# Flow Charting Example

Determine if a word or phrase is a palindrome

- Use Iterative Improvement
- Apply Dijkstra's rules

# Palindrome

A *palindrome* is a symmetric word or phrase.

- madaminedenimadam is a palindrome.
- eveingardenimadam is not a palindrome.

A palindrome is interesting when it has additional meaning as shown in the example above.

# Even Palindrome Examples

An even palindrome has an even number of letters

- DEED
- TIMEEMIT
- TOOT

# Odd Palindrome Examples

An odd palindrome has an odd number of letters

- DOOGOOD
- MADAM
- MARERAM
- MOREROM

# Approach

- MADAM
- ADA
- D

Palindrome

- DEED
- EE

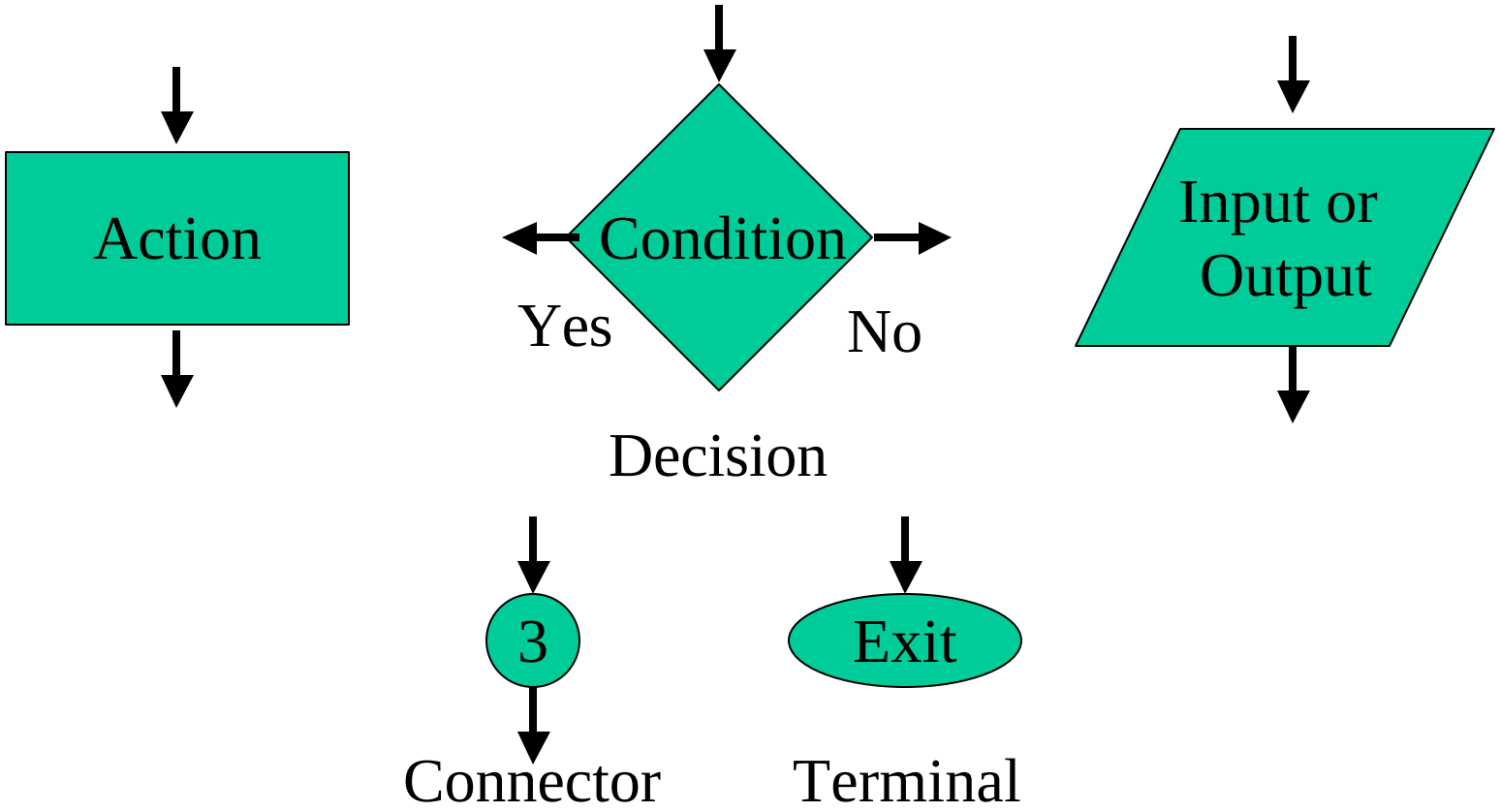
Palindrome

- TIME

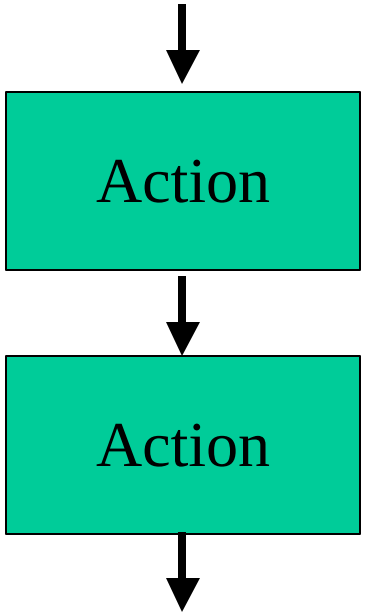
Not Palindrome

# Determine if a particular word or phrase is a palindrome

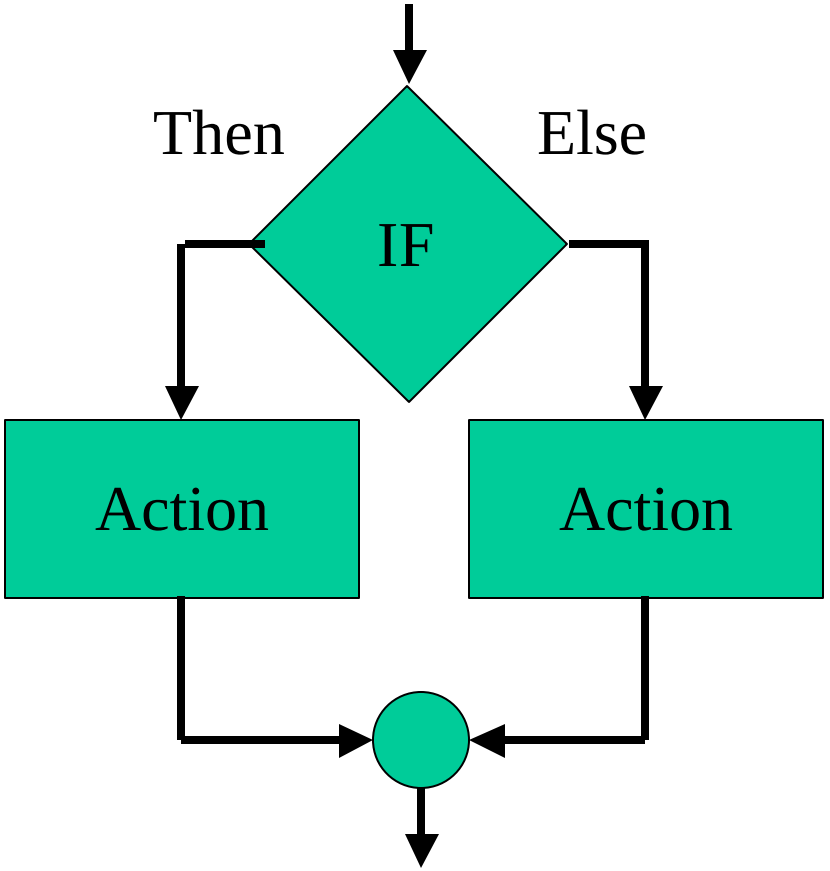
# Flow Chart Symbols



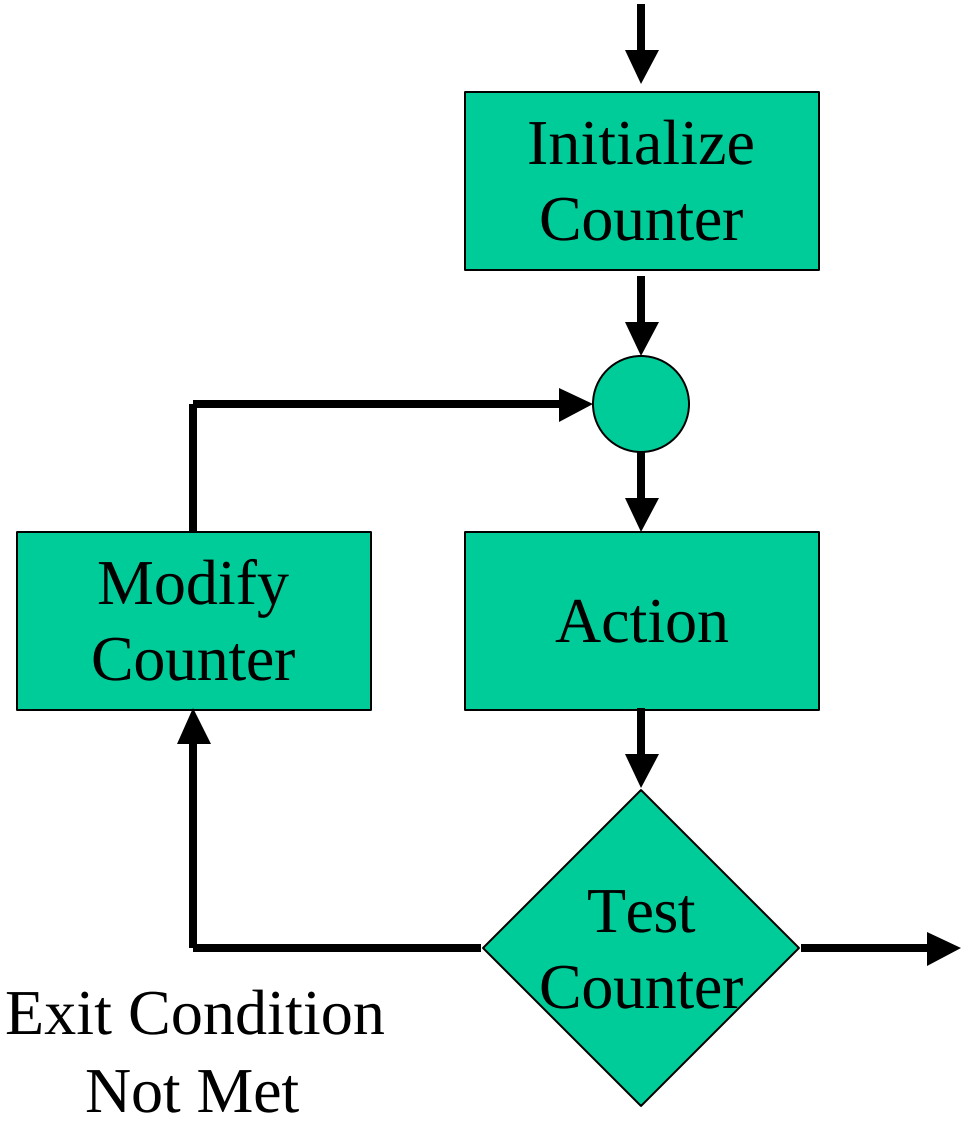
# Simple Logic Structures




Sequence




