CS 101
Computer Programming and Utilization

Instructor: Rohit Gurjar

All material based on Prof. Abhiram Ranade's book and slides
(and previous course instructors)

Language Syntax

- Syntax = grammatical rules indicating how commands must be written.
- Syntax of programming languages is very strict, e.g.
  - “`right(90);`” cannot be written as “`right 90;`”.
  - “`penUp()`” cannot be written as “`penup()`” or “`penUp`”, i.e. without parentheses.
  - We will later learn other kinds of statements which will have their own syntax which must be adhered to.
- Lot of flexibility is still allowed, e.g.
  - Wherever a number is acceptable, often an ”expression” such as `360/n` is acceptable
  - repeat statement is allowed wherever other statements are allowed, e.g. we can have a repeat inside another repeat.

Comments

- A program will be executed on a computer, but it will also be read by people.
- Sometimes readers may not understand why the program is written the way it is written.
- To help such human readers, you can place “comments” in your program.
  - Anything placed between /* and */ is a comment
  - Anything between // and end of line is a comment
  - A comment is meant only for human readers and is ignored by the computer during execution.

A program with Comments

```
/* Author: Abhiram Ranade
Program to draw polygon */
#include <simplecpp>
main_program{
  turtleSim();
  cout << “How many sides?”;
  int nsides; cin >> nsides;
  repeat(nsides){
    forward(100);
    right(360.0/nsides);  // Exterior angle of an n sided polygon is 360/n
  }
  wait(10);
}
```

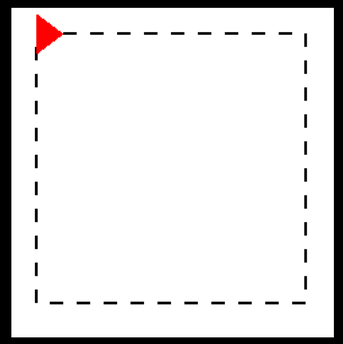
Indentation

```
#include <simplecpp>
main_program{
    turtleSim();
    cout << “How many sides?”;
    int nsides;
    cin >> nSides;
    repeat(nSides){
        forward(100);
        right(360.0/nSides);
    }
    wait(10);
}
```

- You will notice that at the beginning of some lines some space is inserted.
- This space is called indentation.
- The indentation allows you to quickly see which statements constitute the body of the repeat, which statements are part of the main program.
- The general rule: if X is ”inside” Y, then put four additional spaces before every line of X.
- Also note how the { and } are placed.
- Indentation is very helpful for human readers.
- Indentation is ignored during execution.

Repeat within repeat (nested repeat)

```
repeat(4){
    repeat(10){
        forward(10); penUp();
        forward(10); penDown();
    }
    right(90);
}
```



The above program makes this dashed square.

How nested repeat executes

```
repeat(2){
      xxx
     repeat(3){
          yyy
     }
     zzz
}
```

The code on left will execute the statements in the following manner.

```
xxx
yyy
yyy
yyy
zzz
xxx
yyy
yyy
yyy
zzz
```

How nested repeat executes

What will the following program print?

```
#include <simplecpp>
main_program{
  cout << “a”;
  repeat(5){
    cout << “b”;
    repeat(2){  cout << “c”; }
    cout << “d”;
  }
}
```

Commonly used terminology

- “Control is at statement w”: Computer is currently executing statement w.
- “Control flow”: The order in which statements get executed.
  - Execution starts at top and goes down.
  - Retraced if there is a repeat statement.
- Variable: region of memory designated for storing some value you need
- Example: `nsides` which we saw earlier.
- Named so because the value stored in the region can vary
- How to change the value: later.

Why picture drawing

- The calculations/actions needed to solve any problem will contain patterns
  - For example, a sequence of calculations may be repeated
- Key programming activity: The pattern in the calculations must be mirrored by appropriate statements in program.
  - If some calculation is to be repeated 5 times, use repeat(5){} rather than writing out the statement 5 times
- Interesting pictures also contain patterns.
- To draw interesting pictures you need to use repeat statements competently
- By drawing interesting pictures you get some practice at identifying patterns and implementing them in your programs.
- Similar ideas may be used in programming a drone, 3D printer, animation.

Spirit of the course

- Learn C++ statements / concepts.
- Understand patterns in the calculations that you want to do
  - Very important in all programming, not just drawing.
- Goal: if you can solve a problem by hand, possibly taking an enormous amount of time, by the end of the course, you should be able to write a program for it.
- Learn new ways of solving problems!
- Do not be afraid of using the computer.
- “What if I write xyz in my program instead of pqr?” : Just do so and find out.
  - Be adventurous.
- Exercise your knowledge by writing programs – that is the real test.

