

Designing Software With Flowcharts And Pseudo-code

In this section you will learn two different ways of laying out a computer algorithm independent of programming language

A Model For Creating Computer Software

Specify the problem

Develop a design (algorithm)

Implement the design

Maintain the design

What Is An Algorithm?

The steps needed to solve a problem

Characteristics

- Specific
- Unambiguous
- Language independent

Developing An Algorithm: Top-Down Approach

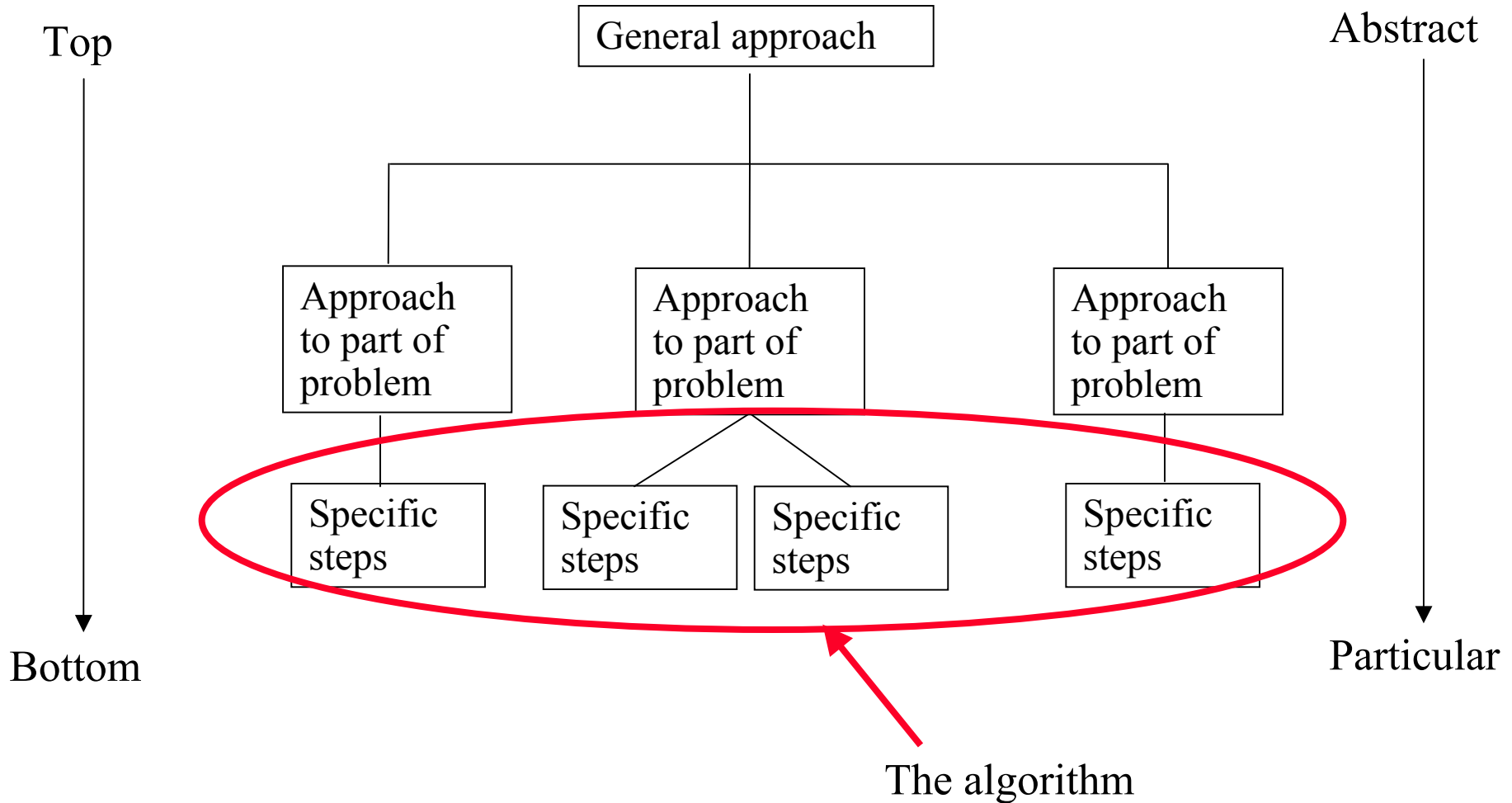


Figure extracted from Computer Science Illuminated by Dale N. and Lewis J.