If – Then - Else




# Iteration Test After

For K = 1 to 5

Do 13 K = 1,5

For L = 100 to –15 Step -40

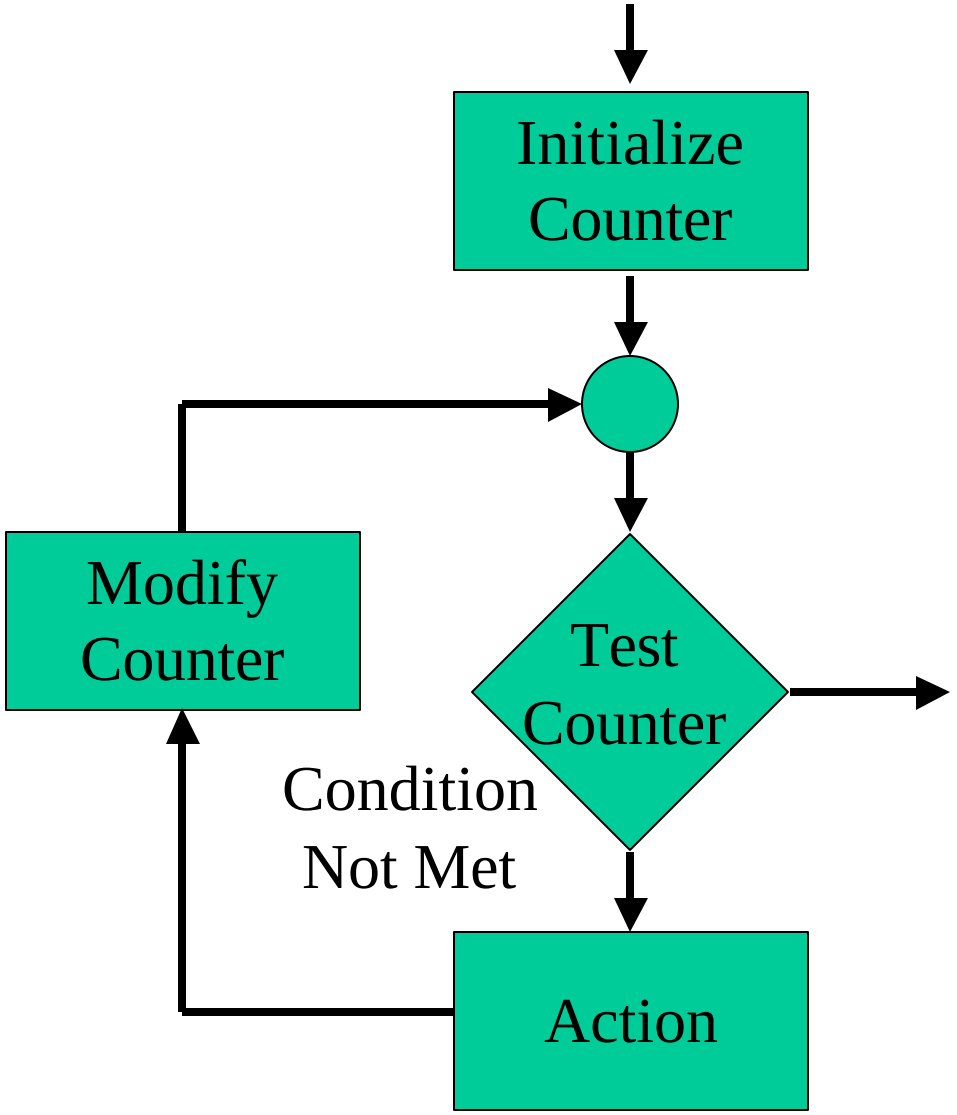
Do 13 L = 100, -15, -40

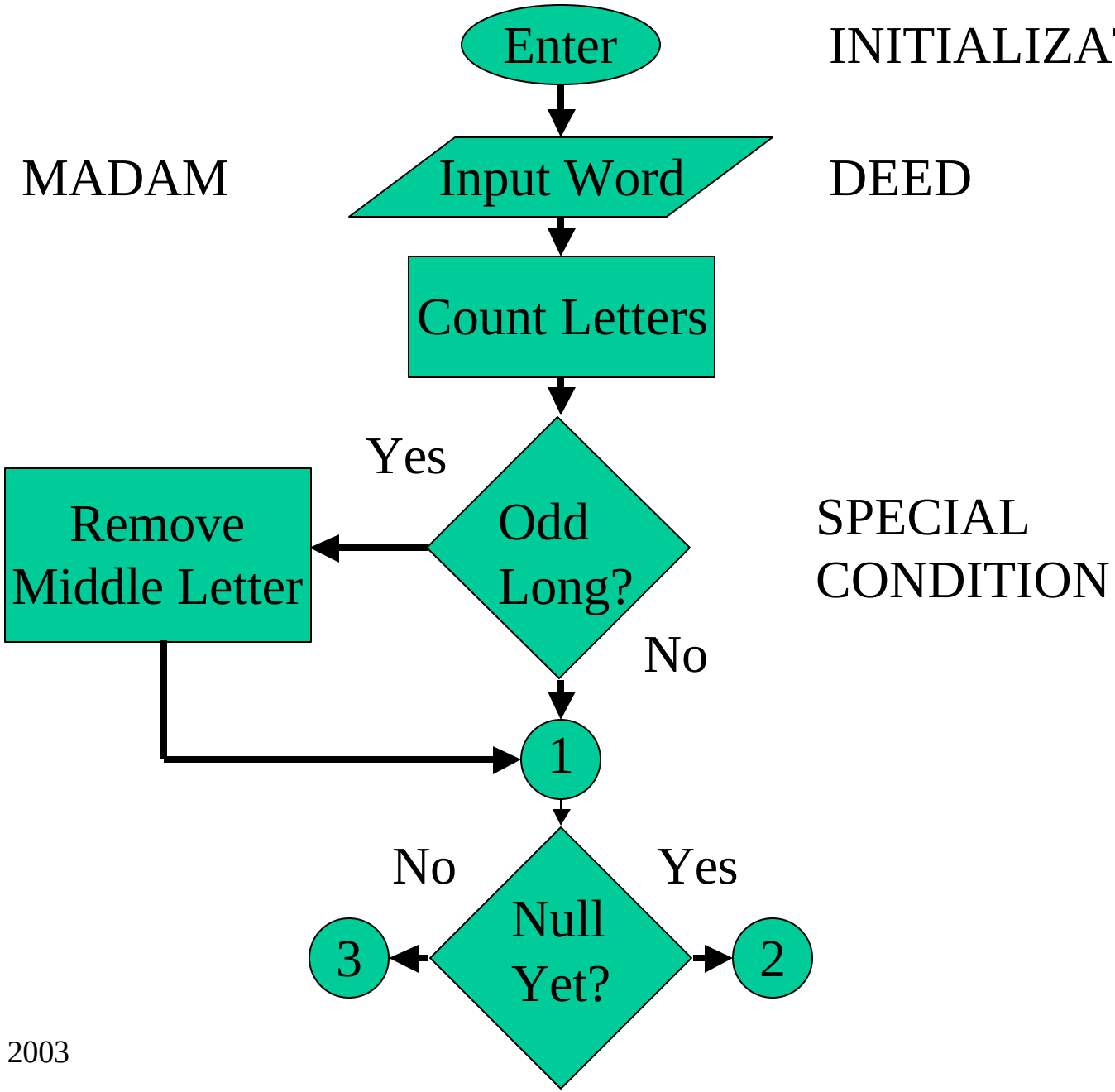
For I = J to K Step L

Do 13 I = J, K, L

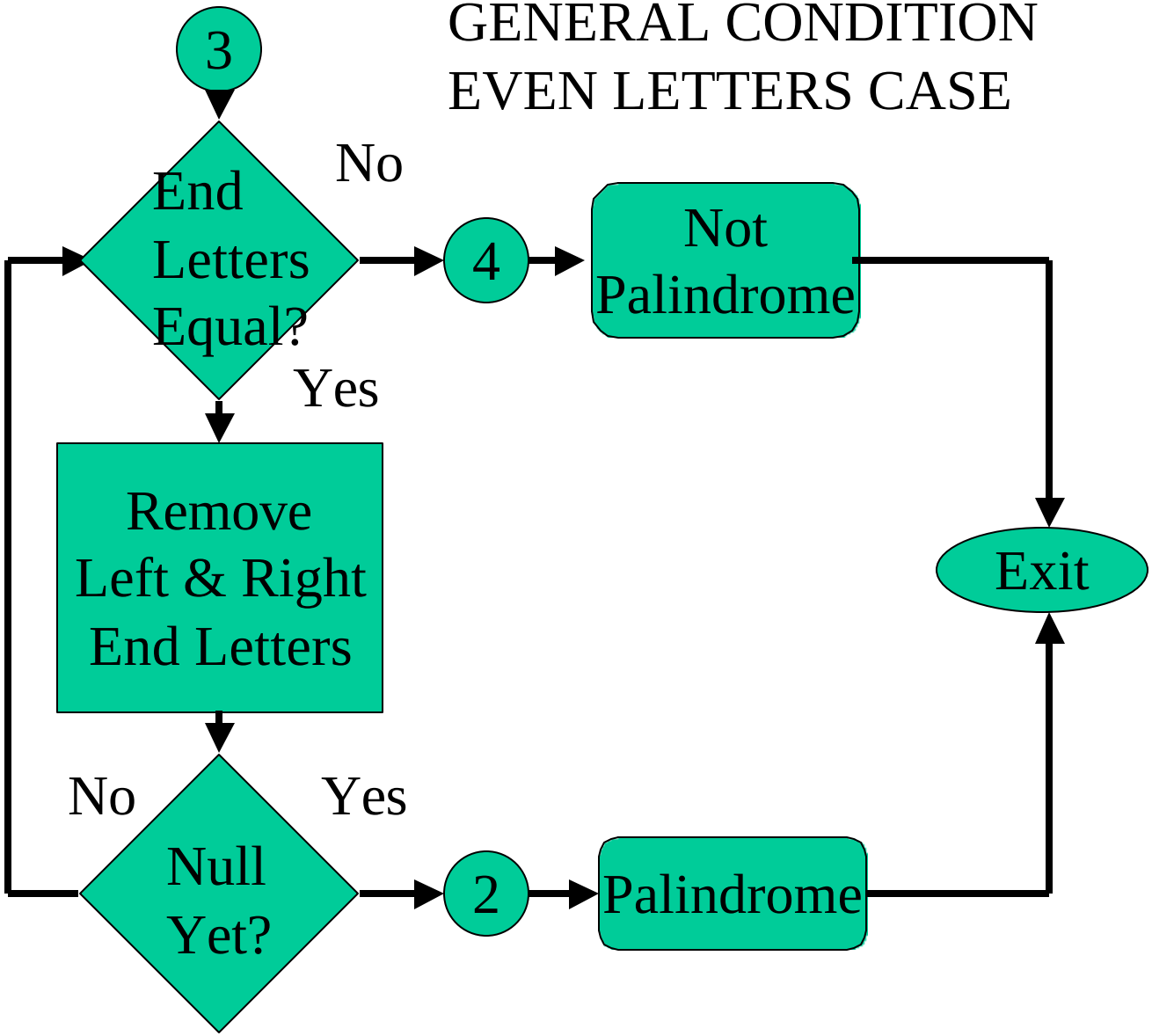
Exit Condition Met

# Iteration Test First




Enter
INITIALIZATION
MADAM
Input Word
DEED
Count Letters
Yes
Remove Middle Letter
Odd Long?
SPECIAL CONDITION
No
1
No
Yes
3
Null Yet?
2

