Algorithms

An algorithm is a set of instructions that a computer follows to solve a problem. It is a step-by-step procedure that describes how to perform a computation. Algorithms are used in many areas of computer science, such as artificial intelligence, machine learning, and data science. They are also used in other fields, such as mathematics, engineering, and operations research.

The word "algorithm" comes from the name of the Persian mathematician Al-Khwarizmi, who lived in the 9th century. Al-Khwarizmi wrote a book called "Kitab al-Jabr wa-l-Muqabala" (The Book of Restoration and Reduction), which introduced the concept of algebra and described how to solve linear and quadratic equations using a step-by-step process.

Algorithms can be simple or complex, depending on the problem they are designed to solve. Some algorithms are easy to understand and implement, while others are more difficult. When there are multiple algorithms for a particular problem (and there often are!), the best algorithm is typically the one that solves it the fastest.

An algorithm is made up of three basic building blocks: **sequencing**, **selection**, and **iteration**.

- **Sequencing**: An algorithm is a step-by-step process, and the order of those steps are crucial to ensuring the correctness of an algorithm.
- **Selection**: Algorithms can use selection to determine a different set of steps to execute based on a Boolean expression.
- **Iteration**: Algorithms often use repetition to execute steps a certain number of times or until a certain condition is met.

By combining sequencing, selection, and iteration, we can come up with an algorithm that is correct and efficient for a given problem.

A good algorithm must have the following characteristics:

- **Correctness**: The algorithm should produce the expected output for all possible inputs.
- **Finiteness**: The algorithm must have a well-defined termination condition. This means that it eventually reaches an endpoint or halts after a finite number of steps.
- **Input and output**: The algorithm should take inputs, which are the initial data or information provided to the algorithm, and produce outputs, which are the results or solutions generated by the algorithm after processing the inputs. The relation between the inputs and outputs is determined by the algorithm's logic.
- **Determinism**: The algorithm should be deterministic, meaning that given the same inputs and executed under the same conditions, it will always produce the same outputs. The behavior of an algorithm should be predictable and consistent.

An algorithm is also expected to have the following:

- **Efficiency**: The algorithm should solve the problem in a timely manner, preferably with the most optimal use of resources.
- **Clarity**: The algorithm should be clear and understandable, making it easier for developers to implement and maintain.
- **Generality**: The algorithm should be applicable to different instances of a problem or adapted to different situations.

Flowcharts

A flowchart is a graphical representation of a process or algorithm. It is a visual tool that helps to illustrate the sequence of steps involved in a process. Flowcharts are used in various fields such as computer programming, economics, industrial processes, and cognitive psychology.

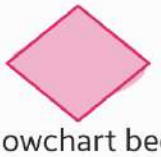
Flowcharts use different symbols to represent different types of steps in a process. The most common symbols used in flowcharts are:

- **Terminal**: The oval symbol indicates the start, stop, and halt in a program's logic flow.

- **Input/Output**: A parallelogram denotes any function of input/output type.

- **Processing**: A box represents arithmetic instructions.

- **Decision**: Diamond symbol represents a decision point.



- **Connectors**: Whenever flowchart becomes complex or it spreads over more than one page, it is useful to use connectors to avoid any confusions. It is represented by a circle.

- **Flow lines**: Flow lines indicate the exact sequence in which instructions are executed. Arrows represent the direction of flow of control and relationship among different symbols of flowchart.

Flowcharts have several advantages such as:

- They illustrate professional models and processes creatively.
- They favor the visual understanding of the processes, by representing them in a simple way.
- They allow the detailed study of the stages of the processes and the definition of the problematic or risky moments, without having to see them live.

However, flowcharts also have some disadvantages such as:

- It is difficult to draw flowcharts for large and complex programs.
- There is no standard to determine the amount of detail.
- Difficult to reproduce the flowcharts.
- It is very difficult to modify the flowchart.

Here are some guidelines to follow when creating a flowchart:

- **Define the process**: Before you start creating a flowchart, make sure you have a clear understanding of the process you want to represent. Identify the inputs, outputs, and steps involved in the process.

- **Determine the symbols to use**: Flowcharts use different symbols to represent different types of steps in a process. The most common symbols used in flowcharts are described above.