Techniques For Laying Out An Algorithm

Pseudo-code

Flowcharts

Pseudo-Code

Employs 'programming-like' statements to depict the algorithm

No standard format (language independent)

Pseudo-Code Statements

Output

Input

Process

Decision

Repetition

Statements are carried out in order

Example: calling up a friend

- 1) Look up telephone number
- 2) Enter telephone number
- 3) Wait for someone to answer

: :

Variables

Are symbols used to store values

The value stored can change during the algorithm

Pseudo-Code: Output

Used to display information

General format:

Line of text: Output 'Message'

Variable: Output Name of variable

Example

Output 'Available credit limit: ' limit

Pseudo-Code: Input

Used to get information

Information is stored in a variable

General format:

Input: Name of variable

Example:

Input user_name

Pseudo-Code: Process

For computer programs it's usually an assignment statement (sets a variable to some value)

General form:

variable \leftarrow arithmetic expression

Example:

$x \leftarrow 2$

$x \leftarrow x + 1$

$a \leftarrow b * c$

Pseudo-Code: Decision Making

If-then

General form:

if (condition is met) then
statement(s)

Example:

if temperature < 0 then
wear a jacket

If-then-else

General form:

if (condition is met) then
statement(s)
else
statements(s)

Pseudo-Code: Decision Making (2)

Example:

if (at work) then

 Dress formally

else

 Dress casually

Pseudo-Code: Repetition

repeat-until

while-do

Pseudo-Code: Repetition (2)

repeat-until

Repeat at least once (check condition after statement(s))

General form:

repeat

 statement(s)

until (condition is met)

Example:

repeat

 Go up to buffet table

until full

Pseudo-Code: Repetition (3)

while-do

Repeat zero or more times (check condition before statement(s))

General form:

```
while (condition is met)  
    statement(s)
```

Example:

```
while students ask questions  
    Answer questions
```


Pseudo-Code: Fast Food Example

Use pseudo-code to specify the algorithm for a person who ordering food at a fast food restaurant. At the food counter, the person can either order not order the following items: a burger, fries and a drink. After placing her order the person then goes to the cashier.

Pseudo-Code: Fast Food Example

Approach counter

if want burger then

 order burger

if want fries then

 order fries

if want drink then

 order drink

Pay cashier

Pseudo-Code: Fast Food Example (Computer)

Approach counter

Output 'Order burger?'

Input order_burger

if order_burger = yes then

 order_burger

Output 'Order fries?'

Input order_fries

if order_fries = yes then

 order_fries

Pseudo-Code: Fast Food Example (Computer 2)

Output 'Order drink?'

Input order_drink

If order_drink = yes then

 order drink

Pay cashier

Pseudo-Code: ATM Example

Use pseudo-code to specify the algorithm for an ATM bank machine. The bank machine has four options: 1) Show current balance 2) Deposit money 3) Withdraw money 4) Quit. After an option has been selected, the ATM will continue displaying the four options to the person until he selects the option to quit the ATM.

Pseudo-Code: ATM Example

Approach ATM

Repeat

Output 'Select option'

Output '1) Make withdrawal'

Output '2) Make deposit'

Output '3) Show balance'

Output '4) Quit'

Input option

Pseudo-Code: ATM Example (2)

If option = deposit then

Output 'Enter amount to deposit'

Input amount

balance \leftarrow balance + amount

If option = withdrawal then

Output 'Enter amount to withdraw'

Input amount

balance \leftarrow balance – amount

Output 'Balance is ' balance

Until option = quit

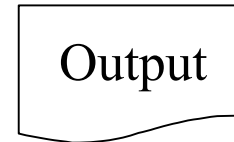
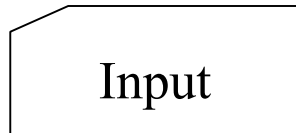
Stop

Can you spot the limitations of this algorithm?

