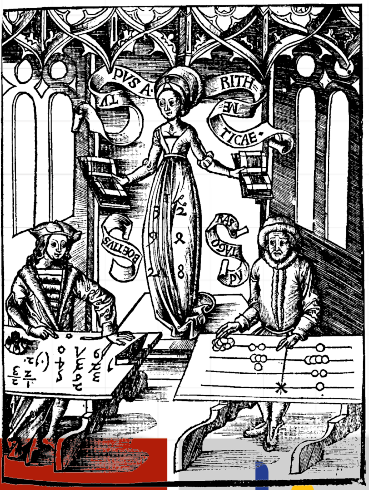


Algorithms (Part I)

ver. 13 z drobnymi modyfikacjami!

Wojciech Myszka

2023-11-13 15:42:44 +0100

HR EXCELLENCE IN RESEARCH

Wrocław University of Science and Technology

Short summary

We already know:

Short summary

We already know:

1. What the computer is.

Short summary

We already know:

1. What the computer is.
2. How it is built.

Short summary

We already know:

1. What the computer is.
2. How it is built.
3. What the processor is…

Short summary

We already know:

1. What the computer is.
2. How it is built.
3. What the processor is…
4. …and how it operates.

Short summary

We already know:

1. What the computer is.
2. How it is built.
3. What the processor is…
4. …and how it operates.
5. We understand (a little bit) that computers really need programs.

Short summary

We already know:

1. What the computer is.
2. How it is built.
3. What the processor is…
4. …and how it operates.
5. We understand (a little bit) that computers really need programs.
6. What the program is.

Short summary

We already know:

1. What the computer is.
2. How it is built.
3. What the processor is…
4. …and how it operates.
5. We understand (a little bit) that computers really need programs.
6. What the program is.

Short summary

We already know:

1. What the computer is.
2. How it is built.
3. What the processor is…
4. …and how it operates.
5. We understand (a little bit) that computers really need programs.
6. What the program is.

We do not know:

Short summary

We already know:

1. What the computer is.
2. How it is built.
3. What the processor is…
4. …and how it operates.
5. We understand (a little bit) that computers really need programs.
6. What the program is.

We do not know:

1. How to make a program…

How the software is made

The procedure looks like:

How the software is made

The procedure looks like:

- Interview with the customer.

How the software is made

The procedure looks like:

- Interview with the customer.
- Project and algorithm(s).

How the software is made

The procedure looks like:

- Interview with the customer.
- Project and algorithm(s).
- Programming

How the software is made

The procedure looks like:

- Interview with the customer.
- Project and algorithm(s).
- Programming
- Testing

How the software is made

The procedure looks like:

- Interview with the customer.
- Project and algorithm(s).
- Programming
- Testing
- Making the documentation

How the software is made

The procedure looks like:

- Interview with the customer.
- Project and algorithm(s).
- Programming
- Testing
- Making the documentation
- Installation and Deployment

How the software is made

The procedure looks like:

- Interview with the customer.
- Project and algorithm(s).
- Programming
- Testing
- Making the documentation
- Installation and Deployment
- Operation/Maintenance

How the software is made

The procedure looks like:

- Interview with the customer.
- Project and algorithm(s).
- Programming
- Testing
- Making the documentation
- Installation and Deployment
- Operation/Maintenance
- ...

How the software is made

The procedure looks like:

- Interview with the customer.
- Project and algorithm(s).
- Programming
- Testing
- Making the documentation
- Installation and Deployment
- Operation/Maintenance
- ...

Definition: Algorithm

The term *algorithm* is quite new…

Definition: Algorithm

The term *algorithm* is quite new…

The word *algorithm* is derived from the name of the Abbasid Persian polymath (mathematician), but also astronomer, astrologer and geographer Muḥammad ibn Mūsā al-Khwārizmī, who lived during the eight/ninth century, and who is credited with providing the step-by-step rules for adding, subtracting, multiplying, and dividing ordinary **decimal numbers**. When translatted into Latin, the name became Algorismus.

Definition: Algorithm

The term *algorithm* is quite new…

The word *algorithm* is derived from the name of the Abbasid Persian polymath (mathematician), but also astronomer, astrologer and geographer Muḥammad ibn Mūsā al-Khwārizmī, who lived during the eight/ninth century, and who is credited with providing the step-by-step rules for adding, subtracting, multiplying, and dividing ordinary **decimal numbers**. When translatted into Latin, the name became Algorismus.

Algorithm:

A step-by-step problem-solving procedure, especially an established, recursive computational procedure for solving a problem in a finite number of steps. Sometimes we give up the requirement for **finiteness**.

