Introduction to Computing

Breaking the Ice

Malay Bhattacharyya

Associate Professor

MIU, CAIML, TIH
Indian Statistical Institute, Kolkata

August, 2025

Let's have some fun!!!

Suppose that you are playing a game with the computer, both first choosing either of the labels EVEN and ODD. You then show a number with your fingers in both hands and the computer promptly show a number (any arbitrary nonnegative integer) without seeing yours. If no finger is shown by you, it is treated as a 0. Depending on whether the sum of the two numbers is EVEN or ODD, the winner will be chosen. Write a program to ensure that the computer has a maximum chance of winning (if possible).

Let's have some fun!!!

Suppose that you are playing a game with the computer, both first choosing either of the labels EVEN and ODD. You then show a number with your fingers in both hands and the computer promptly show a number (any arbitrary nonnegative integer) without seeing yours. If no finger is shown by you, it is treated as a 0. Depending on whether the sum of the two numbers is EVEN or ODD, the winner will be chosen. Write a program to ensure that the computer has a maximum chance of winning (if possible).

Example:
Let us assume you chose the label EVEN and the computer ODD. Then if you show one finger () and the computer shows a seven ([7]), the winner is you because $1 + 7 = 8$ is EVEN.

What we learnt?

Solving a problem on a computer requires:

1. Understanding the logical actions (strategies) to take.
2. Choosing the best strategy.
3. Finding out the best implementation of the chosen strategy.

The SALADS approach

"Think, Think, Practice Thinking."

– Ritwik Ghatak, Indian Filmmaker

S	Shape (the problem into a version expressible by a computer)
A	Ask (questions)
L	Logify (the solution of the problem)
A	Algorithmically (formulate the working principle)
D	Data Structures (are to be chosen appropriately)
S	Scrutinize (the program for testing and verification)

Let's see how ChatGPT 5.0 responds!!!

Alright – let's break down the game step-by-step so we can design a program that maximizes the computer's winning chances.

Game Understanding

- Both you and the computer choose EVEN or ODD beforehand.
- You show a number with 0–10 fingers (both hands).
- The computer then shows a number (any nonnegative integer, without seeing yours).
- The sum of the two numbers determines whether the result is EVEN or ODD.
- Whoever's chosen label matches the sum wins.

The tricky part is:

The computer can't see your number before it shows its own number. That means the computer cannot adapt per round to your specific number — only choose a strategy that maximizes the probability of winning in the long run.

Let's see how ChatGPT 5.0 responds!!!

Key Observation

- If your number is even, then the sum's parity = parity of the computer's number.
- If your number is odd, then the sum's parity = opposite parity of the computer's number.
- Since the computer can't see your number, the only way to get more than 50% win chance is if it knows your parity bias (whether you tend to show even or odd numbers more often).

Strategy

- If you are random and pick even/odd equally:
  $\rightarrow$ The game is perfectly fair, max win probability = 50
- If you have a bias (say you choose even numbers 70% of the time):
  $\rightarrow$ The computer should choose its number so that the resulting sum matches its own label for the most frequent case.
- So, before the real game, the computer should learn your habits (record your previous numbers and check their parity frequency).

Let's see how ChatGPT 5.0 responds!!!

Program Design

1. Track the frequency of your shown parities (EVEN vs ODD).
2. Choose label: Pick the one that matches the parity of the sum for your most common case.
3. Pick number:
   If computer's label is EVEN:
   - If you often show even $\rightarrow$ pick even number.
   - If you often show odd $\rightarrow$ pick odd number.

   If computer's label is ODD:
   - If you often show even $\rightarrow$ pick odd number.
   - If you often show odd $\rightarrow$ pick even number.

Let's see how ChatGPT 5.0 responds!!!

ChatGPT 5.0 finds it hard to solve the following types of programming problems:

- highly adversarial or trick-based
- ambiguously worded
- with hidden or unusual constraints
- large, stateful, or having multiple parts
- covering multiple unrelated domains
- requiring implicit real-world knowledge
- related to human psychology.

Topics to be covered

- Basics
- Computer Fundamentals and Hardware
- Mathematics for Computing
- Programming in Python
- Algorithms
- Data Structures

Course Webpage

https://www.isical.ac.in/~malaybhattacharyya/Courses/In2Comp/Fall2025

Resources

Books

1. D. E. Knuth, The Art of Computer Programming, Volumes 1-4, Pearson Education.
2. R. G. Dromey, How to Solve it by Computer, Pearson Education.
3. Mark Lutz, Learning Python, O'Reilly.
4. Michael T. Goodrich, Roberto Tamassia and Michael H. Goldwasser, Data Structures and Algorithms in Python, Wiley.
5. Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, Data Structures and Algorithms, Pearson.
6. E. Horowitz and S. Sahni, Fundamentals of Data Structures, Universities Press.
7. T. A. Standish, Data Structure Techniques, Addison Wesley.

Competitive programming

ACM ICPC Past Problems
`https://icpc.global/worldfinals/past-problems`

Evaluation

- MID-SEMESTER - 30
- SEMESTER - 50
- ASSIGNMENT - 10 (Scribe + Programming Test)
- PROJECT - 10
- BONUS (For submitting weekly assignments)

Hometasks

- Open an account on GitHub – You need to keep your codes in a repository
- Open an account on Overleaf – You need to scribe for this course
- Open an account on Google Colab – You need an interactive web-based environment for creating and sharing computational staffs
- Open an account on Kaggle – You need an interactive web-based environment for getting access to benchmark datasets and partaking into data science competitions