C Reload

Simple Computer

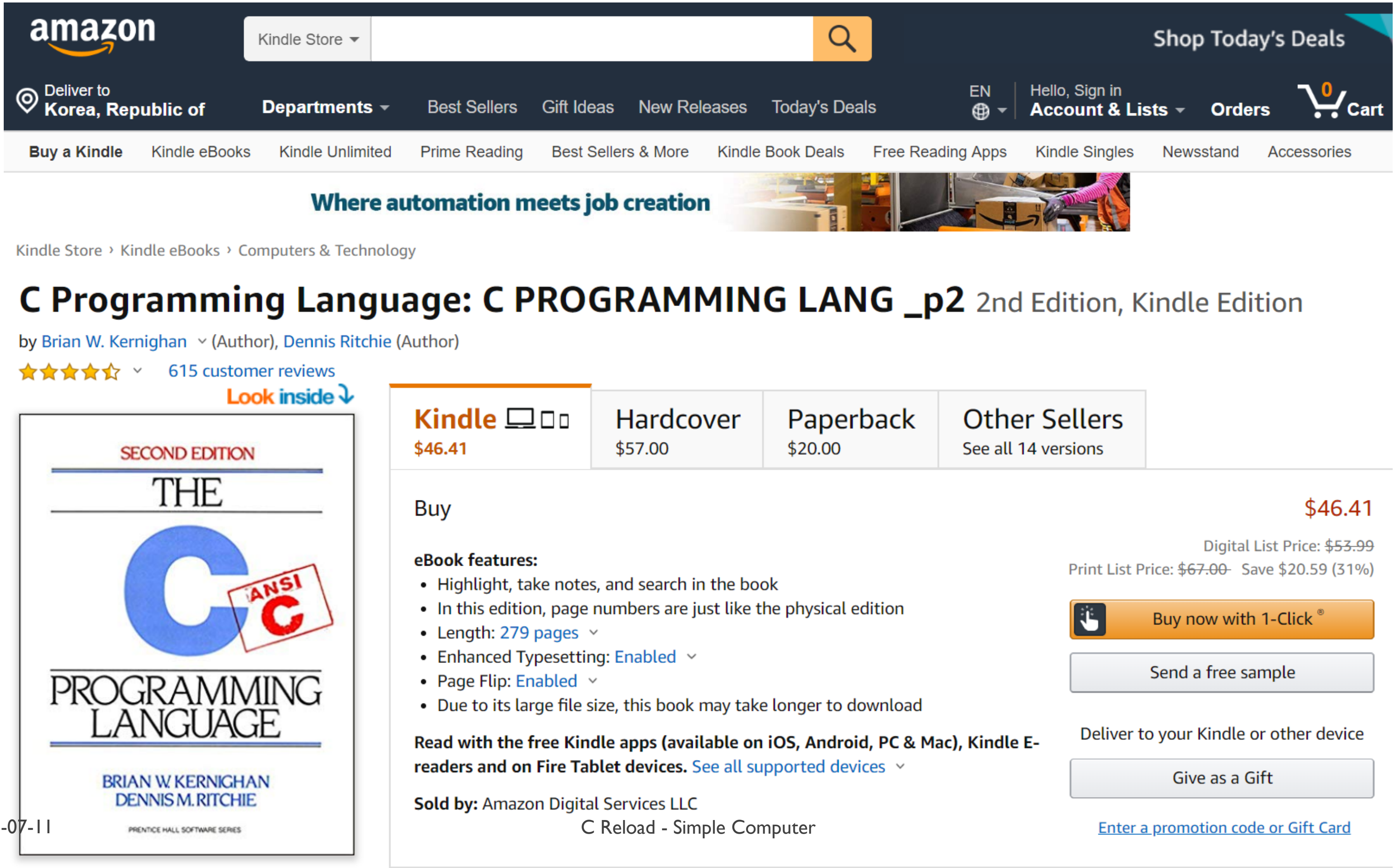
Shin Hong

Programming Environment

- Linux
  - http://peace.handong.edu:8000/register/
- Mac
  - Download and install Xcode from the Mac App store
- Windows
  - MinGW http://www.codebind.com/cprogramming/install-mingw-windows-10-gcc/
  - Download and install Ubuntu 16.04 LTS from the MS app store

Textbook

- The C Programming Language, 2/e, by Brian Kernighan and Dennis Ritchie



C Programming Language: C PROGRAMMING LANG _p2 2nd Edition, Kindle Edition

by Brian W. Kernighan (Author), Dennis Ritchie (Author)

615 customer reviews

Kindle	Hardcover	Paperback	Other Sellers
$46.41	$57.00	$20.00	See all 14 versions

Buy $46.41

Digital List Price: ~~$53.99~~

Print List Price: ~~$67.00~~ Save $20.59 (31%)

eBook features:

- Highlight, take notes, and search in the book
- In this edition, page numbers are just like the physical edition
- Length: 279 pages
- Enhanced Typesetting: Enabled
- Page Flip: Enabled
- Due to its large file size, this book may take longer to download

Read with the free Kindle apps (available on iOS, Android, PC & Mac), Kindle E-readers and on Fire Tablet devices. See all supported devices

Sold by: Amazon Digital Services LLC

Buy now with 1-Click ®
Send a free sample
Deliver to your Kindle or other device
Give as a Gift
Enter a promotion code or Gift Card

What is computing?