# GENERAL CONDITION EVEN LETTERS CASE



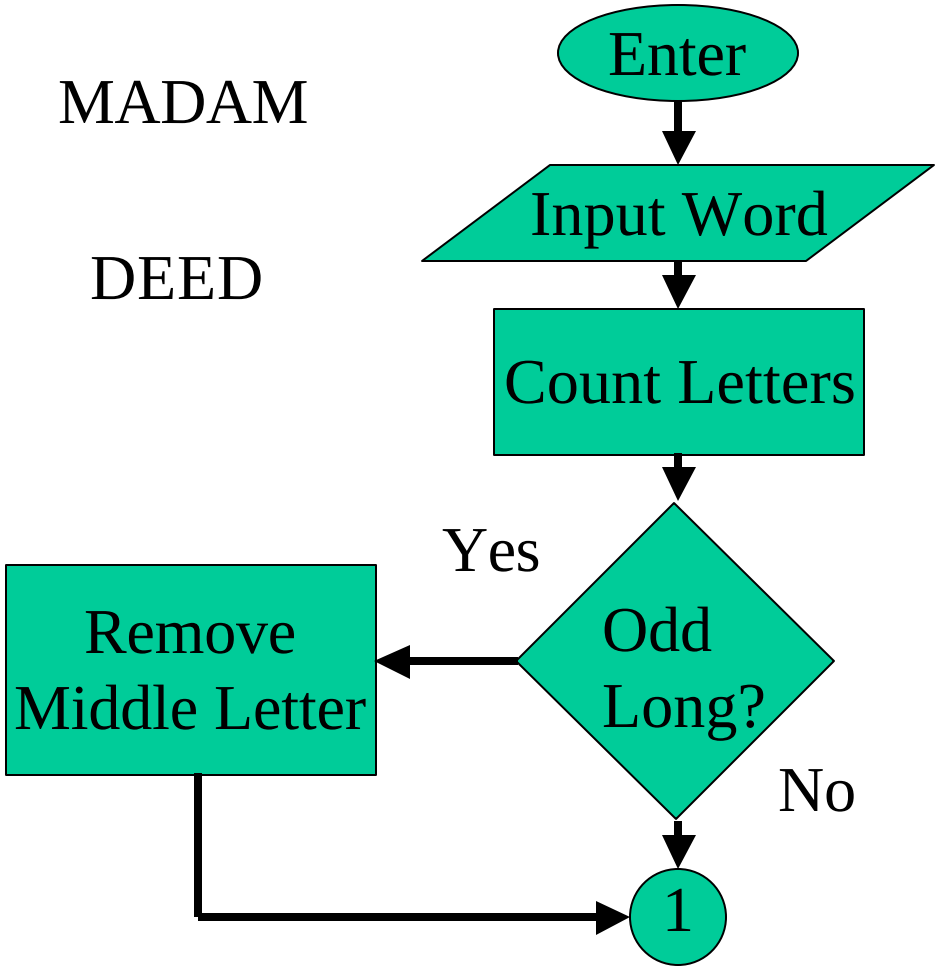
# Flow Chart Refinement

Goals:

- Reduce the number of tests
- Reduce the number of steps

MADAM

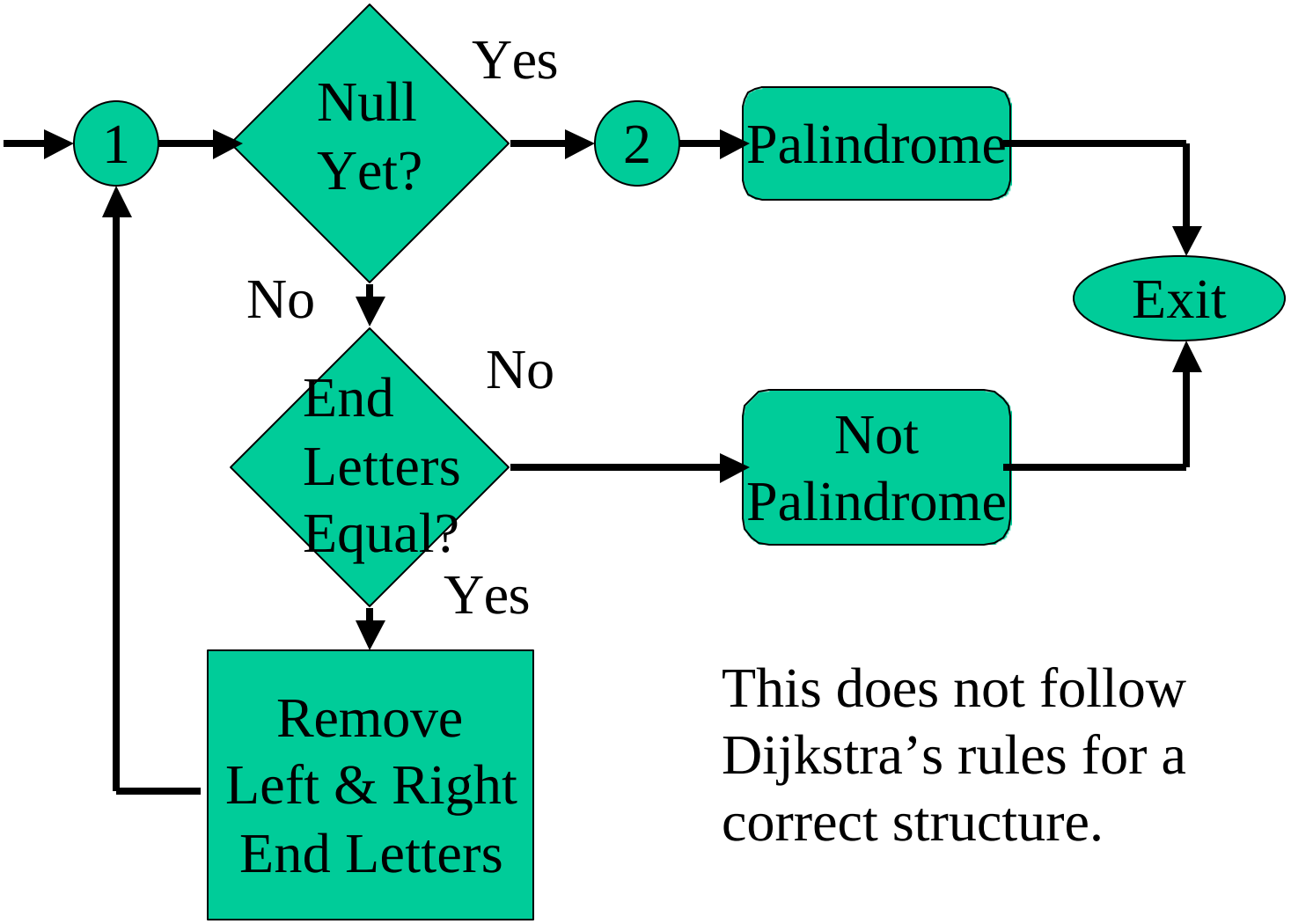
DEED




INITIALIZATION

SPECIAL CONDITION

# GENERAL CONDITION
# EVEN LETTERS CASE




This does not follow Dijkstra's rules for a correct structure.

