Cairo University

Programming and Problem Solving

Presented by Dr. Reda Zein
Prepared By Eng. / Samer Hany

Agenda

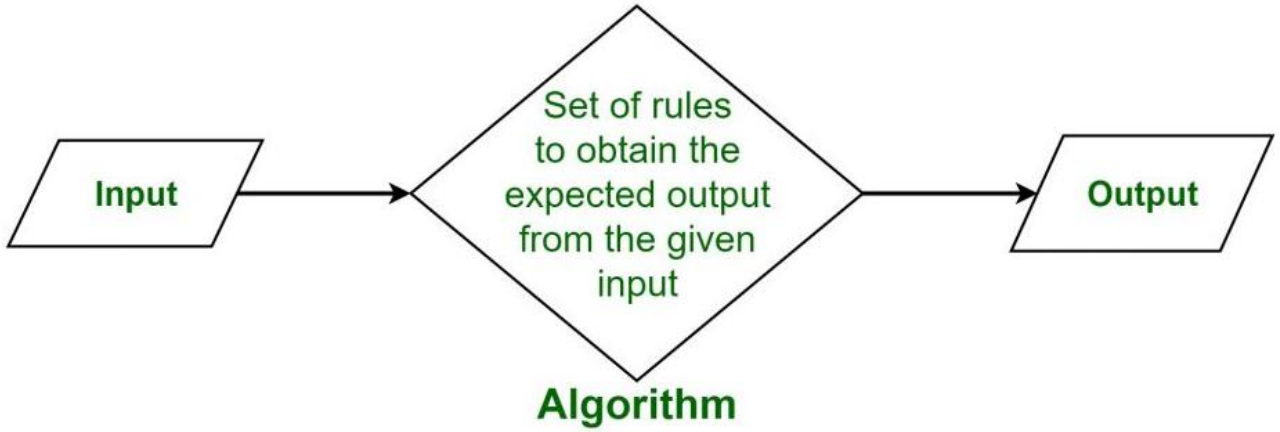
PROGRAMMING & PROBLEM SOLVING

- Algorithms
- Flowcharts
- Pseudocode

Algorithms

ALGORITHM

A process or set of steps to be followed in calculations or other problem-solving operations, especially by a computer.



Algorithms .. *their essential properties?*

The four essential properties of an algorithm:

i. ***Exact:*** **Precisely and unambiguously described, so that there remains no uncertainty.**

ii. **Terminate: The ultimate purpose of an algorithm is to solve a problem.**

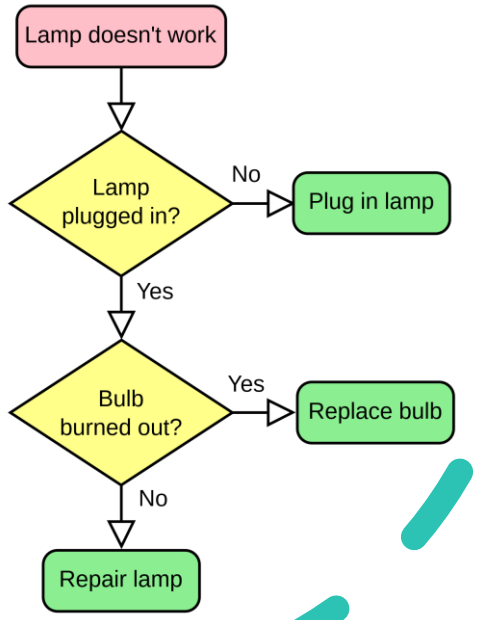
iii. ***Effective:*** **Must give the correct answer.**

iv. ***General:*** **Must solve every instance of the problem. E.g., a program that computes the area of a rectangle, within the limits of the programming language and the machine.**

Flowcharts

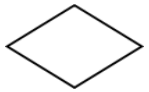
FLOWCHART

- A diagram that represents a process, system or computer algorithm.
- Example:
  Dealing with a non-functioning lamp



Flowcharts

BASIC ELEMENTS

Symbol	Purpose	Description
	Flow line	Indicates the flow of logic by connecting symbols.
	Terminal(Stop/Start)	Represents the start and the end of a flowchart.
	Input/Output	Used for input and output operation.
	Processing	Used for arithmetic operations and data-manipulations.
	Decision	Used for decision making between two or more alternatives.
	On-page Connector	Used to join different flowline
	Off-page Connector	Used to connect the flowchart portion on a different page.
	Predefined Process/Function	Represents a group of statements performing one processing task.

