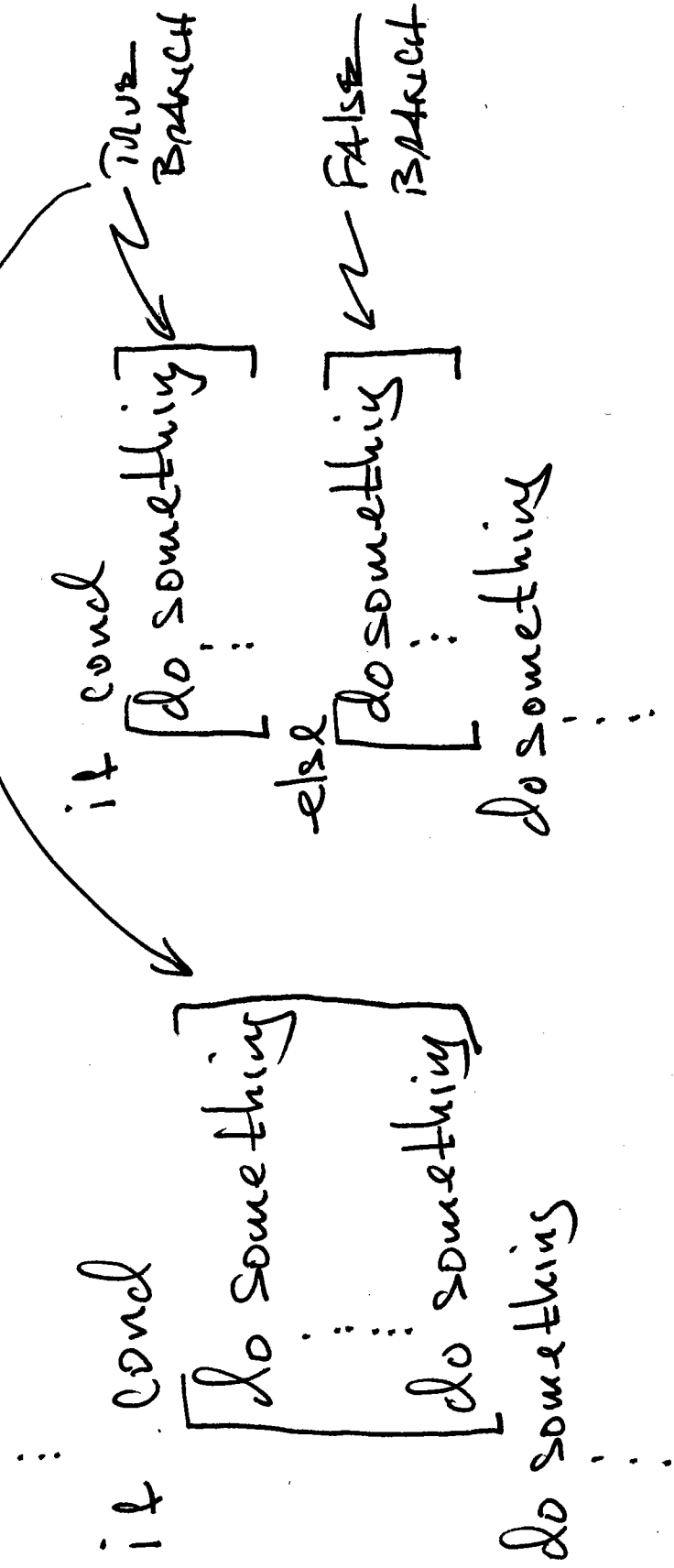


CNPS 10 1-15-08

CONDITIONAL OPERATIONS



Ex. SAFE Division

- 1.) get a, b
- 2.) if b=0
- 3.) Print 'ERROR: DIVISION BY ZERO'
- 4.) else
- 5.) quotient ← $\frac{a}{b}$
- 6.) Print quotient
- 7.) stop

TRACE:
ASSUME USER ENTRIES 9 4



Ex.

- 1.) response ← 'yes'
- 2.) while response = 'yes'
- 3.) get a, b
- 4.) if b = 0
- 5.) Print 'Error: Div. By zero'
- 6.) else
- 7.) print $\frac{a}{b}$
- 8.) Print 'Do you wish to continue?'
- 9.) get response
- 10.) stop

do [do something ↺ loop body

do something

while cond ← loop REPEITION CONDITION

do something

⋮

Ex.

- 1.) do
- 2.) get a, b
- 3.) if b = 0
- 4.) print 'Error'
- 5.) else
- 6.) print a/b
- 7.) print 'Continue?'
- 8.) get response
- 9.) while response = 'yes'
- 10.) stop

WE OFTEN HAVE WHILE LOOPS THAT FOLLOW

THIS PATTERN!

```

i ← 1 ← initialize loop control variable (LCV)
while (i ≤ n) ← TEST loop REP. COND. (LRC)
{
  do something with i
  i ← i + 1 ← INCREMENT LCV
}

```

...