- **Start with the start and end points**: Every flowchart must have starting and ending points. They are represented by rounded rectangles or ellipses, with the words "Start" and "End" inside. Since the process may have several different results, depending on the decisions made during the flow, the flowchart can have several possible endpoints, with a circle or oval representing each endpoint. Because you don't know how much space you'll need, begin by drawing the "Start" shape and draw the "End" symbol last.

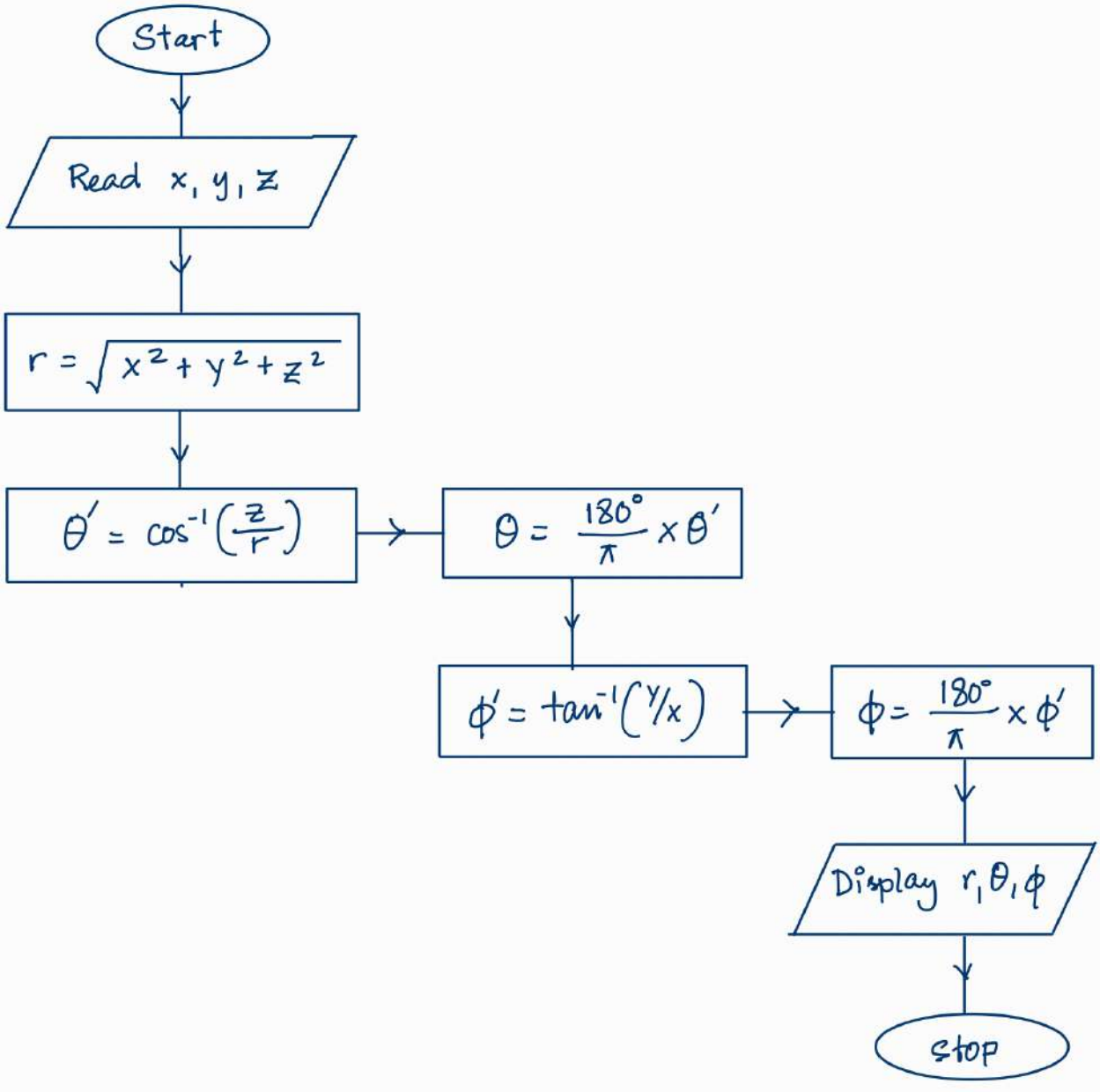
- **Add the actions and decisions**: A rectangle stands for an action or operation that must be taken to move to the next step in the process. The process itself and the variables involved are written inside the shape. For input and output, use parallelograms. The rectangle often contains an action verb, cuing the user as to the action to take. A diamond symbol represents a decision point. Write the specific conditions near the arrows leading from the conditional diamond, which will often be a simple "TRUE" and "FALSE." Where necessary, supplement conditionals with logical "OR" (circles with plus "+" symbols inside) and "AND" (circles with an "x" inside) operators.