Algorithm

In mathematics, computing, linguistics, and related subjects, an **algorithm** is a sequence of **finite instructions**, often used for calculation and data processing. It is formally a type of **effective** method in which a list of **well-defined** instructions for completing a task will, when given an **initial state**, proceed through a well-defined series of successive states, eventually **terminating** in an end-state. The transition from one state to the next is not necessarily deterministic; some algorithms, known as probabilistic algorithms, incorporate randomness.

Recipe I

- **The ingredients** include 8 ounces of semisweet chocolate pieces, 2 tablespoons of water, a 1/4 cup of powdered sugar, 6 separated eggs, and so on.
- **Recipe:**
  Melt chocolate and 2 tablespoons of water in a double boiler. When melted, stir in powdered sugar; add butter bit by bit. Set aside. Beat egg yolks until thick and lemon-coloured, about 5 minutes. Gently fold in chocolate. Reheat slightly to melt chocolate, if necessary. Stir in rum and vanilla. Beat egg whites until foamy. Beat in 2 tablespoons sugar; beat until stiff peaks form.

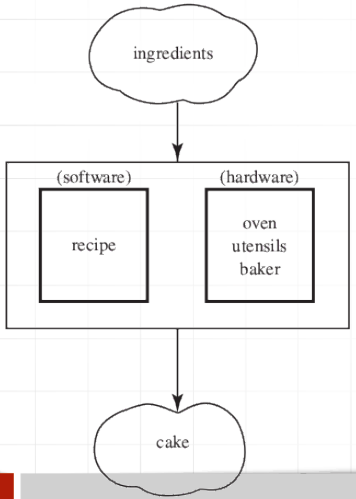
Recipe II

Gently fold whites into the chocolate-yolk mixture. Pour into individual serving dishes. Chill at least 4 hours. Serve with whipped cream, if desired. Makes 6 to 8 servings.

From: *French Cooking* [2]

Recipe

Hardware vs. Software [1]



Short algorithm

- We are given a list of personnel records (one for each employee, each containing the employee's name, personal details, and salary.)
- We are interested in the total sum of all salaries of all employees.
- Here is an algorithm for carrying out this task:
  1. make a note of the number 0;
  2. proceed through the list, adding each employee's salary to the noted number;
  3. having reached the end of the list, produce the noted number as output.

Short algorithm

Some comments

1. Is the algorithm correct?

Short algorithm

Some comments

1. Is the algorithm correct?
2. The *text* of the algorithm is short and fixed length.

Short algorithm

Some comments

1. Is the algorithm correct?
2. The *text* of the algorithm is short and fixed length.
3. The *process* algorithm describes varies with the length of data (number of personal records).

Short algorithm

Some comments

1. Is the algorithm correct?
2. The *text* of the algorithm is short and fixed length.
3. The *process* algorithm describes varies with the length of data (number of personal records).
4. Algorithm solves **each problem** of “this class”: two companies, the first with one employee and the second with a million, can both feed their employee list into the same algorithm, and the salary summation problem will be solved equally well for each.

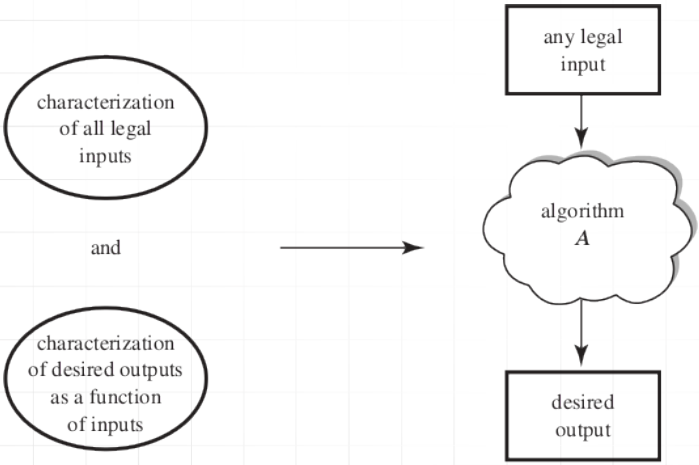
Short algorithm

Some comments

1. Is the algorithm correct?
2. The *text* of the algorithm is short and fixed length.
3. The *process* algorithm describes varies with the length of data (number of personal records).
4. Algorithm solves **each problem** of “this class”: two companies, the first with one employee and the second with a million, can both feed their employee list into the same algorithm, and the salary summation problem will be solved equally well for each.
5. . . .

Algorithmic solution

[1]



Euclidean algorithm

When: Somewhere between 400 and 300 B.C.,

Who: the great Greek mathematician Euclid (who first described it in his Elements)

What: described an algorithm for finding the greatest common divisor (**gcd**) of two positive integers. The **gcd** of *m* and *n* is the largest integer that exactly divides both *m* and *n*. **Euclidean algorithm** (as it is called) is considered to be the first non-trivial algorithm ever devised.

