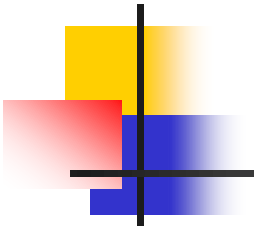
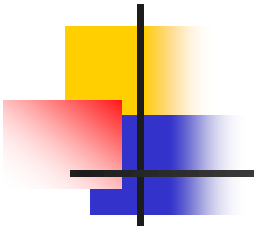


Introduction to Computer Systems

Department of Computer Science and Information Systems

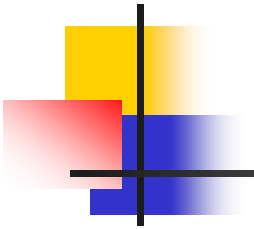
Lecturer: Steve Maybank
sjmaybank@dcs.bbk.ac.uk
Spring 2020

Week 8a: Pseudo Code and Algorithms



Algorithms

- Definition: an algorithm is an ordered set of unambiguous, executable steps that defines a terminating process.
- The steps are often called instructions.
- Termination is by a special instruction: halt.

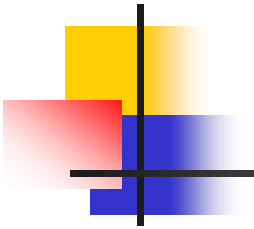


Device

- An algorithm requires a device which carries out the instructions

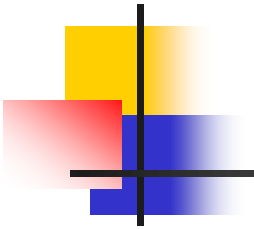
- CPU: machine code
- CPU+compiler: high level language
- Mathematica:

$$\int (x-1)/(x+1)dx$$



Terminology

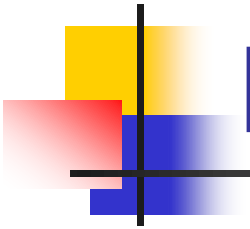
- Programming language: system for representing algorithms in a human readable form suitable for conversion to machine code
- Program: representation of an algorithm in a programming language
- Pseudo code: writing system for the informal representation of algorithms
- Hardware: physical device to carry out the steps or instructions in a program.



Representation of an Algorithm

The same algorithm can be represented in many different ways:

- F = (9/5)C+32
- F <- (9*C)/5+32
- To obtain F multiply C by (9/5) and then add 32.
- Lookup table for Fahrenheit and Centigrade

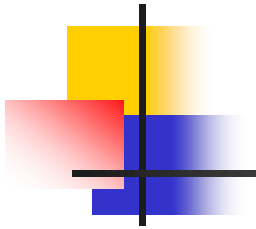


Program 1 for XOR

input: binary digits a,b

output: a XOR b

- if a==0 and b==0 return 0
- if a==0 and b==1 return 1
- if a==1 and b==0 return 1
- if a==1 and b==1 return 0
- halt

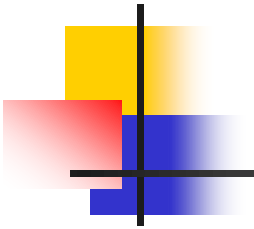


Program 2 for XOR

input: binary digits a, b

output: a XOR b

1. if a==b return 0
2. else return 1
3. halt

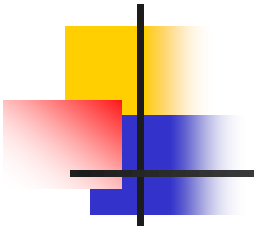


Program 3 for XOR

input: binary digits a,b

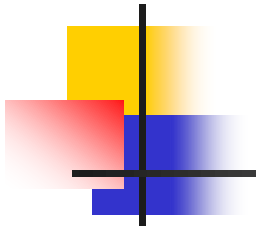
output: a XOR b

1. convert a to the integer ia
2. convert b to the integer ib
3. ic=remainder on dividing ia+ib by 2
4. return ic
5. halt



Pseudo Code for Algorithm Representation

- Very useful informal representation of algorithms
- Easier than using program code when designing an algorithm
- Basic structures (assignment, loops, arrays …) are similar across many programming languages.