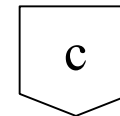
Summary Of Pseudo-Code Statements

Statement	Purpose
Output	Display information
Input	Get information
Process	Perform an atomic (non-divisible) activity
Decision	Choose between different alternatives
Repetition	Perform a step multiple times

Basic Flowcharts Element



Off page
Connector

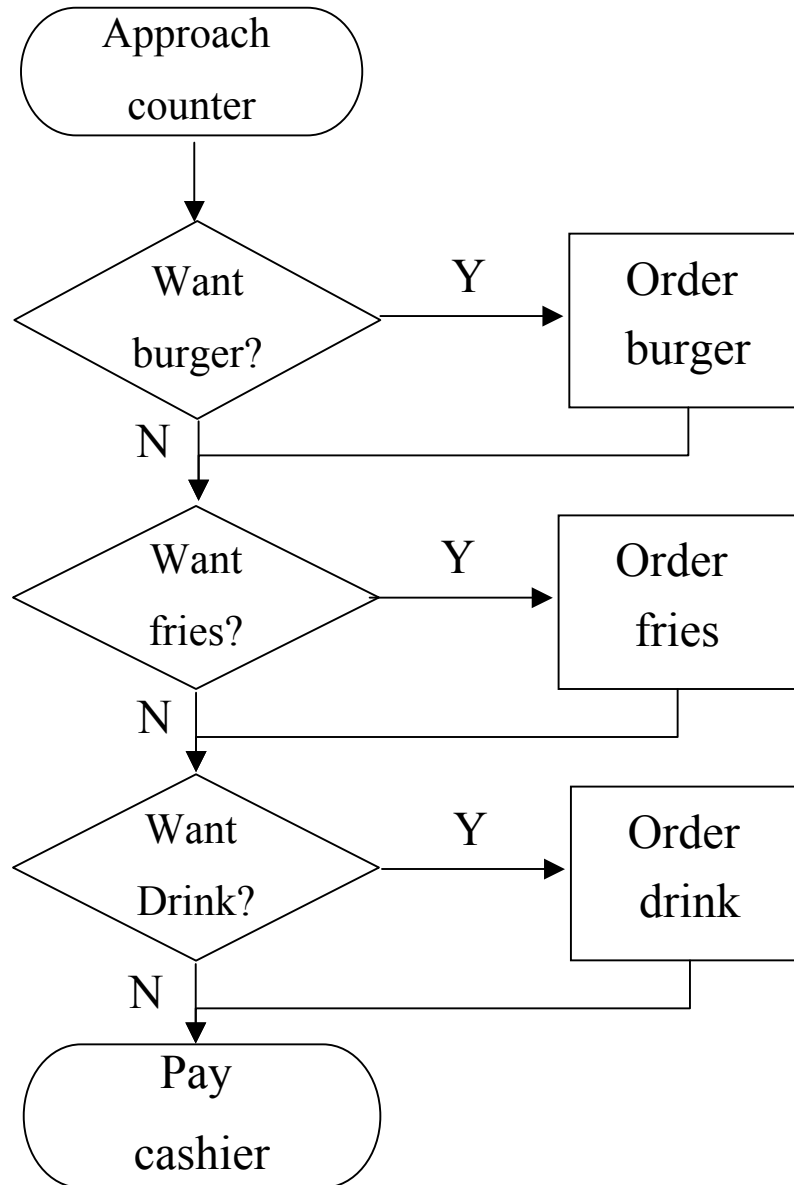