Data representation and computation

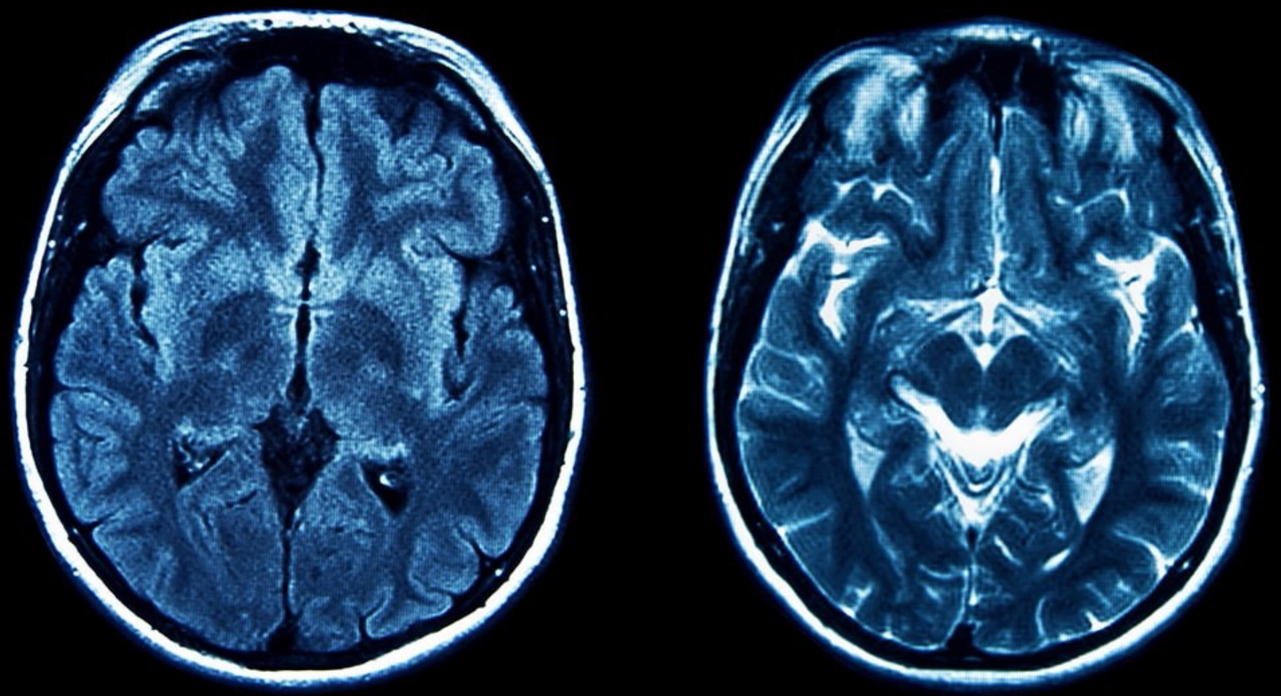
A basic question

- How is a computer able to do so many things?
  - Search for information
  - Process pictures, identify features, edit them
  - Create animation
  - Play chess
- Most real life problems can be viewed as mathematical problems on numbers
- A computer is good at solving math problems

What is in this picture?

https://en.wikipedia.org/wiki/File:Jackson%27s_Chameleon_2_edit1.jpg

MRI of a brain: does it have a tumor?