Euclidian algorithm

Having two positive integers m and n, find ***the greatest common divisor (gcd)***

Euclidian algorithm

Having two positive integers m and n, find ***the greatest common divisor (gcd)***

1. *[Find the reminder] Divide m by n let r be the reminder. (We have* $0 \leq r < n$*.)*

Euclidian algorithm

Having two positive integers m and n, find ***the greatest common divisor (gcd)***

1. *[Find the reminder] Divide m by n let r be the reminder. (We have* $0 \leq r < n$*.)*
2. *[Is zero?] If* $r = 0$ *finish the procedure; the answer is n.*

Euclidian algorithm

Having two positive integers m and n, find ***the greatest common divisor (gcd)***

1. *[Find the reminder] Divide m by n let r be the reminder. (We have* $0 \leq r < n$*.)*
2. *[Is zero?] If* $r = 0$ *finish the procedure; the answer is n.*
3. *[Simplifying] Let* $m \leftarrow n$, $n \leftarrow r$ *and came back to step 1.*

Reminder

$$\frac{5}{3} =$$

Reminder

$$\frac{5}{3} = 1\frac{2}{3}$$

Reminder

$$\frac{5}{3} = 1\frac{2}{3} = 1.6666(6)$$

Reminder

$$\frac{5}{3} = 1\frac{2}{3} = 1.6666(6) \overset{0.0001}{=} 1.6667$$

Reminder

$$\frac{5}{3} = 1\frac{2}{3} = 1.6666(6) \overset{0.0001}{=} 1.6667 = 1 \quad \text{with reminder} \quad 2$$

Reminder

$$\frac{5}{3} = 1\frac{2}{3} = 1.6666(6) \overset{0.0001}{=} 1.6667 \quad = 1 \quad \text{with reminder} \quad 2$$

because

$$1 \times 3 + 2 = 5$$

Euclidean Algorithm

Example

m	n	r
24	44	

1. [Find the reminder] Divide m by n let r be the divider. (We have $0 \leq r < n$.)
2. [Is zero?] If $r = 0$ finish the procedure; the answer is n.
3. [Simplifying] Let $m \leftarrow n$, $n \leftarrow r$ and came back to step 1.

Euclidean Algorithm

Example

m	n	r
24	44	24

24/44 = 0 r 24

1. [Find the reminder] Divide m by n let r be the divider. (We have $0 \leq r < n$.)
2. [Is zero?] If $r = 0$ finish the procedure; the answer is n.
3. [Simplifying] Let $m \leftarrow n$, $n \leftarrow r$ and came back to step 1.

Euclidean Algorithm

Example

m	n	r
24	44	24

1. [Find the reminder] Divide m by n let r be the divider. (We have $0 \leq r < n$.)
2. [Is zero?] If $r = 0$ finish the procedure; the answer is n.
3. [Simplifying] Let $m \leftarrow n$, $n \leftarrow r$ and came back to step 1.

Euclidean Algorithm

Example

m	n	r
24	44	24
44		

1. [Find the reminder] Divide m by n let r be the divider. (We have $0 \leq r < n$.)
2. [Is zero?] If $r = 0$ finish the procedure; the answer is n.
3. [Simplifying] Let $m \leftarrow n$, $n \leftarrow r$ and came back to step 1.

Euclidean Algorithm

Example

m	n	r
24	44	24
44	24	

1. [Find the reminder] Divide m by n let r be the divider. (We have $0 \leq r < n$.)
2. [Is zero?] If $r = 0$ finish the procedure; the answer is n.
3. [Simplifying] Let $m \leftarrow n$, $n \leftarrow r$ and came back to step 1.

Euclidean Algorithm

Example

m	n	r
24	44	24
44	24	20

44/24 = 1 r 20

1. [Find the reminder] Divide m by n let r be the divider. (We have $0 \leq r < n$.)
2. [Is zero?] If $r = 0$ finish the procedure; the answer is n.
3. [Simplifying] Let $m \leftarrow n$, $n \leftarrow r$ and came back to step 1.

Euclidean Algorithm

Example

m	n	r
24	44	24
44	24	20

1. [Find the reminder] Divide m by n let r be the divider. (We have $0 \leq r < n$.)
2. [Is zero?] If $r = 0$ finish the procedure; the answer is n.
3. [Simplifying] Let $m \leftarrow n$, $n \leftarrow r$ and came back to step 1.