Variables

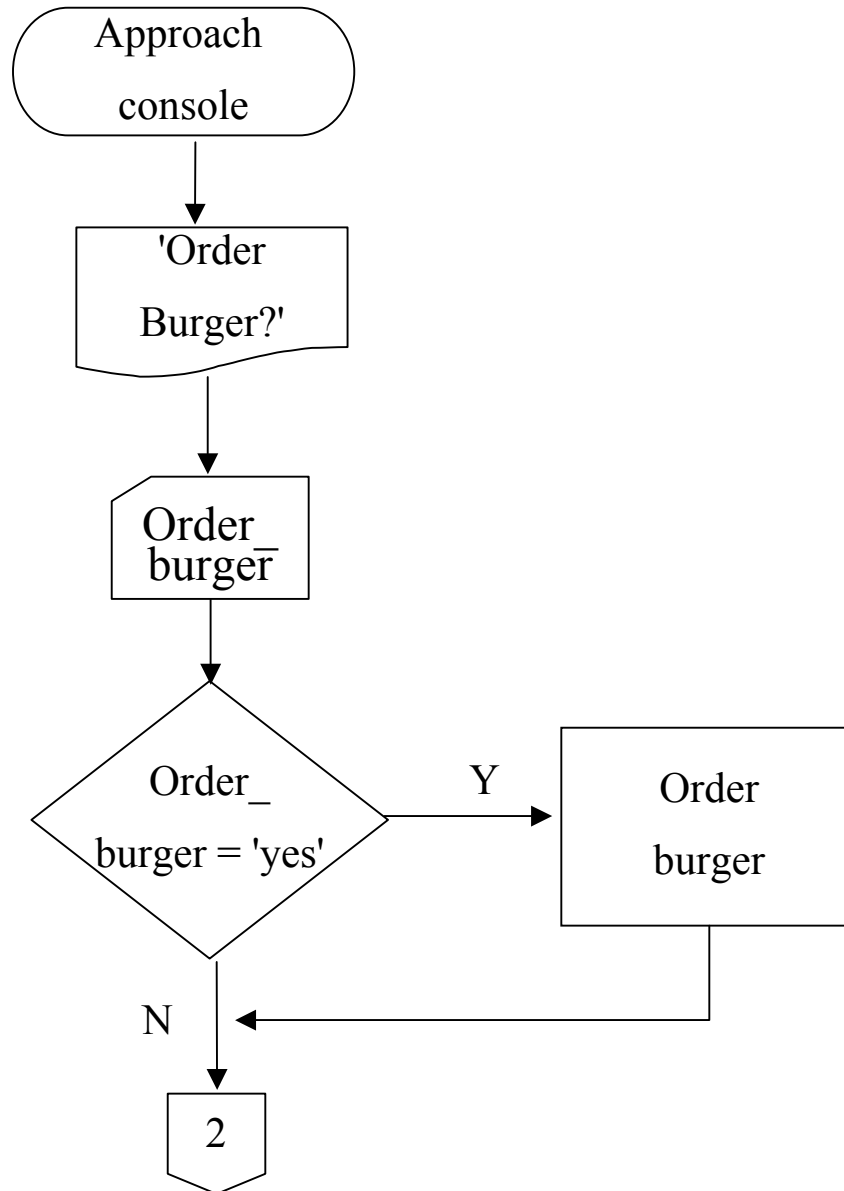
Flowchart: Fast Food Example

Draw a flowchart to outline the algorithm for a person who ordering food at a fast food restaurant. At the food counter, the person can either order not order the following items: a burger, fries and a drink. After placing her order the person then goes to the cashier.