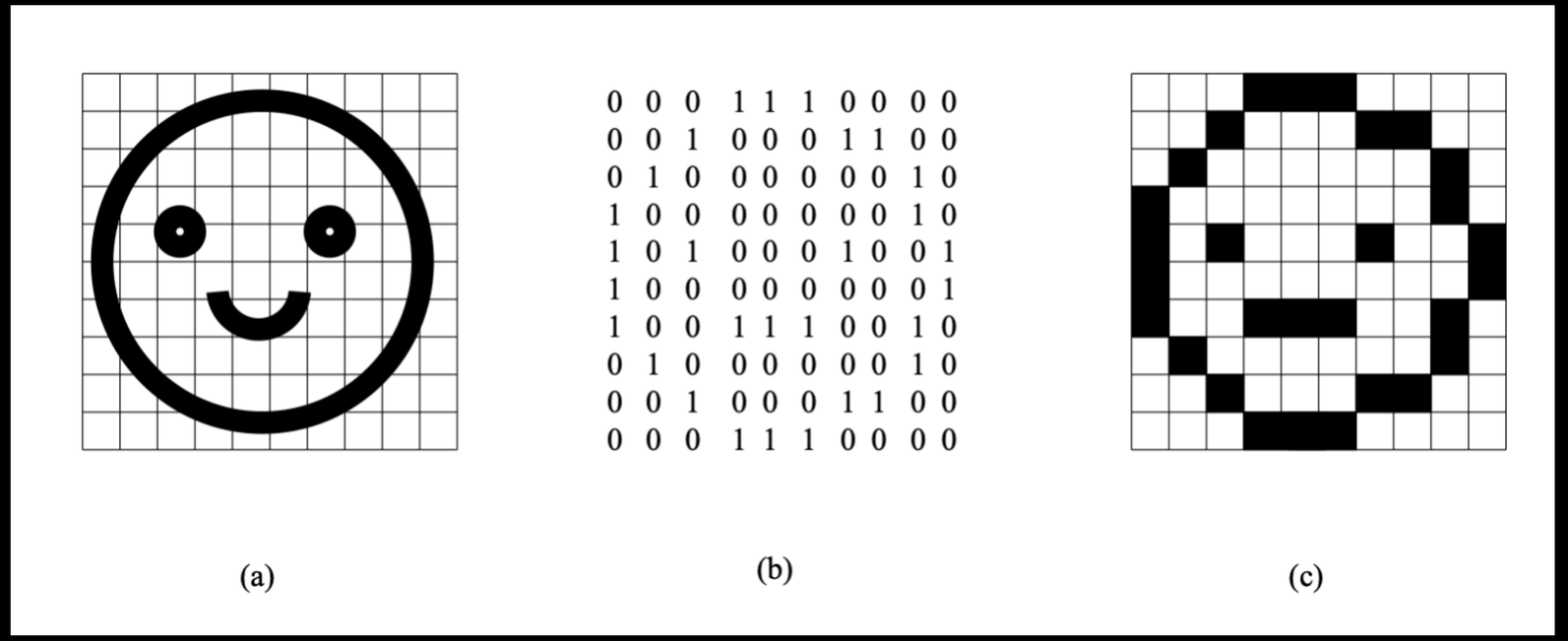


https://sjra.com/can-you-see-a-brain-tumor-on-an-mri-scan/

Representing a black & white picture using numbers

- Suppose picture is 10cm x 10cm.
- Break it up into 0.1 mm x 0.1 mm squares
- 1000 x 1000 squares. 1 million “pixel”s
- If square is mostly white, represent it as 0.
- If square is mostly black, represent it as 1.
- Picture = 1 million numbers!

Picture, representation, reconstruction



Better representation

- Better representation if picture divided into more cells.
- Pictures with different “gray levels”:
  - use numbers 0, 0.1, …, 1.0 to represent level of darkness rather than just 0, 1.
- Pictures with colours: picture = 3 sequences
  - sequence for red component,
  - sequence for blue component,
  - sequence for green component
- Add up the colours to get the actual colour.

An image processing problem

- Input: sequence P of 1 million numbers (0 or 1)
  - Representing a 10cm x 10cm black and white picture,
  - Given in left to right, top to bottom order.
- A very simple image processing problem:
  - Is there a vertical line in the picture?
- Expressing the problem mathematically: What property does the sequence need to have if it is to contain a vertical line?
  - All 0s, except for 1s in consecutive rows of some column
  - Going down a column = move 1000 positions in the sequence
- 1s in positions i, i+1000, i+2000, i+3000, i+4000,… for some i
- ”Is there a vertical line?” = “Does sequence P satisfy above property?”
- One way of solving the problem:
- Try all values of i..

Does the MRI scan contain a tumour?

- In principle, same as asking whether the picture contains a single vertical line:
- Identify a set of properties that the sequence of numbers representing the picture must satisfy if the picture contains a tumor.
- Identify computations that can check if the given number sequence satisfies the required properties.
- In practice requires enormous ingenuity
- Main concern of the deep subject “Computer Vision/deep learning”

Language/text using numbers

- Define a code for representing letters.
- Commonly used code: ASCII
  (American Standard Code for Information Interchange)
- Unicode for characters of other languages.
- Letter ‘a’ = 97 in ASCII, ‘b’ = 98, …
- Uppercase letters, symbols, digits also have codes. Code also for space character.
- Words = sequences of ASCII codes of letters in the word.
- ‘computer’ = 99, 111,109,112,117,116,101,114.
- Does the word “computer” occur in a paragraph?
- Does a certain sequence of numbers occur inside another sequence of numbers?

Exercise

- Suppose you are given a sentence in a language you cannot understand. Would you still be able to count the number of words in the sentence? Can you express this as a question on sequences of numbers representing the ASCII codes of different characters?
- How will you represent Chess playing as a question on numbers? Start by representing a chess board with pieces on it using numbers.

How numbers are written in a computer?

- A computer has a large set of capacitors
  - 8 GB RAM means $2^{36} \approx 6 \times 10^{10}$ capacitors
  - Each capacitor represents a bit (0 or 1)
    - Low charge represents 0
    - High charge represents 1
- Once you represent 0 and 1, you can represent anything.
- Binary number system
- Electrical circuits are designed to perform basic operations like addition, multiplication, copying.

Binary number system

- A sequence $a_n\, a_{n-1} \ldots \;\; a_1\, a_0\, a_{-1} \ldots a_{-k}$ of 0s and 1s
- Will represent the number
  $a_n\, 2^n + a_{n-1}\, 2^{n-1} + \ldots + a_1\, 2^1 + a_0\, 2^0 + a_{-1}\, 2^{-1} + \ldots + a_{-k}\, 2^{-k}$
- Converting decimal integer v to binary
  - Divide v by 2, remainder gives a_0
  - Repeat previous step with the quotient to get a_1, a_2, …
- Converting fraction f to binary
  - If $f > 0.5$, $a_{-1} = 1$ and 0 otherwise.
  - Similarly other bits…

Positive/negative numbers

- Typically number of capacitors used for an integer
  - 8, 16, 32, 64
- One of the bits (leftmost) indicates sign
- -25 will be represented as
  - 10000000000000000000000011001
- Numbers stored from -2^{31} from $2^{31}-1$
- Two's complement