# Program Correctness

- Dijkstra rules produce predictable behavior.
- Programs following these rules are “correct”.
  - Does not mean the right problem is solved.
  - Does not necessarily produce the most efficient solution.
- Led to structured programming languages.
  - PL/I, Pascal, modern Cobol and Fortran, C, C++

# Dijkstra

Do not use unconditional transfers.*

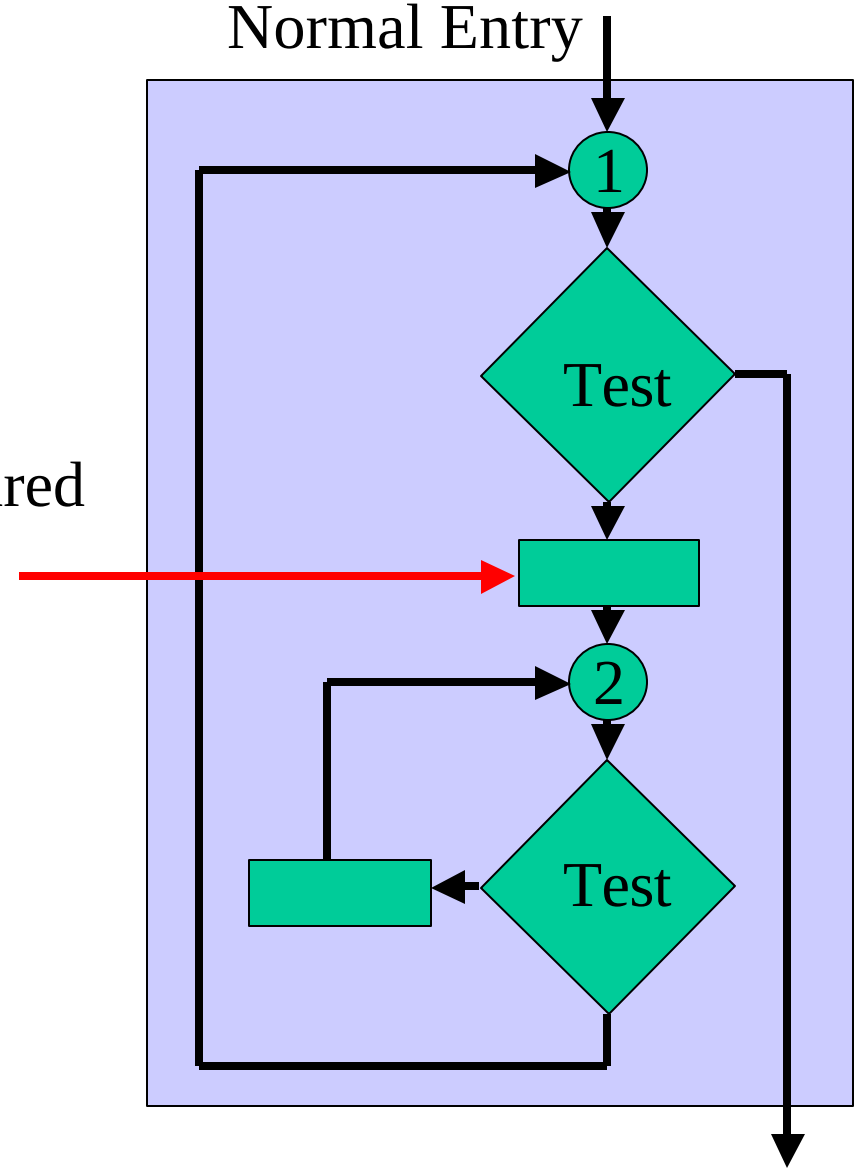
- These are called “GoTo” statements.
- Undisciplined use of GoTo statements produced unpredictable results or results difficult to understand.

* E. W. Dijkstra, “GoTo Statement Considered Harmful”, *Communications of the ACM*, 11 (3) 147-149 (March 1968)

# Example Difficulty

Nested Loops

Unstructured Entry

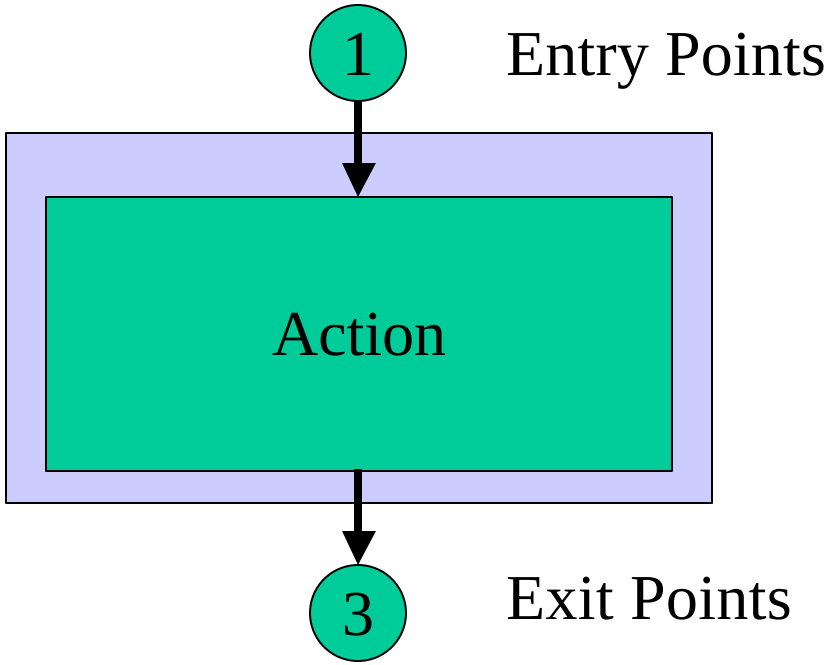



Unstructured Entry misses outer loop initialization.

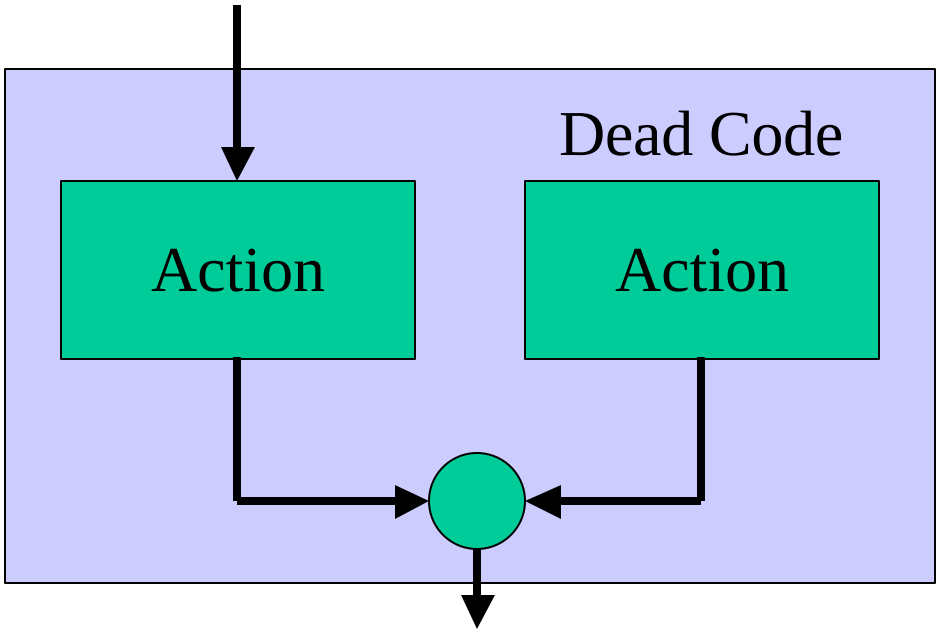
# Dijkstra's Rules for Correct Program Structure