- What is programming?

George Boole

George Boole (1815--1864)

Formulate a calculus of reasoning

- Claim that logic should be considered as a branch of math, rather than a part of philosophy
- Argue that there are math laws to express the operation of human mind
- Showed that Aristotle's syllogistic logic could be rendered as algebraic equitation

A Brief History of Computing by G. O’Regan

Charles Babbage

orm

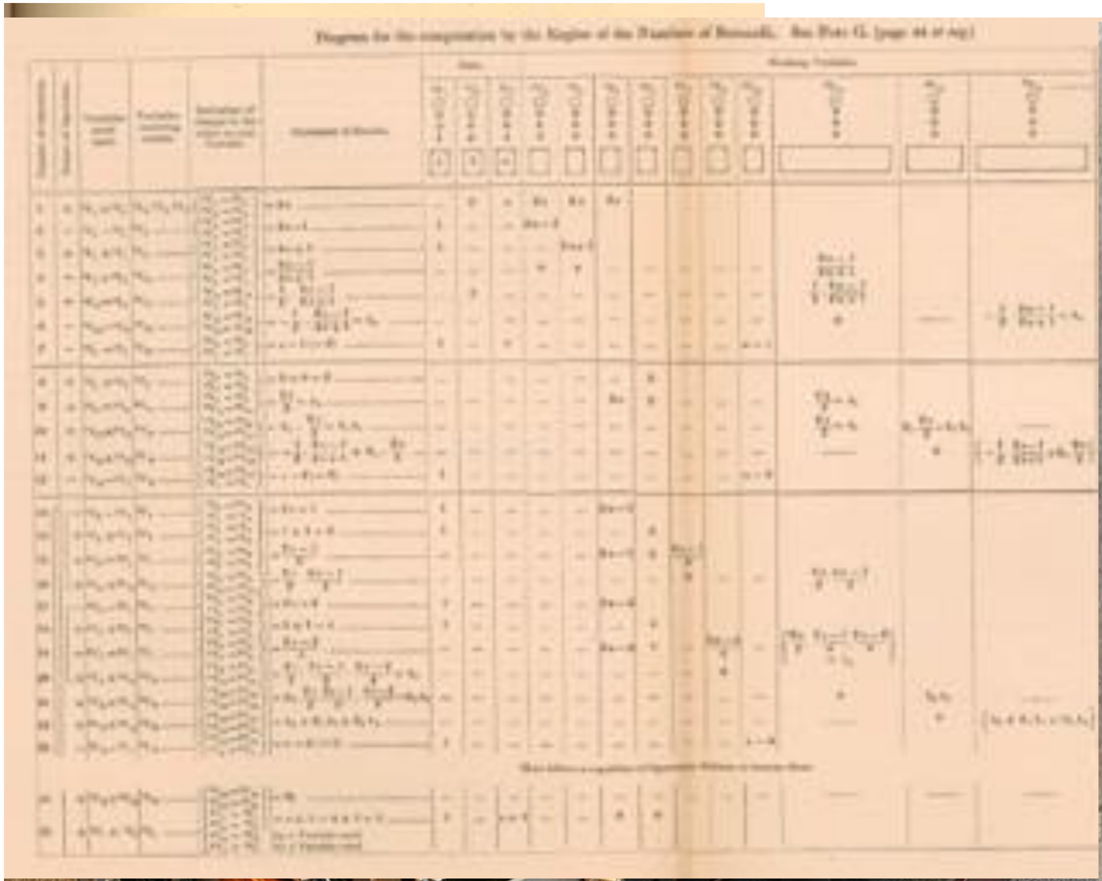
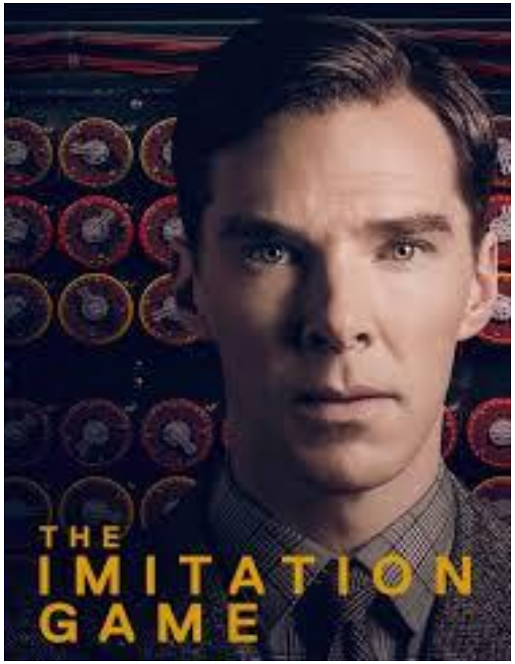
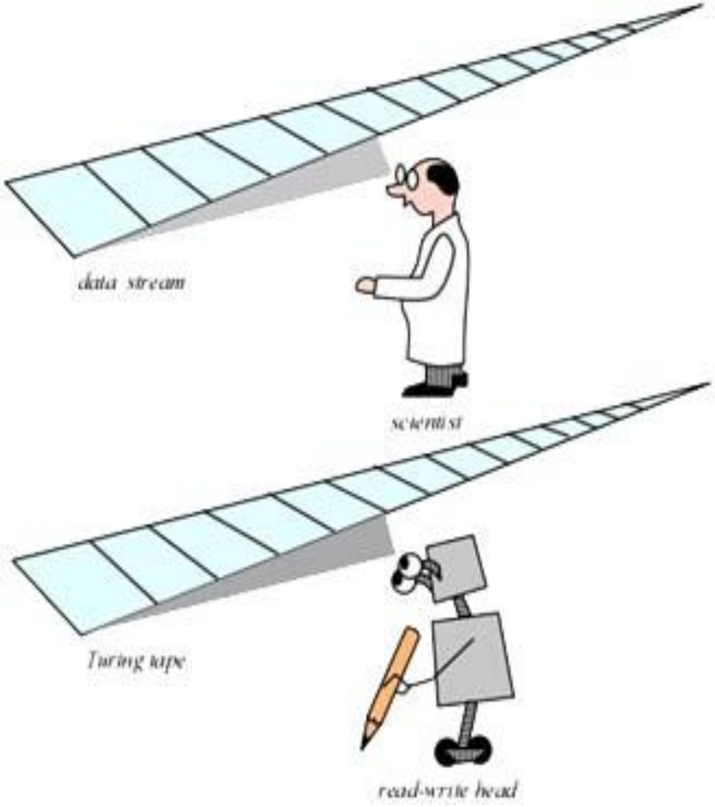