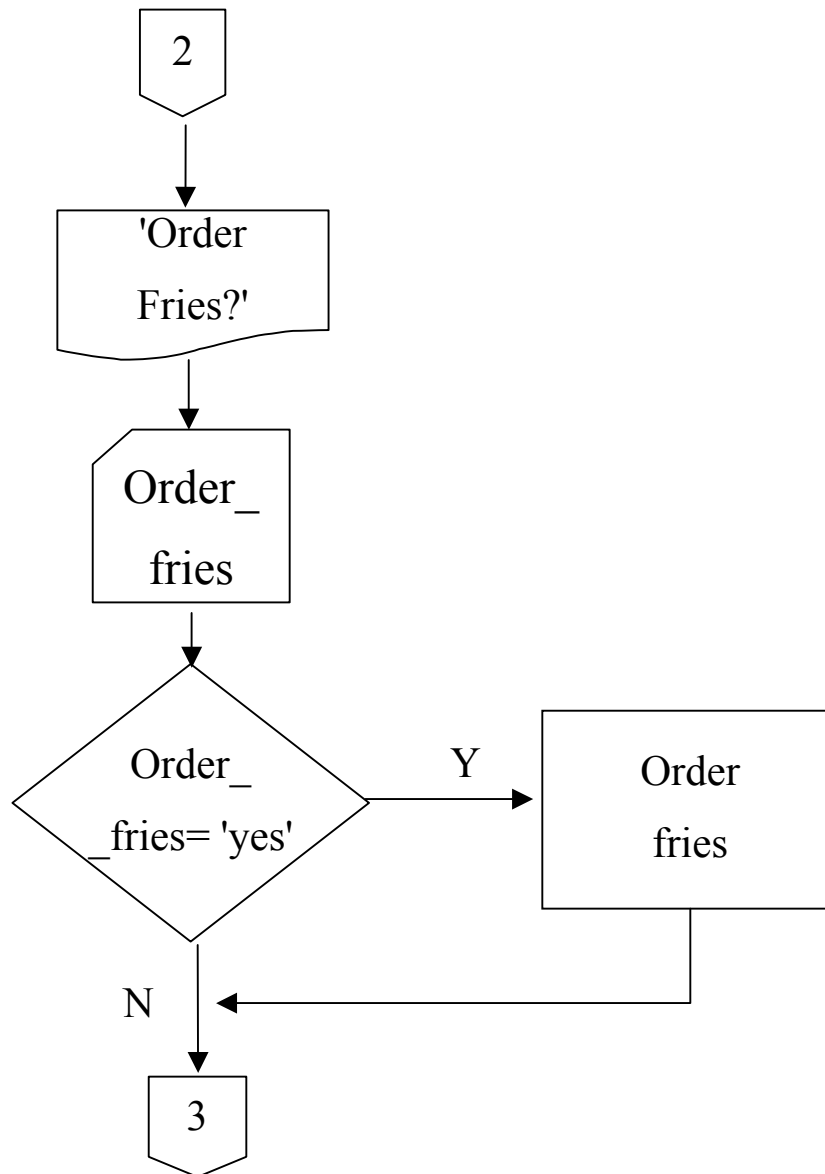
Flowchart: Fast Food Example



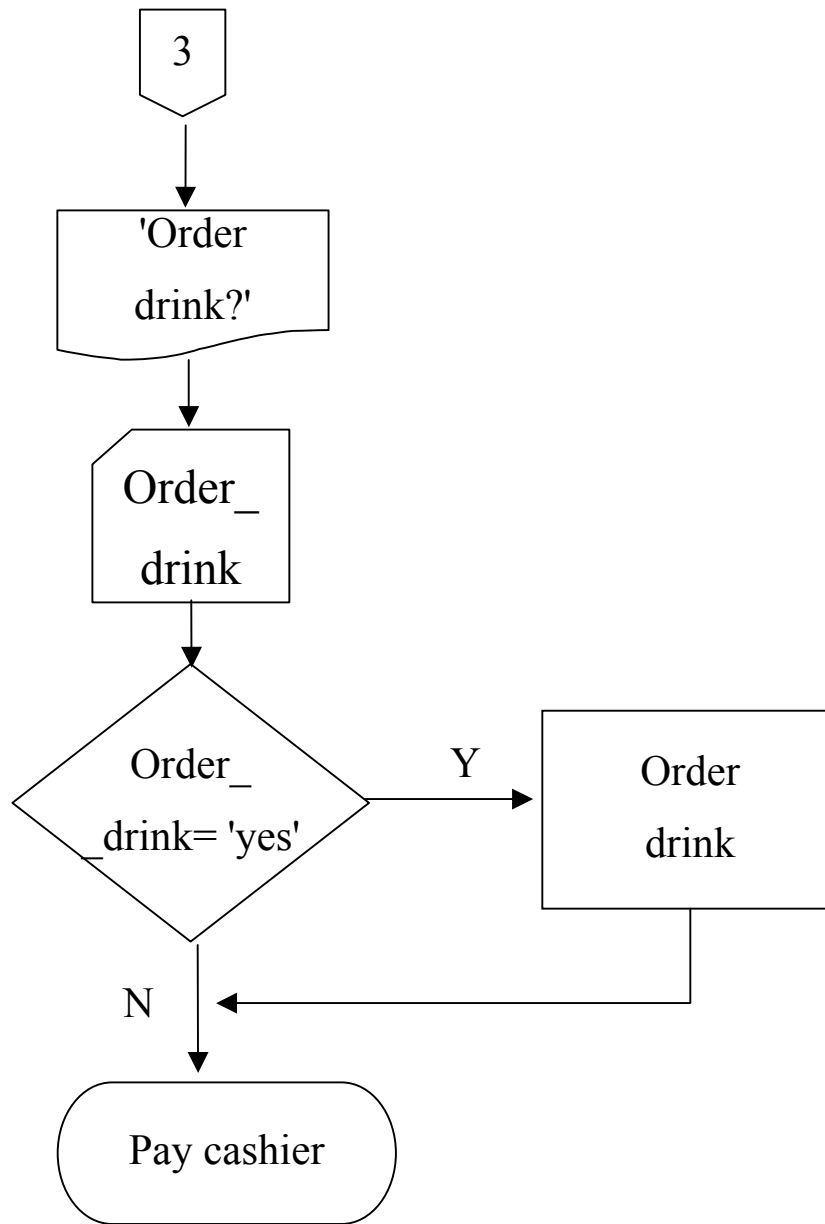
Flowchart: Fast Food Example (Computer)