- Well Defined Entry Points
- Well Defined Exit Points
- No Dead Code
- No Infinite Loops

# Well-Defined Entry and Exit Points



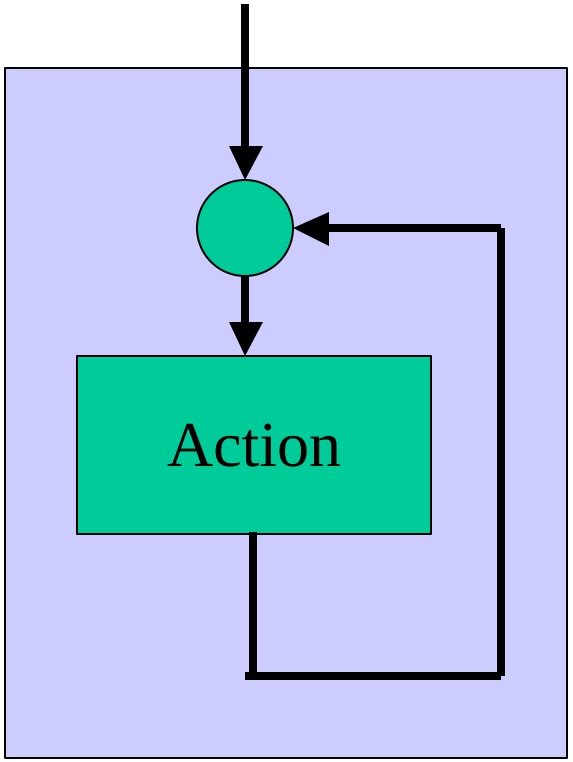
# No Dead Code



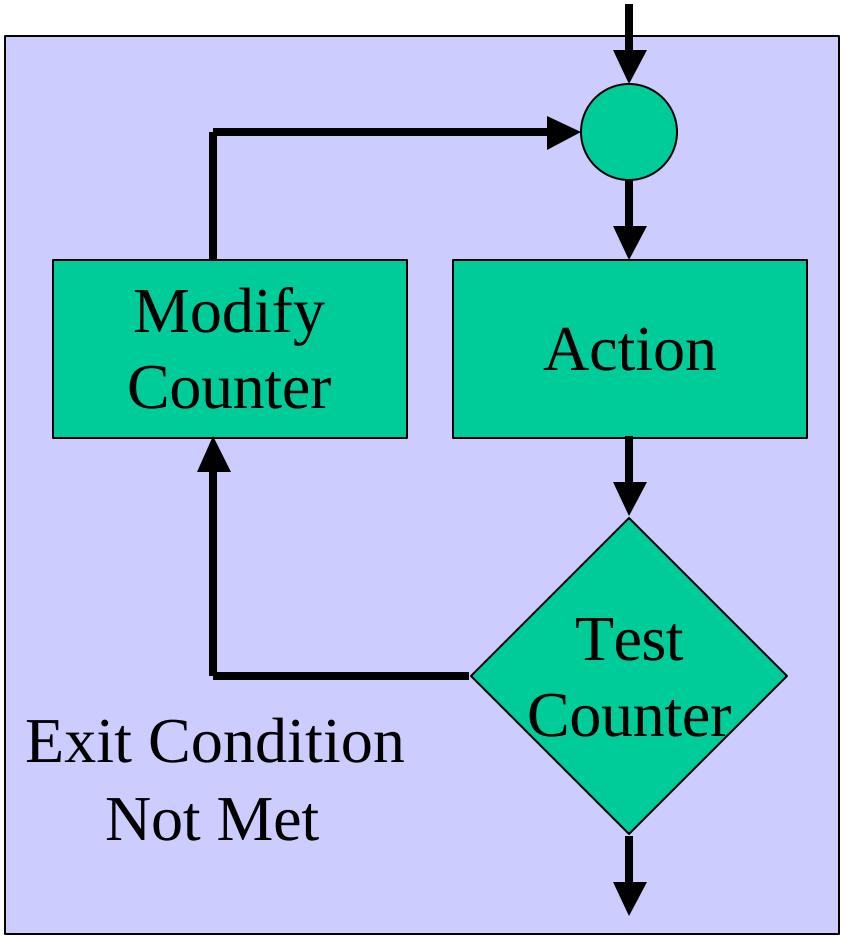

This is a problem when eliminating the decision from an If-Then-Else without removing the alternative code.

# No Infinite Loops

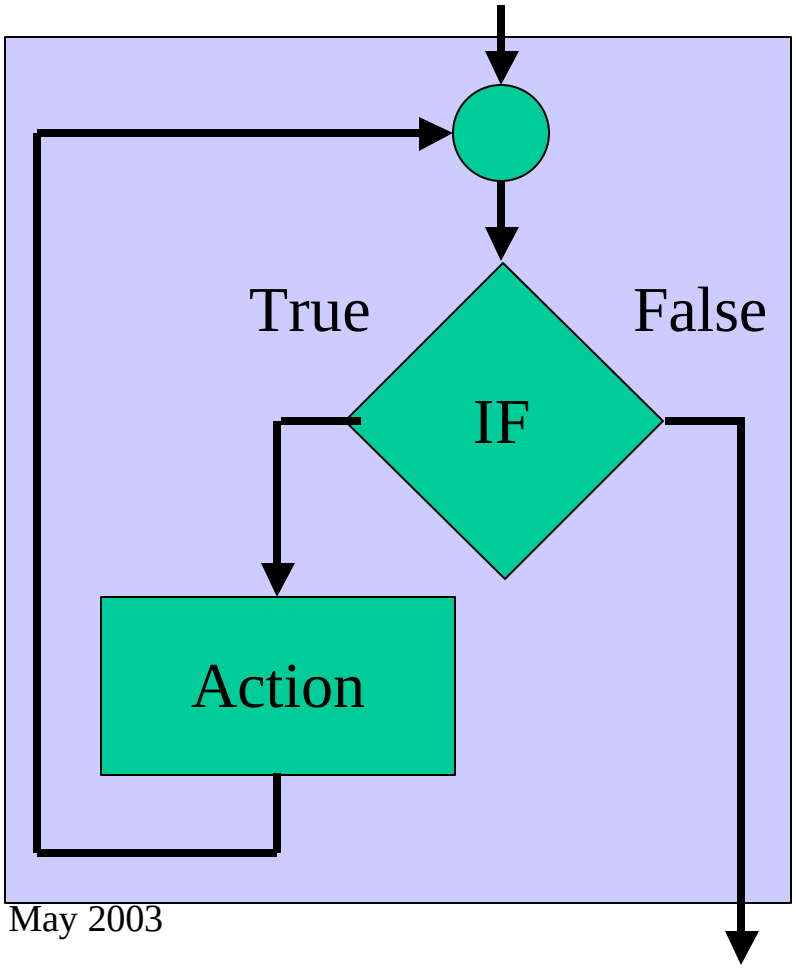
No terminating condition.