Assignment

- General form

  name = expression

- Examples

  q = 3

  funds = balance + savings

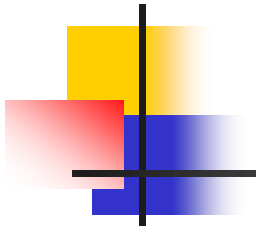
Conditional Selection

- General form

```
if (condition)
        activity1
else
        activity2
endIf
```

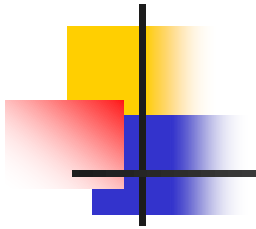
Example

```
if (year is leap year)
    dailyTotal = total/366
else
    dailyTotal = total/365
endIf
```



Nested if Statement

```
if (not raining)
        if (temperature == hot)
                go swimming
        else
                play golf
        endIf
else
        watch television
endIf
```



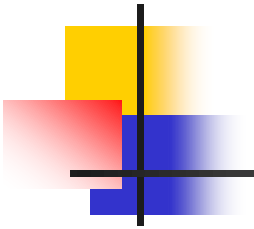
Repeated Execution

- General form

```
while (condition)
      activity
endWhile
```

Example

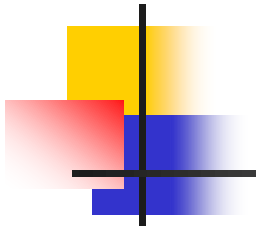
```
while (tickets remain)
    sell a ticket
endWhile
```



Function

- Definition of a function: set of instructions which can be used as a single unit in different contexts.
- Example

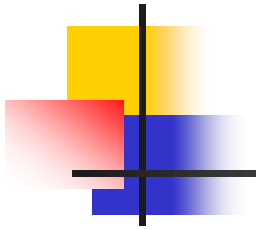
```
function greetings()
        count = 3
        while (count > 0)
                print("Hello")
                count = count-1
        endWhile
endFunction
```



Parts of a Function

```
function greetings()
    count = 3
    while (count > 0)
        print("Hello")
        count = count-1
    endWhile
endFunction
```

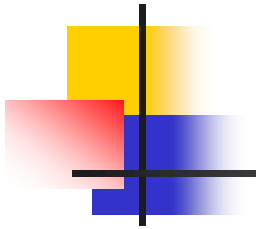
```
# keywords: function, endFunction
# name: greetings
# parameters: none
# header: function greetings()
# body: code apart from the header
#     and endFunction
# return value: none (print does not
#     return a value to the calling
#     program)
```



Example of a Function

```
function temp(c)
    f = (9/5)*c+32
    return f
endFunction
```

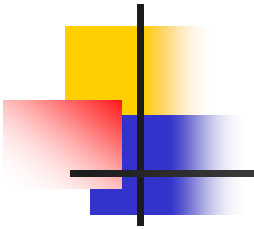
```
# keywords: function, endFunction
# name: temp
# parameter: c
# header: function temp(c)
# body: 'f=(9/5)*c+32' and 'return f'
# return value: value of f
```



Function Call

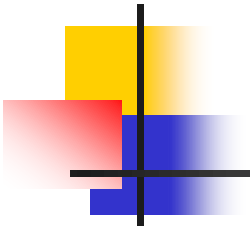
```
c = 0
while (c <= 100)
    f = temp(c)
    c = c+1
    print(f)
endWhile
```

```
# The variable f in the while loop
# is different from the variable f
# in the definition of the function
# temp, even though the two
# variables have the same name
```



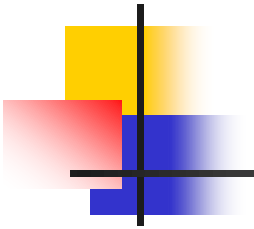
Which Numbers are Printed?

```
last = 0
current = 1
while (current < 100)
        print(current)
        temp = last
        last = current
        current = last+temp
endWhile
```



Exercise 1

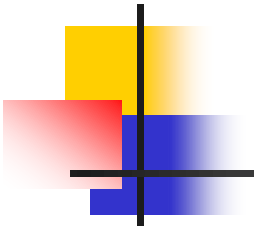
- The Euclidean algorithm finds the greatest common divisor of two strictly positive integers assigned to the variables x and y.
- As long as x and y are both not zero divide the larger number by the smaller. Replace the larger value with the remainder. When x or y has the value 0 the value of the other variable is the GCD
- Write a pseudocode version of the Euclidean algorithm



Example 2

- A year n is a leap year if n is divisible by 400 or if n is divisible by 4 but not by 100. In all other cases n is not a leap year

- Write a pseudo code version of a function leapYear that takes an integer n as a parameter and returns True if n is a leap year and False otherwise.

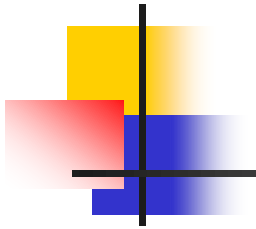


Exercise 3

- Write an algorithm in pseudo code to carry out the following task:

  input: a 1-dimensional array A of integers

  output: the integer -1 if $A[i] \geq A[i+1]$ for
  $0 \leq i \leq \text{Length}[A]-2$,
  otherwise the least integer i such that $A[i] < A[i+1]$.



Exercise 4

- Design an algorithm for finding all the factors of a positive integer. For example, in the case of the integer 12, your algorithm should report the values 1, 2, 3, 4, 6 and 12