Flowchart: Fast Food Example (Computer 2)



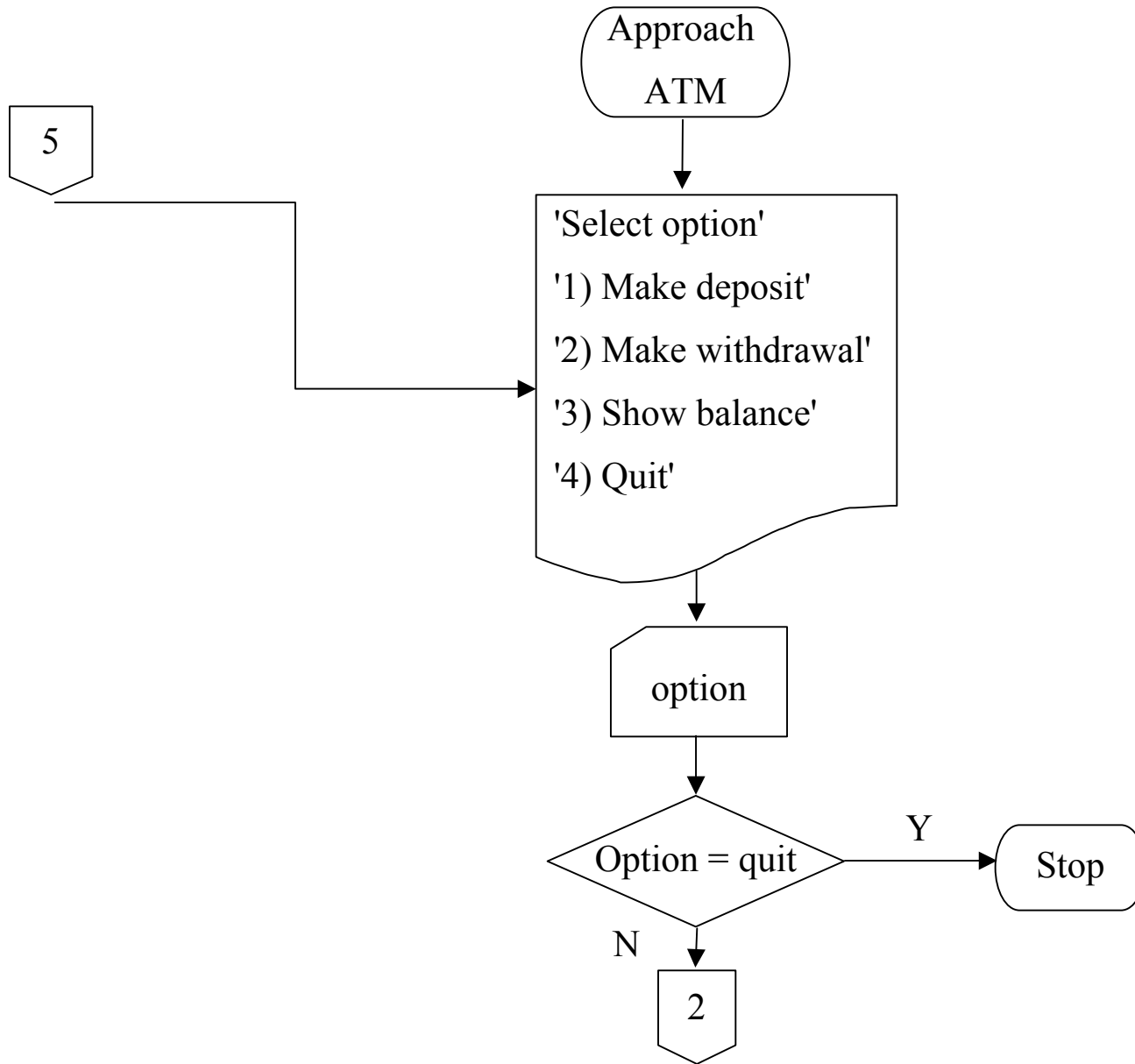
Flowchart: Fast Food Example (Computer 3)