Euclidean Algorithm

Example

m	n	r
24	44	24
44	24	20
24		

1. [Find the reminder] Divide m by n let r be the divider. (We have $0 \leq r < n$.)
2. [Is zero?] If $r = 0$ finish the procedure; the answer is n.
3. [Simplifying] Let $m \leftarrow n$, $n \leftarrow r$ and came back to step 1.

Euclidean Algorithm

Example

m	n	r
24	44	24
44	24	20
24	20	

1. [Find the reminder] Divide m by n let r be the divider. (We have $0 \leq r < n$.)
2. [Is zero?] If $r = 0$ finish the procedure; the answer is n.
3. [Simplifying] Let $m \leftarrow n$, $n \leftarrow r$ and came back to step 1.

Euclidean Algorithm

Example

m	n	r
24	44	24
44	24	20
24	20	4

24/20 = 1 r 4

1. [Find the reminder] Divide m by n let r be the divider. (We have $0 \leq r < n$.)
2. [Is zero?] If $r = 0$ finish the procedure; the answer is n.
3. [Simplifying] Let $m \leftarrow n$, $n \leftarrow r$ and came back to step 1.

Euclidean Algorithm

Example

m	n	r
24	44	24
44	24	20
24	20	4

1. [Find the reminder] Divide m by n let r be the divider. (We have $0 \leq r < n$.)
2. [Is zero?] If $r = 0$ finish the procedure; the answer is n.
3. [Simplifying] Let $m \leftarrow n$, $n \leftarrow r$ and came back to step 1.

Euclidean Algorithm

Example

m	n	r
24	44	24
44	24	20
24	20	4
20		

1. [Find the reminder] Divide m by n let r be the divider. (We have $0 \leq r < n$.)
2. [Is zero?] If $r = 0$ finish the procedure; the answer is n.
3. [Simplifying] Let $m \leftarrow n$, $n \leftarrow r$ and came back to step 1.

Euclidean Algorithm

Example

m	n	r
24	44	24
44	24	20
24	20	4
20	4	

1. [Find the reminder] Divide m by n let r be the divider. (We have $0 \leq r < n$.)
2. [Is zero?] If $r = 0$ finish the procedure; the answer is n.
3. [Simplifying] Let $m \leftarrow n$, $n \leftarrow r$ and came back to step 1.

Euclidean Algorithm

Example

m	n	r
24	44	24
44	24	20
24	20	4
20	4	0

20/4 = 5 r 0

1. [Find the reminder] Divide m by n let r be the divider. (We have $0 \leq r < n$.)
2. [Is zero?] If $r = 0$ finish the procedure; the answer is n.
3. [Simplifying] Let $m \leftarrow n$, $n \leftarrow r$ and came back to step 1.

Euclidean Algorithm

Example

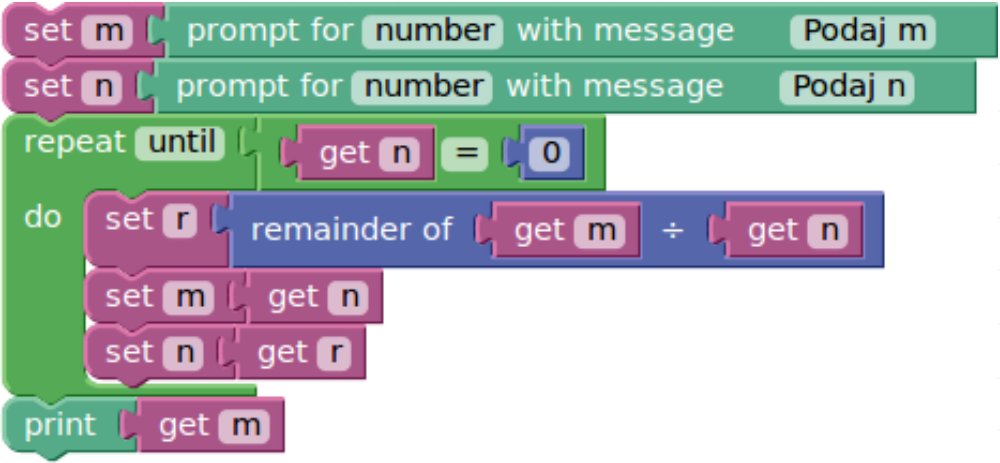
m	n	r
24	44	24
44	24	20
24	20	4
20	4	0

1. [Find the reminder] Divide m by n let r be the divider. (We have $0 \leq r < n$.)
2. [Is zero?] If $r = 0$ finish the procedure; the answer is n.
3. [Simplifying] Let $m \leftarrow n$, $n \leftarrow r$ and came back to step 1.