Diagram for the computation by the Engine of the Numbers of Bernoulli. See Note G. (page 44 et seq.)



THE
IMITATION
GAME



data stream
scientist
Turing tape
read-write head

Computer Model

- Memory is a map from addresses to values
- A memory address is a non-negative integer
- A value is either a number or instruction
    - number: 0 to 255
    - instruction
        - receive an input
        - produce an output
        - evaluate an expression over memory addresses
        - assign a value to a memory address
        - jump to a memory address
        - finish
- A processor loads and executes instructions from address 0

Simple Computer Example

- Memory addresses: 0, 1, ..., 29
- Instruction
  - READ [m] Receive a new input and write it on address m
  - WRITE [m] Print out the number at address m
  - ASSIGN [m] [c] Put a number c to address m
  - MOVE [md] [ms] Put the value at address ms to address md
  - ADD [md] [mx] [my] Put the add of values in memory addresses mx and my to md
  - MINUS [md] [mx] [my] Put the minus of values in memory addresses mx and my to md
  - MOD [md] [mx] [my] Put the modulo of values in memory addresses mx and my to md
  - EQ [md] [mx] [my] Put 1 to md if the value at mx is equivalent to the value at my. Otherwise, put 0 to md.
  - LESS [md] [mx] [my] Put 1 to md if the value at mx is less than the value at my. Otherwise, put 0 to md.
  - JUMP [m] [c] Jump to address m if the value at c is not zero
  - TERM Finish the program execution

Exercise 1. Prime Number

- Write a program for the simple computer model that checks whether or not a given positive number is prime
  - Construct a map/table of the 30 memory addresses
  - Ex. check whether a given number is even

```
0: READ 21
1: EQ 23 21 22
2: JUMP 23 10
3: JUMP 24 7
4: ASSIGN 24 20
5: MINUS 21 21 20
6: JUMP 1
7: ASSIGN 24 22
8: MINUS 21 21 20
9: JUMP 1 20
10: PRINT 24
11: TERM
```

```
...
20: “1”
21:
22: “0”
23:
24: “1”
```