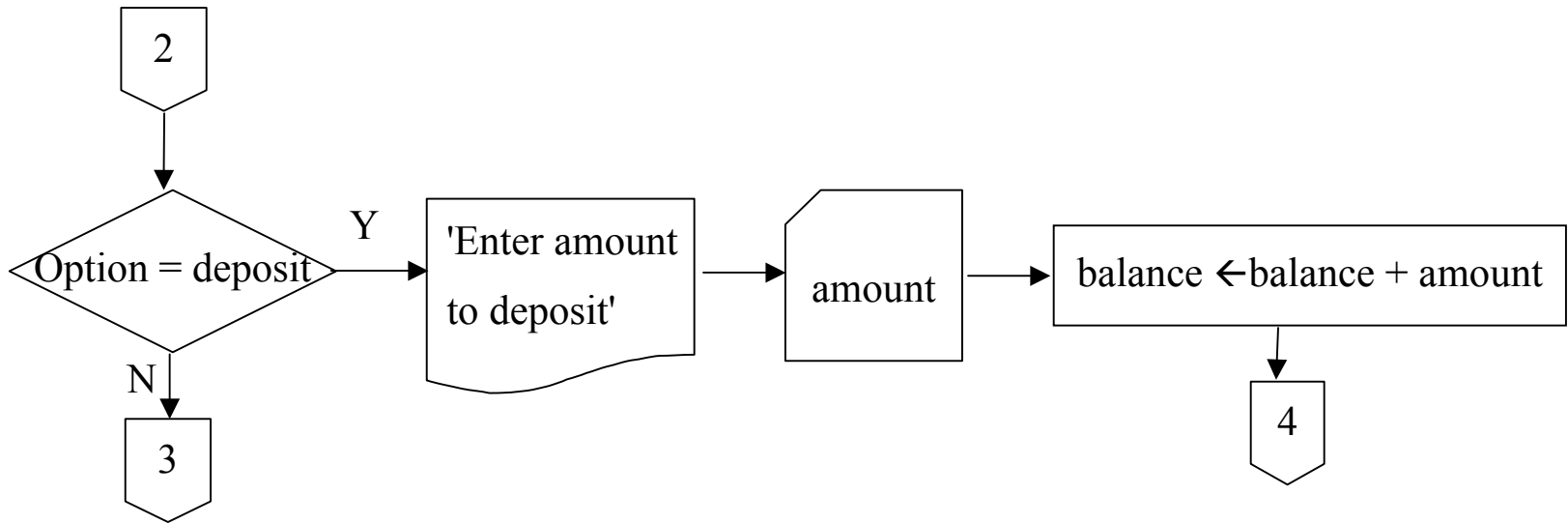
Flowchart: ATM Example

Draw a flowchart to outline the algorithm for an ATM bank machine. The bank machine has four options: 1) Show current balance 2) Deposit money 3) Withdraw money 4) Quit. After an option has been selected, the ATM will continue displaying the four options to the person until he selects the option to quit the ATM.