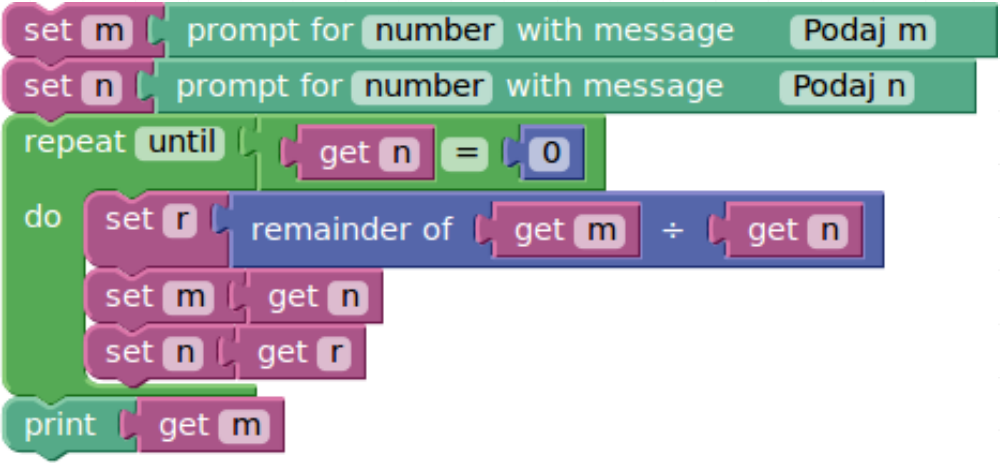
Homework

Implement Euclidean Algorithm in spreadsheet.

gcd in blockly

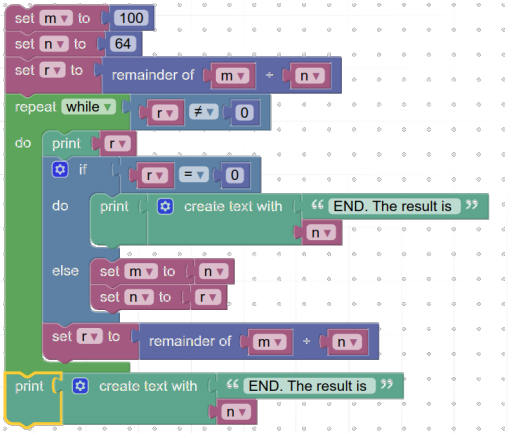


gcd in blockly



Homework: explain differences between this version of algorithm and after-mentioned algorithm.

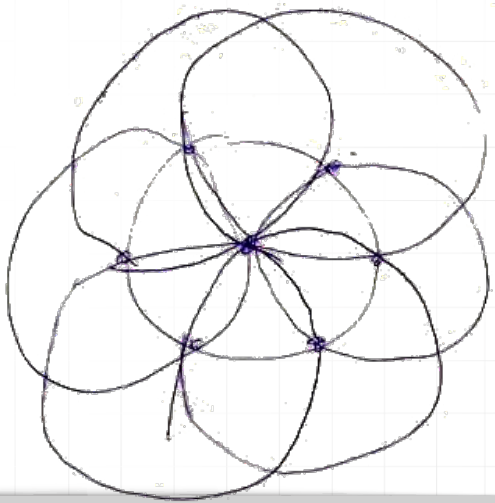
another version of gcd (ftom the lecture)



Compare and ubderstand the differences

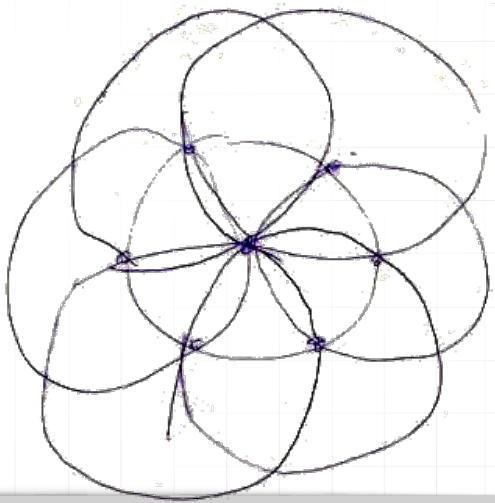
Problem

- Someone asked us for drawing a “flower”, which should look like this



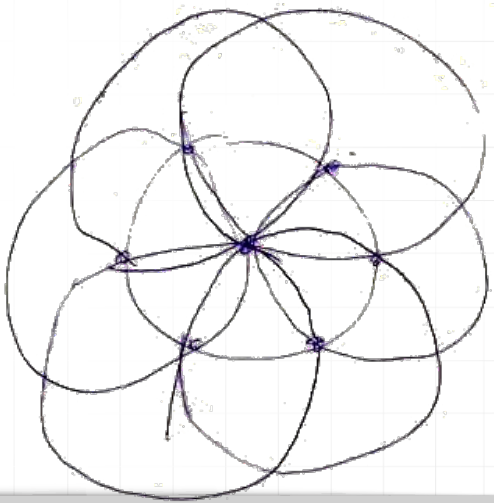
Problem

- Someone asked us for drawing a “flower”, which should look like this
- But better…



Problem

- Someone asked us for drawing a “flower”, which should look like this
- But better…
- How to make this?