Pseudocode

PSEUDOCODE

A description of the implementation of an algorithm in plain language that humans can easily understand.

Pseudocode

ADVANTAGES OF PSEUDOCODE

- Improves the readability of any approach.
- Focuses on problem solving without worrying about the syntax of a specific programming language.
- Acts as a bridge between the program and the algorithm or flowchart.
- Works as a rough documentation of the program.

Pseudocode

STEPS TO WRITE PSEUDOCODE

1. Understand what the algorithm does.
2. Break the problem down into smaller parts.
3. Start by writing the purpose of the algorithm.
4. Write only one statement per line.
5. Use indentation to show hierarchy.
6. Capitalize key commands (e.g. IF, ELSE, etc.)
7. Use standard programming structures.
8. Keep it simple and concise.
9. Test your pseudocode logic before programming.

Pseudocode

STATEMENTS

- Assignment: ← or :=
- Comparison: =, ≠, <, >, ≤, ≥
- Arithmetic: +, −, ×, /, mod
- Logical: and, or

Pseudocode

KEYWORDS

- **START:** This is the start of your pseudocode.
- **INPUT:** This is data retrieved from the user through typing or through an input device.
- **READ / GET:** This is input used when reading data from a data file.
- **PRINT, DISPLAY, SHOW:** This will show your output to a screen or the relevant output device.
- **COMPUTE, CALCULATE, DETERMINE:** This is used to calculate the result of an expression.
- **SET, INIT:** To initialize values
- **INCREMENT:** To increase the value of a variable

Pseudocode

CONDITIONALS

- **IF – ELSE – ENDIF**
- Example:

```
INPUT time
IF time < 10 THEN
    PRINT “Good Morning”
ELSE IF time < 20 THEN
    PRINT “Good Day”
ELSE
    PRINT “Good Evening”
ENDIF
```

Pseudocode

ITERATION

- **FOR - ENDFOR**
- Example:

```
FOR each character in “HELLO WORLD!”
    PRINT character
ENDFOR
```

Pseudocode

ITERATION

- **WHILE - ENDWHILE**
- Example:

```
SET i := 1
SET sum := 0
WHILE sum < 20
    sum := sum + i^2
    INCREMENT i
ENDWHILE
PRINT i
PRINT sum
```

Pseudocode

FUNCTIONS

- To define a function:

```
FUNCTION calculate_sum (a, b)
    sum := a + b
    RETURN sum
END FUNCTION
```

- To call a function:

```
INPUT a, b
sum := calculate_sum (a, b)
PRINT sum
```

Pseudocode

EXCEPTIONS (ERROR HANDLING)

- For error handling:

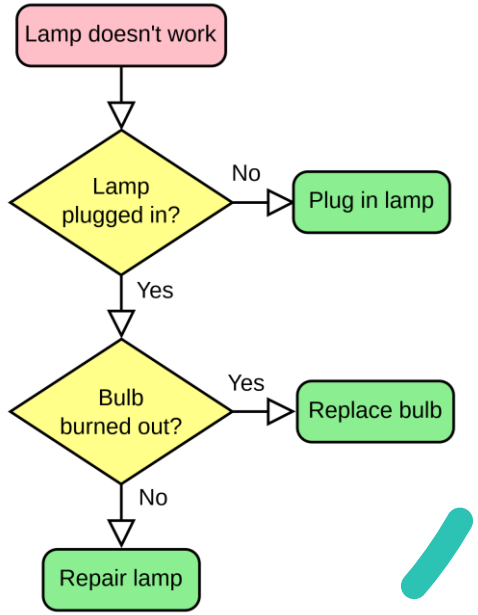
```
TRY
    INPUT birth_year
    age := this_year – birth_year
    PRINT age
CATCH
    PRINT “Please enter a valid date and try again”
END TRY
```

Pseudocode

EXAMPLE

- *Dealing with a non-functioning lamp*

```
// This program handles dealing
with a non-functioning lamp
START
IF lamp not plugged in
        plug in lamp
ELSE IF bulb burned out
        replace bulb
ELSE
        repair lamp
ENDIF
END
```



Pseudocode

FURTHER READINGS

- https://www.programiz.com/article/flowchart-programming
- https://en.wikipedia.org/wiki/Flowchart
- https://medium.com/@ngunyimacharia/how-to-write-pseudocode-a-beginners-guide-29956242698
- https://www.wikihow.com/Write-Pseudocode