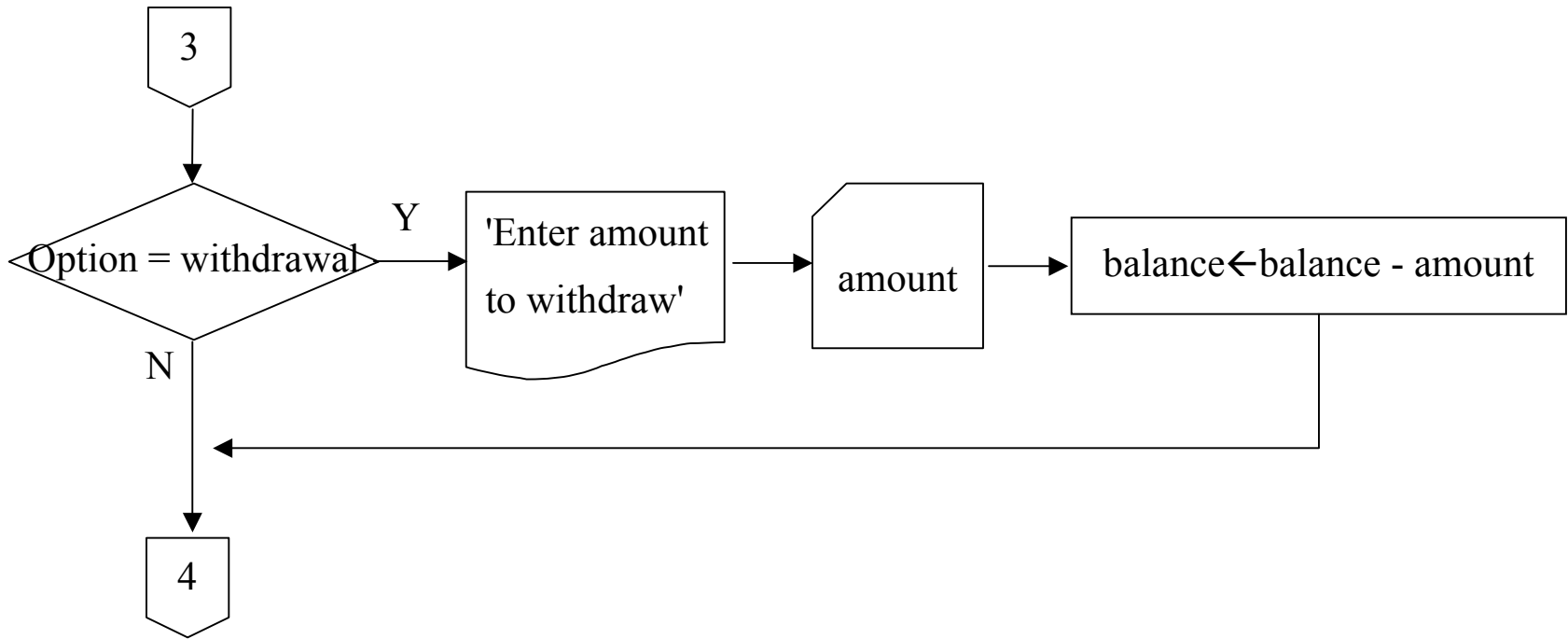
Flowchart: ATM Example



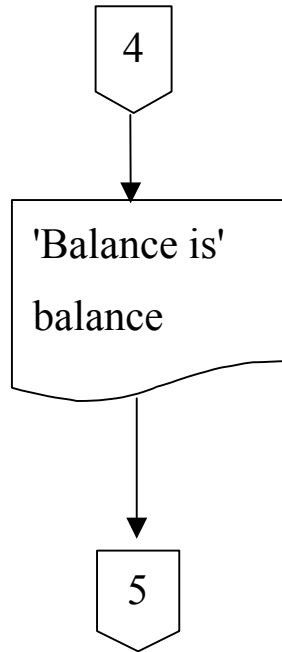
Flowchart: ATM Example (2)



Flowchart: ATM Example (3)



Flowchart: ATM Example (4)



Can you spot the limitations of this algorithm?

Summary

Laying out an algorithm using flowcharts and pseudo-code

Learning basic elements of algorithms:

- Input
- Output
- Decision-Making
- Repetition
- Processes