Possible Methods

1. A compass. . .

Possible Methods

1. A compass…
2. A graphic program (CorelDraw, OpenOffice.org Draw, whatever…)

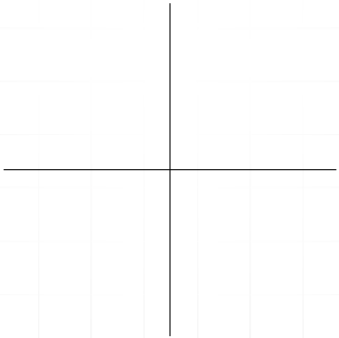
Possible Methods

1. A compass…
2. A graphic program (CorelDraw, OpenOffice.org Draw, whatever…)
3. Construct…

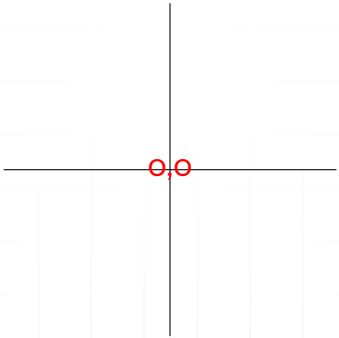
Possible Methods

1. A compass…
2. A graphic program (CorelDraw, OpenOffice.org Draw, whatever…)
3. Construct…
4. Write a computer program…