NOTE: THE LCV IN THIS CASE IS i .

Ex

- 1.) $i \leftarrow 1$
- 2.) while $i \leq 5$
- 3.) print i^2
- 4.) $i \leftarrow i + 1$
- 5.) stop

OUTPUT

1 ✓
 4 ✓
 9 ✓
 16 ✓
 25 ✓

Trace:

$\frac{i}{1}$
 2
 3
 4
 5
 6

- loop body executes 5 times
- final value of i is 6.

LAST EX

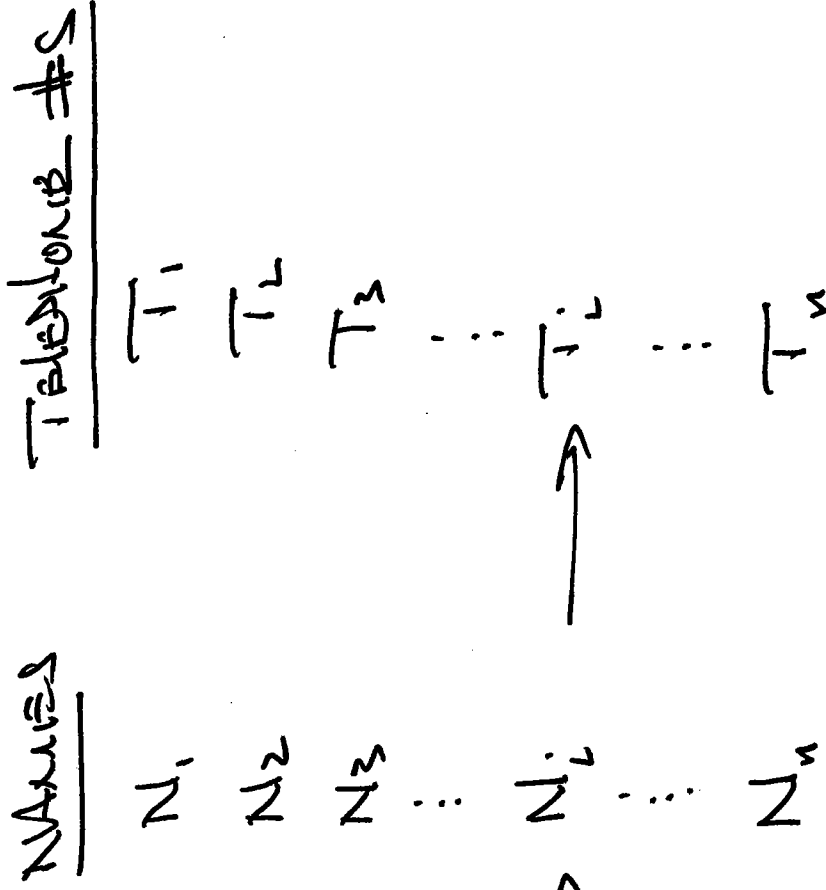
- 1.) for $i \leftarrow 1$ TO 5
- 2.) Print i^2
- 3.) stop

for $i \leftarrow 1$ TO n
do something with i

EQUIVALENT TO :

$i \leftarrow 1$
while $i \leq n$
do something with i
 $i \leftarrow i + 1$

Ex.



NAME \longrightarrow

INPUT: $n \geq 1$ (total # of names), N_1, \dots, N_n (the names), T_1, \dots, T_n (corresponding table #s), AND NAME (THE name to search for.)

✓

OUTPUT: THE TELK. # T_i FOR WHICH

NAME = N_i , OR IF NO SUCH i EXISTS

PRINT A MESSAGE \emptyset THAT EFFECT.

SEQUENTIAL SEARCH

12

1.) get $n, N_1, \dots, N_n, T_1, \dots, T_n, NAME$

2.) $i \leftarrow 1$

3.) $found \leftarrow false$

4.) while $(i \leq n \text{ and } \text{not found})$

if $N_i = NAME$

$found \leftarrow true$

print T_i

else

$i \leftarrow i + 1$

10.) if not found

print 'sorry' NAME 'not found'

12.) Stop

logical operators: and, or, not

Proposition: A STATEMENT OR ASSERTION WHICH CAN (IN PRINCIPLE) BE EVALUATED AS true or false.

Ex $6 < 7$

$10 < 3$

$0 = 1$

$1 + 5 = 6$



Propositions

$x < 7$

$x + y = 12$

'hello'



NOT Propositions

Let A, B be Propositional variables.

and

| A | B | A and B |
|---|---|---------|
| F | F | F |
| F | T | F |
| T | F | F |
| T | T | T |

OR

inclusive
OR

| A | B | A or B |
|---|---|--------|
| F | F | F |
| F | T | T |
| T | F | T |
| T | T | T |

Exclusive or : xor

| A | B | A xor B |
|---|---|---------|
| F | F | F |
| F | T | T |
| T | F | T |
| T | T | F |