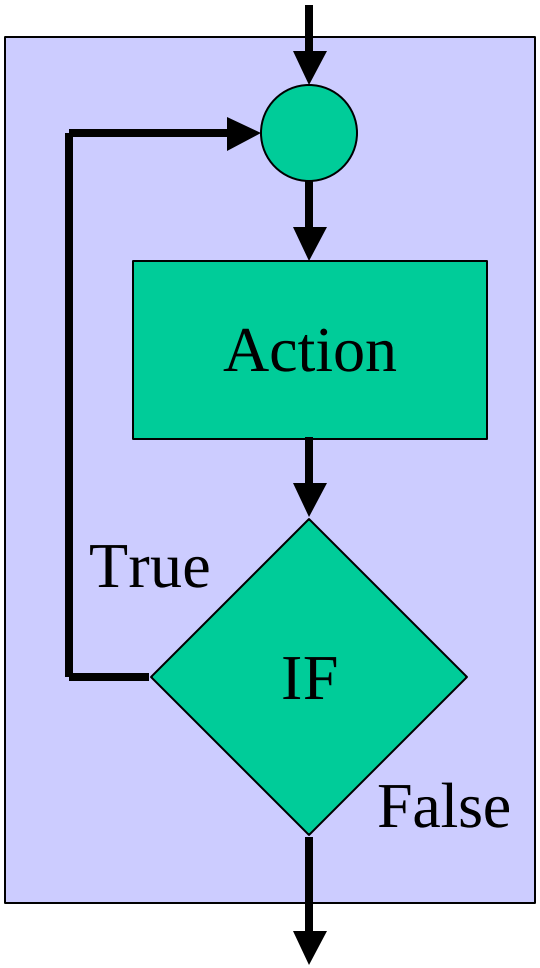

Terminating condition that can never be met.