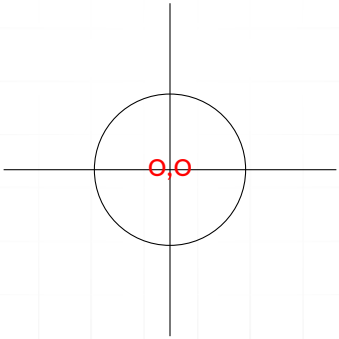
Flower



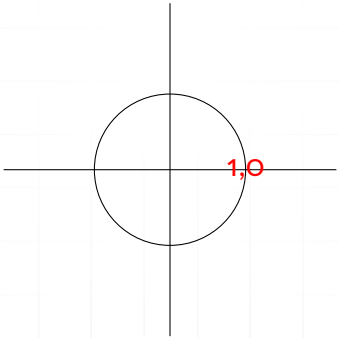
Flower



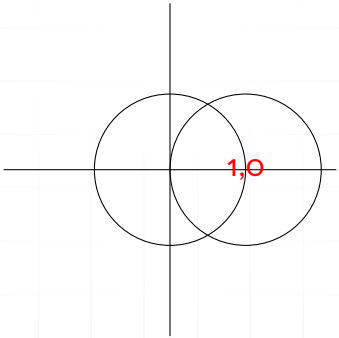
Flower



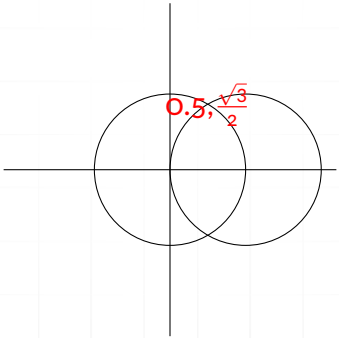
Flower



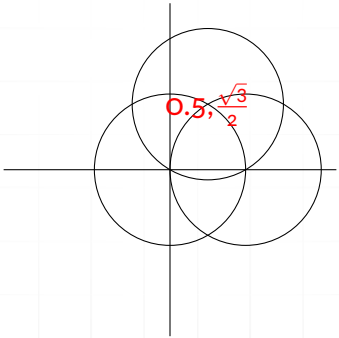
Flower



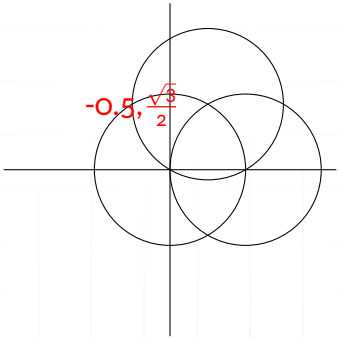
Flower



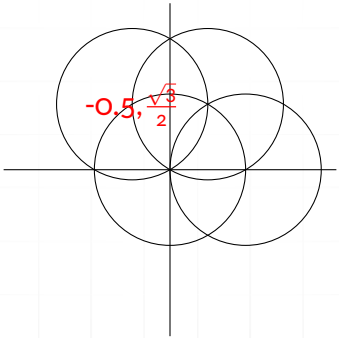
Flower



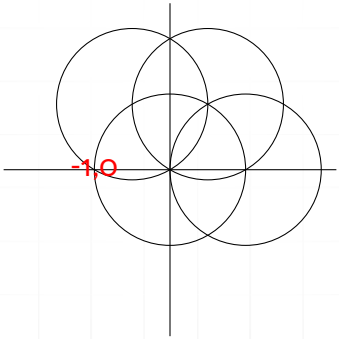
Flower



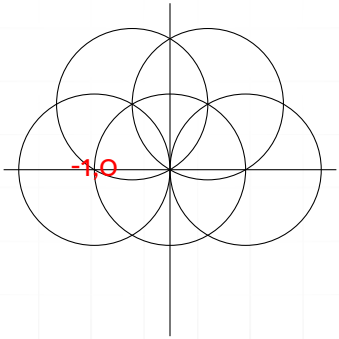
Flower



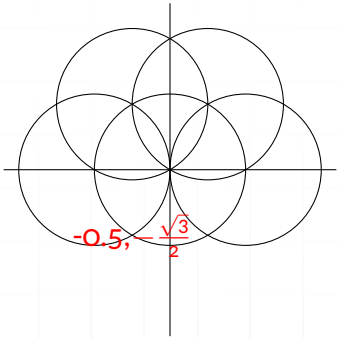
Flower



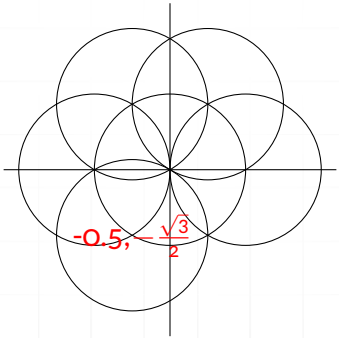
Flower



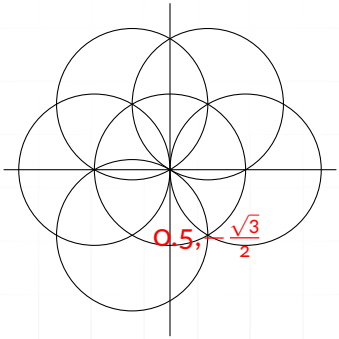
Flower



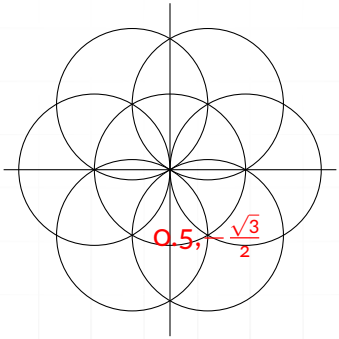
Flower



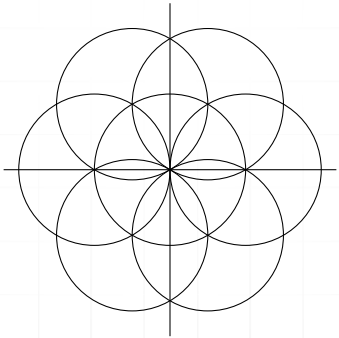
Flower



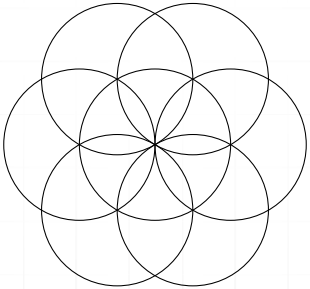
Flower



Flower



Flower



How this was made?

```
\draw (0,0) circle (1cm);
%
\draw (1,0) circle (1cm);
\draw (0.5,0.866) circle (1cm);
\draw (-0.5,0.866) circle (1cm);
\draw (-1,0) circle (1cm);
\draw (-0.5,-0.866) circle (1cm);
\draw (0.5,-0.866) circle (1cm);
```

Python implementation

```
from turtle import *
setup(600,600,300,300)
title("Kwiatek")
speed(1)
up()
goto (100,0)
down()
setheading(90)
circle(100)
for _ in range(6):
        setheading(heading() — 60)
        down()
        circle(100)
        setheading(heading() + 60)
        up()
        circle(100, 60)

exitonclick()
```

How to make this

Example 1

Input: Two numbers J and K
Output: Result of calculation $J^2 + 3K$

Simple arithmetical problem: elementary arithmetic operations on two numbers.

Example 2

Input: Positive whole number *K*
Output: Sum of all numbers from 1 to *K*

The arithmetical problem, the number of operations depends on input data.

Example 3

Input: Positive whole number *K*
Output: “YES” when *K* prime number, “NO”, when is not.

Decision problem. The arithmetical problem, but the result is not numerical.

Example 4

Input: List *L* words in a given language.
Output: List *L* ordered according to the alphabet (lexicographic order).

This is not an arithmetical problem.

Example 5

Input: Road map with cities and distances between them. There are two cities highlighted on the map, say, *A* and *B*.
Output: The shortest path from *A* to *B*.

It is a problem of searching for the best solution (shortest path among all available paths).

Example 6

Input: All people in the room.
Output: Height of the highest (lowest) person.

Example 6

Input: All people in the room.
Output: Height of the highest (lowest) person.

OR the name of the highest (lowest) person.

Example 6

Input: All people in the room.
Output: Height of the highest (lowest) person.

OR the name of the highest (lowest) person.

What is the difference between these two algorithms?

Example 7

Appendix -Troubleshooting

Problem

My 802.11b PC Card will not associate with the ASUS Wireless Router.

Solution

Follow these steps:

1. Try to bring the devices closer together; the PC Card may be out of range of the ASUS Wireless Router.
2. Confirm that the ASUS Wireless Router and PC Card have the same SSID.
3. Confirm that the ASUS Wireless Router and PC Card have the same Encryption settings, if enabled.
4. Confirm that the ASUS Wireless Router's Air and Link LEDs are solid green.
5. Confirm that the authorization table includes or excludes the MAC address of the WLAN PC card if "Wireless Access Control" is enabled.

Problem

The throughput seems slow.

Solution

To achieve maximum throughput, verify that your antennas are well-placed, not behind metal, and do not have too many obstacles between them. If you move the client closer to the ASUS Wireless Router and throughput increases, you may want to consider adding a second the ASUS Wireless Router and implementing roaming.

- Check antenna, connectors and cabling.
- Verify network traffic does not exceed 37% of bandwidth.
- Check to see that the wired network does not exceed 10 broadcast messages per second.
- Verify wired network topology and configuration.

Example 9

The battery pack runs down too quickly.

- Charge it sufficiently (→ *step 1 in "Read This First"*).
- You are using the camera in an extremely cold location (page 99).
- The battery terminal is dirty. Clean the battery terminal with a cotton swab, etc., and charge the battery pack.
- The battery pack is dead (page 99). Replace it with a new one.

Cannot turn on the camera.

- Install the battery pack correctly (→ *step 1 in "Read This First"*).
- The battery pack is discharged. Install charged battery pack (→ *step 1 in "Read This First"*).
- The battery pack is dead (page 99). Replace it with a new one.

Example 10

This troubleshooting guide provides solutions to some common problems that you may encounter while installing and/or using ASUS Pocket Wireless AP. These problems require simple troubleshooting that you can perform by yourself. Contact the ASUS Technical Support if you encounter problems not mentioned in this section.

Problem	Action
The ASUS Pocket Wireless AP does not power up.	• Use a test meter to measure the voltage output of the power source through the power plug. • Check if the power plug is properly connected to the device.
Other devices cannot communicate with the ASUS Pocket Wireless AP through a wired network connection.	• Verify your network configuration to ensure that there is no IP address duplication. Turn off the device in question, then ping the assigned IP address of the device. Make sure no other device responds to that address. • Check if the cables have the proper pin outs and connectors. You may also use another LAN cable. • Make sure the hub, switch, or computer connected to the ASUS Pocket Wireless AP supports 10Mbps or 100Mbps speed. Do this by check the ASUS Pocket Wireless AP and the Hub LEDs. When you connect the ASUS Pocket Wireless AP to a 10/100 Mbps hub, both the Hub LED and the ASUS Pocket Wireless AP Ethernet LEDs should light up.

Summary I

To summarize, an algorithmic problem consists of:

1. the characterization of a legal, possibly infinite, collection of potential input sets,
   and
2. a specification of the desired outputs as a function of the inputs.

It is assumed that either a description of the allowed basic actions or a hardware configuration together with its built-in basic actions are also provided in advance. A solution to an algorithmic problem consists of an algorithm, composed of elementary instructions prescribing actions from the agreed-on set. This algorithm, when executed for any legal input set, solves the problem, producing the output as required.

Bibliography


David Harel and Yishai Feldman.
Algorithmics: The Spirit of Computing.
Addison-Wesley, 3 edition, June 2004.


Patricia Sinclair and Ruth Malinowski.
French cooking.
Weathervane Books, [New York], 1978.

Colophon

Presentation typeset using system LaTeX 2_ε and beamer class using Carlito font by Łukasz Dziedzic.
Title illustration presents an excerpt from an illustrated encyclopedia written by German humanist writer Gregor Reisch entitled Madame Arithmetica.