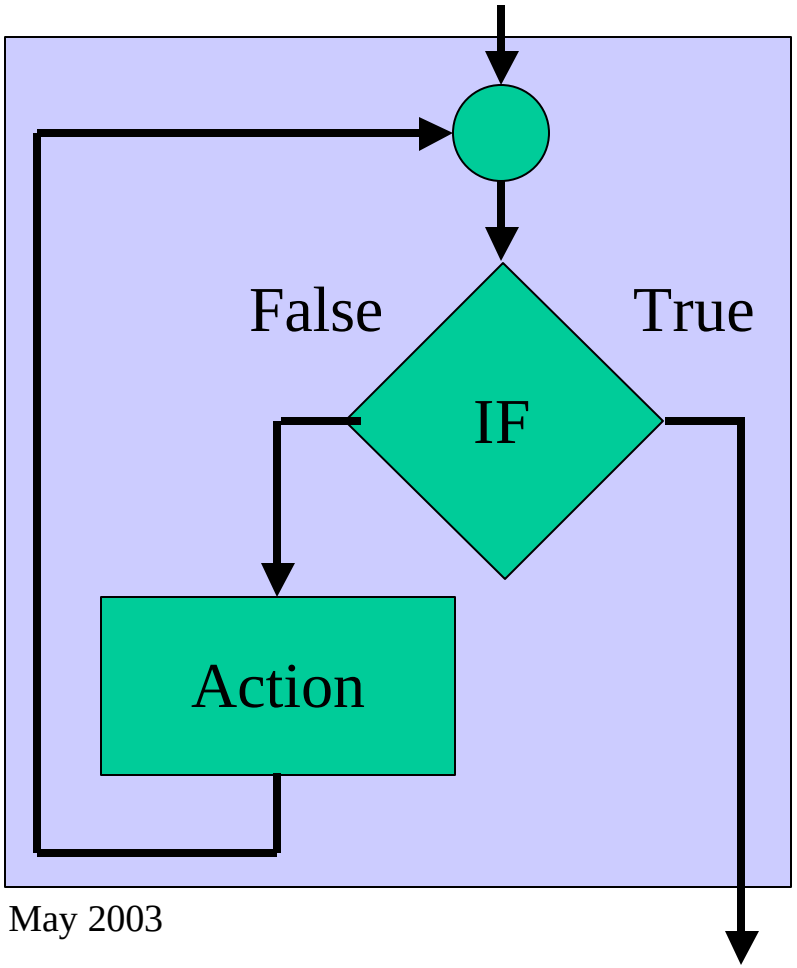
# DO WHILE



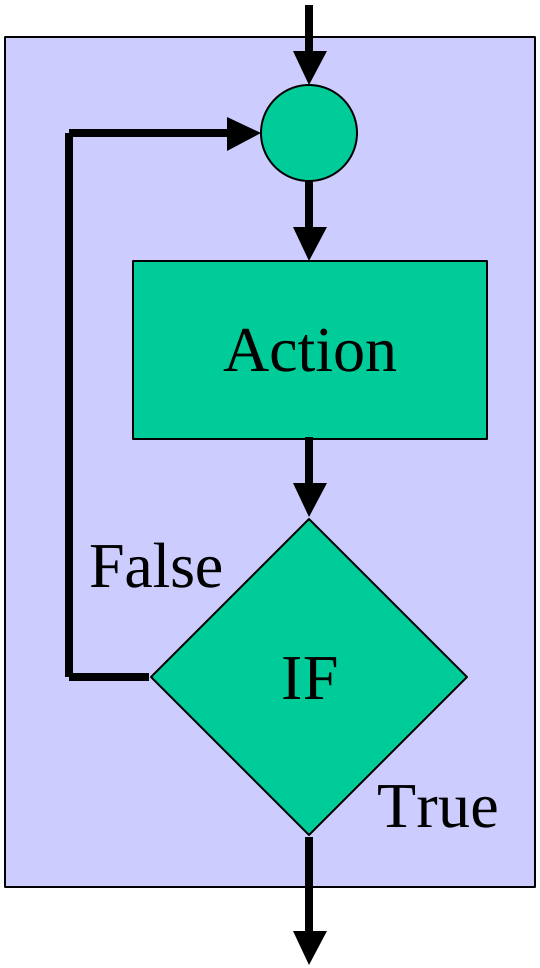

# CONTINUE WHILE



# DO UNTIL



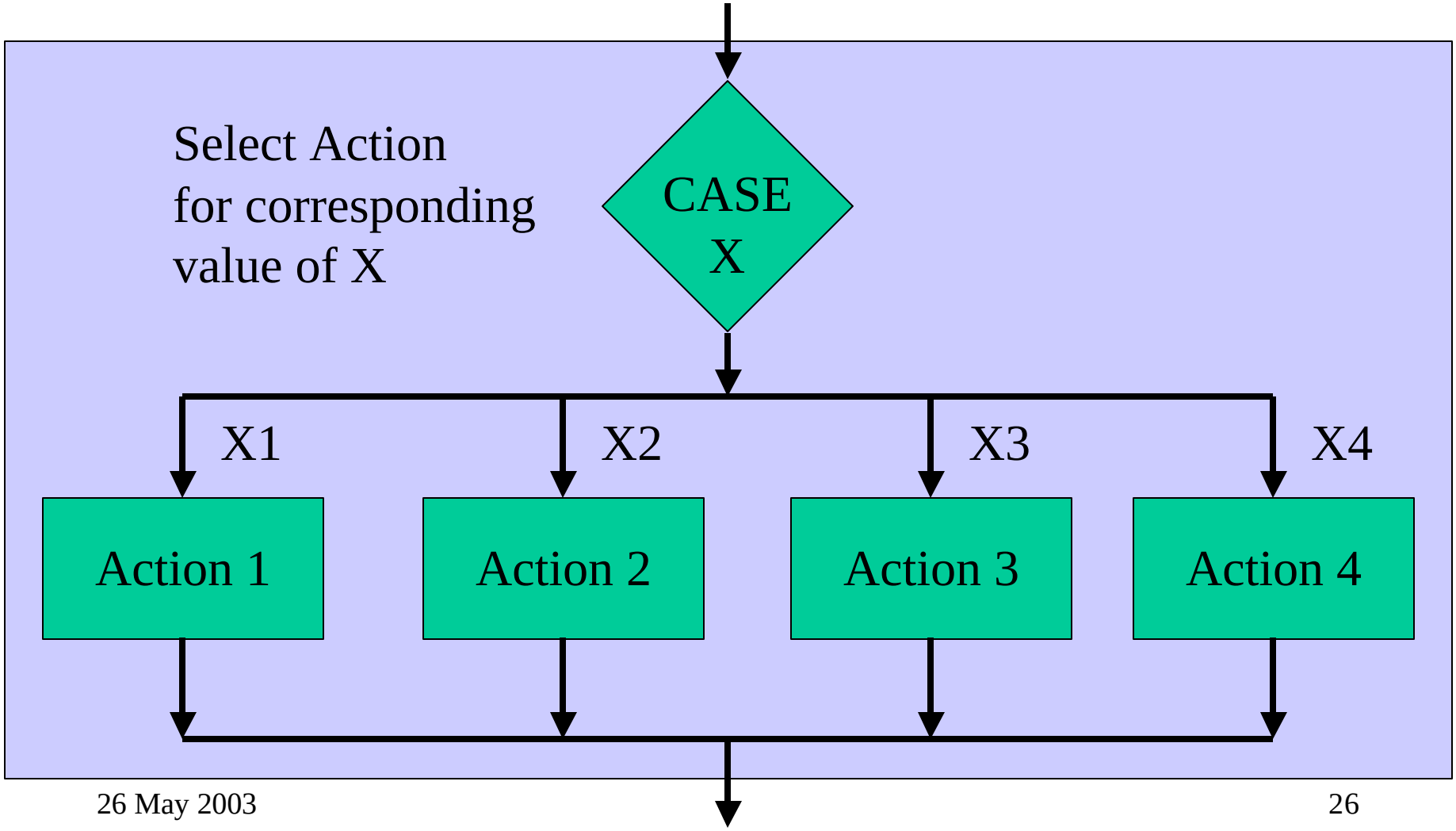

# CONTINUE UNTIL



# Terminology

- “Continue While” and “Continue Until” are non-standard terms.
- Issue: Whether the test condition occurs before or after the action.
- Texts are not in agreement.
- Microsoft Visual Basic for Applications does the test before the action in both cases.

# SELECT or CASE

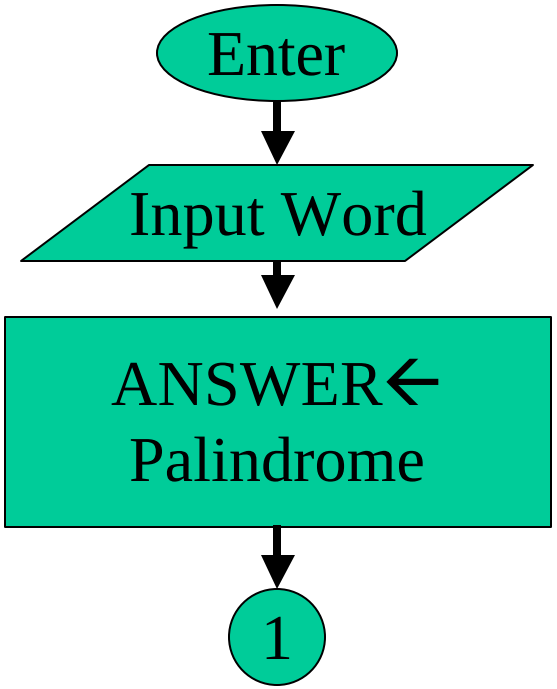


# Refine Flow Chart Again

Follow Dijkstra's Rules
for a Correctly Structured Program

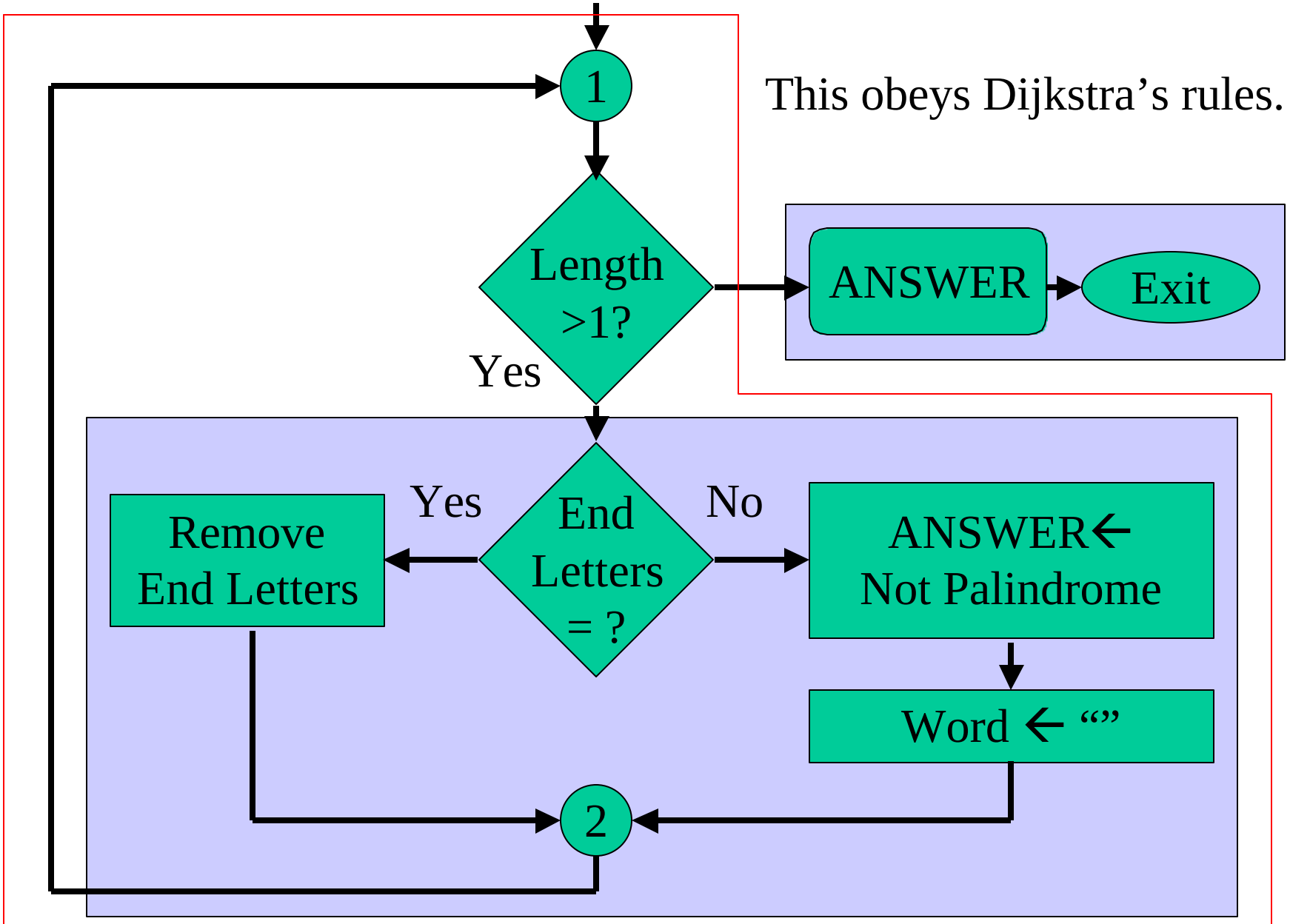
MADAM

DEED




INITIALIZATION

ASSUME PALINDROME




This obeys Dijkstra’s rules.

# Pseudocode for Palindrome Detection

```
ENTER
        INPUT word
        answer ← "Palindrome"
        DO WHILE Length(word) > 1
                IF Left_End_Letter = Right_End_Letter
                        THEN Remove_End_Letters
                        ELSE
                                answer ← "Not Palindrome"
                                word ← ""
                END IF
        LOOP
        PRINT answer
EXIT
```