- **Connect the symbols**: The lines with arrows indicate the workflow through the process and connect each process. The reader follows the arrows from the start, usually starting from the top down or from left to right, through each input and decision point, until the process reaches the end point. If you draw flowcharts manually, edit shapes and texts before drawing connectors. In other words, before you connect shapes, check and make sure that all steps have been added so that you won't have to make modifications frequently.

- **Review and refine**: Once you have created the flowchart, review it to ensure that it accurately represents the process. Refine the flowchart as necessary to make it clearer and more understandable.

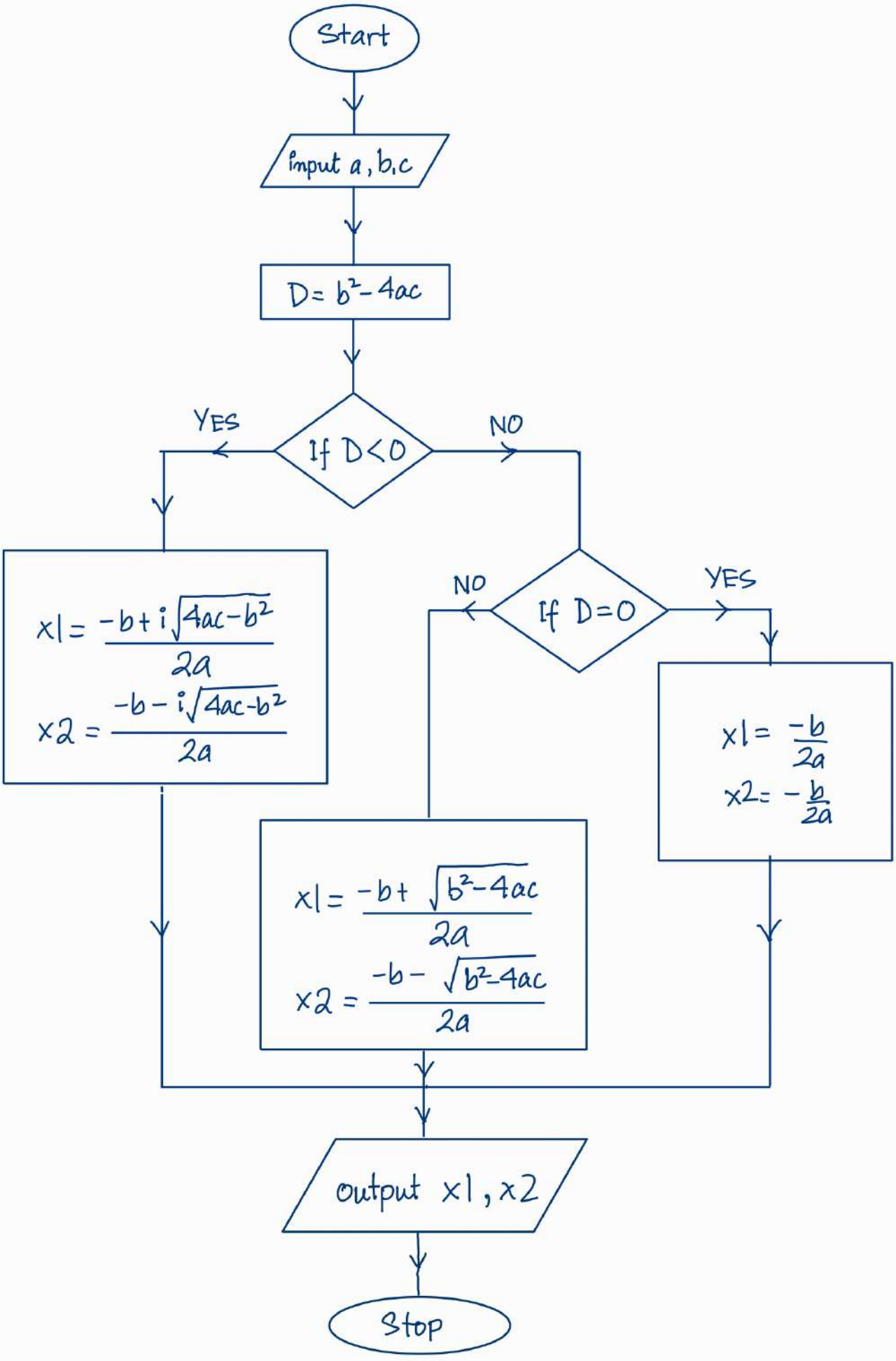
Example 1:

Draw a flowchart to convert from cartesian to spherical polar coordinates.



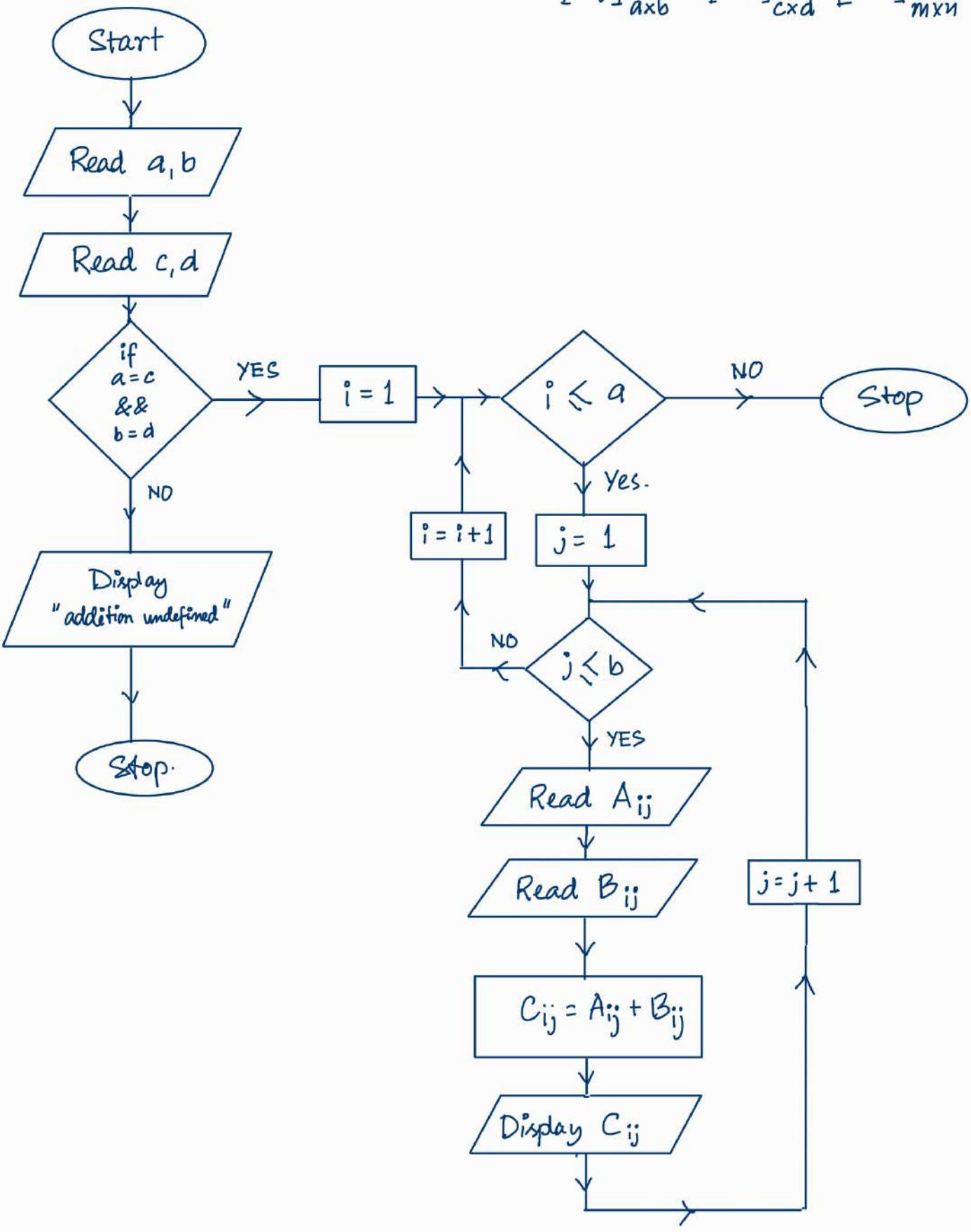
Example 2:

Draw a flowchart to calculate the roots of a quadratic equation: $ax^2 + bx + c = 0$



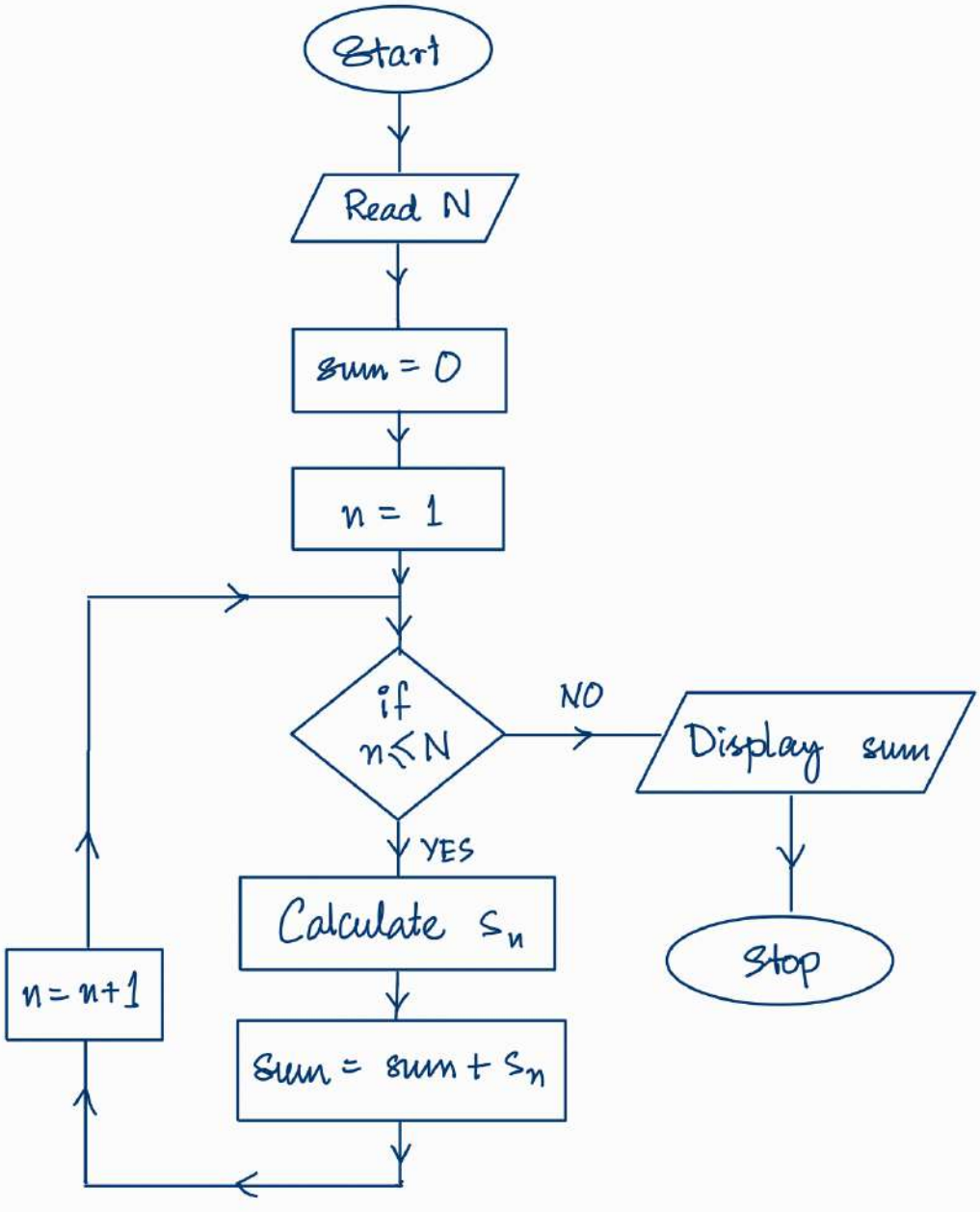
Example 3:

Draw a flowchart to calculate the sum of two matrices. $[A_{ij}]_{axb} + [B_{ij}]_{cxd} = [C_{ij}]_{mxn}$



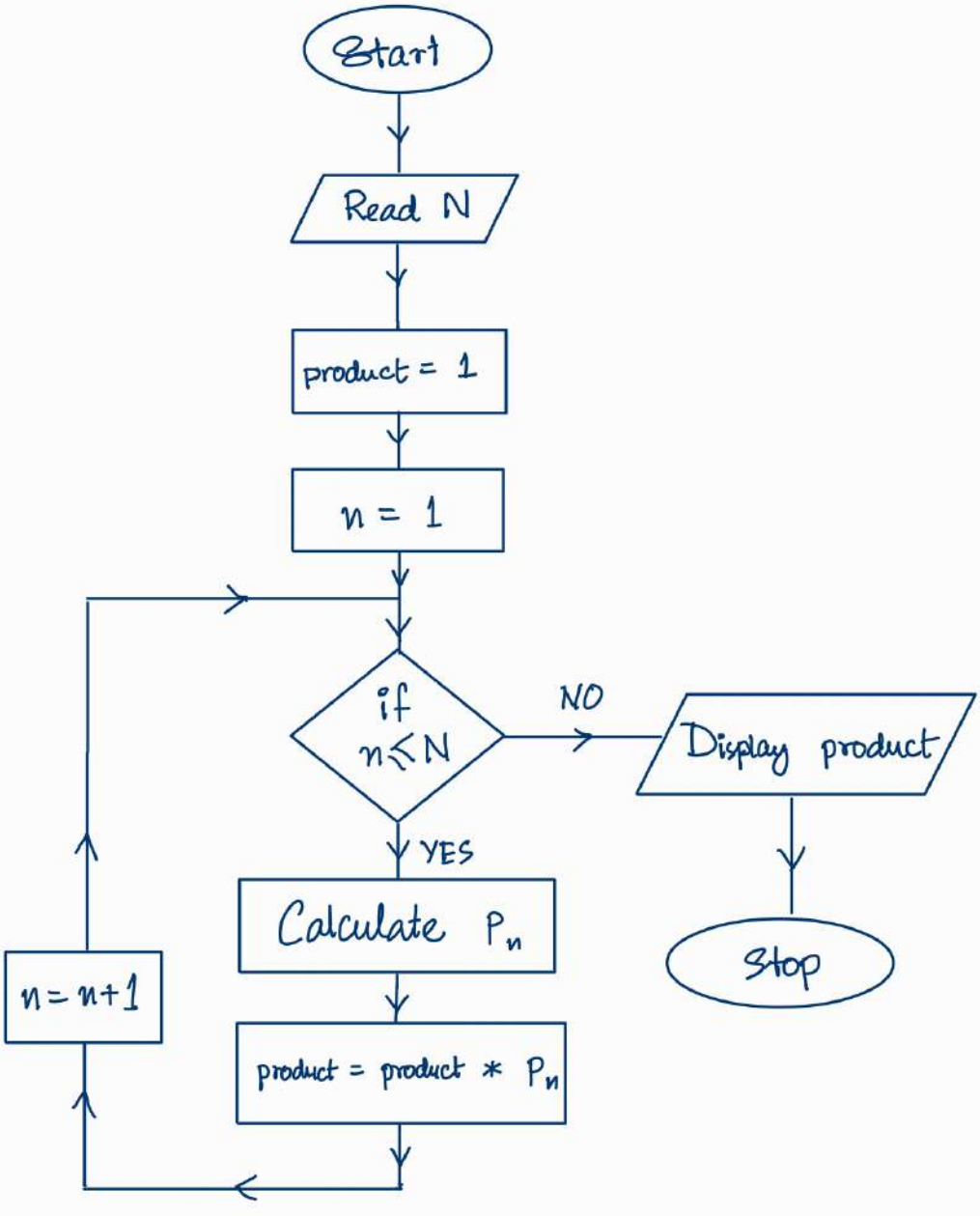
Example 3:

Draw a flowchart to calculate the sum of a finite series. $sum = S_1 + S_2 + S_3 + \cdots + S_N$



Example 4:

Draw a flowchart to calculate the product of a finite sequence. $product = P_1 \times P_2 \times P_3 \times \cdots \times P_N$

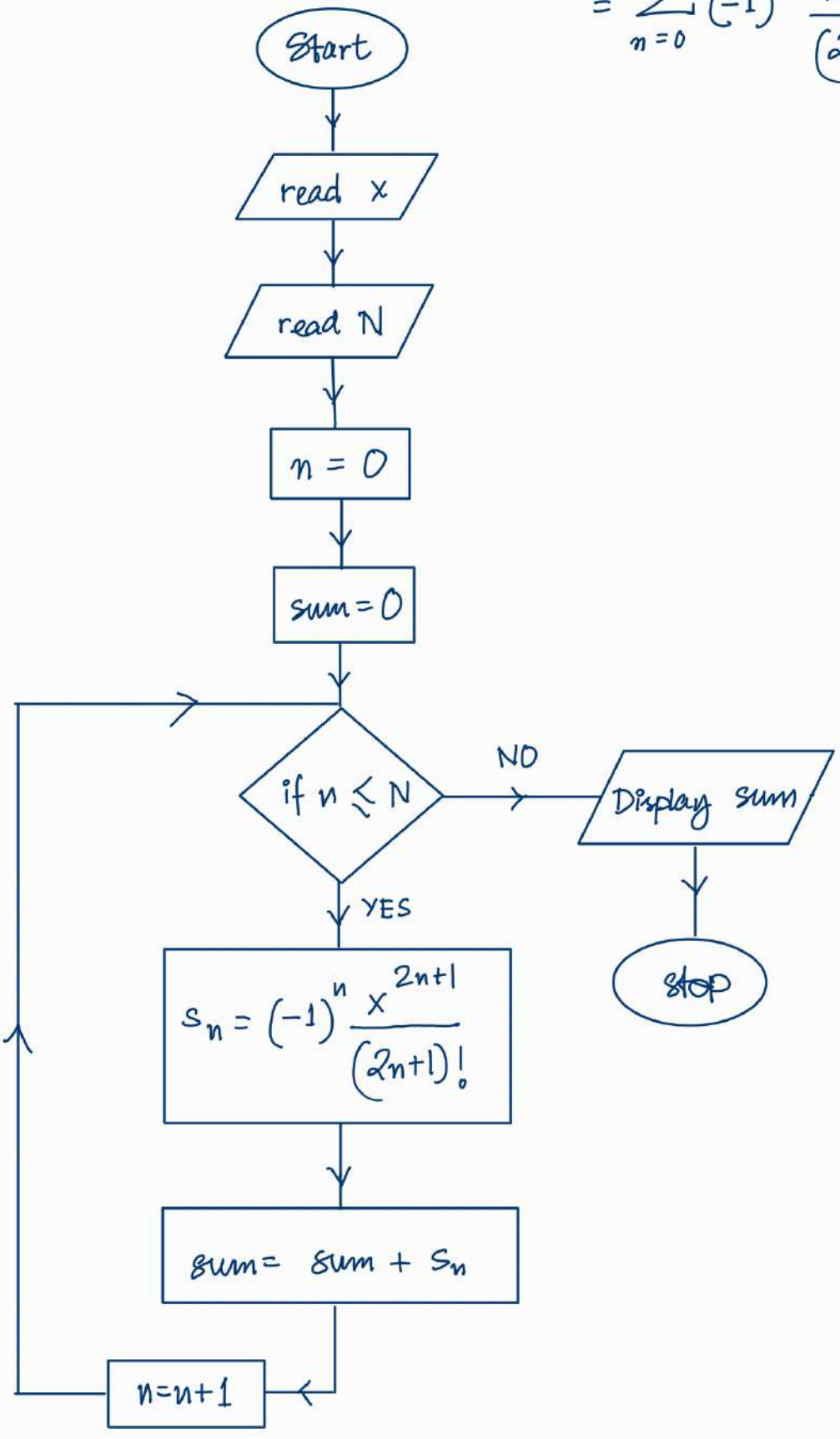


Example 5:

Draw a flowchart to calculate sin(x) as a series.

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \cdots$$

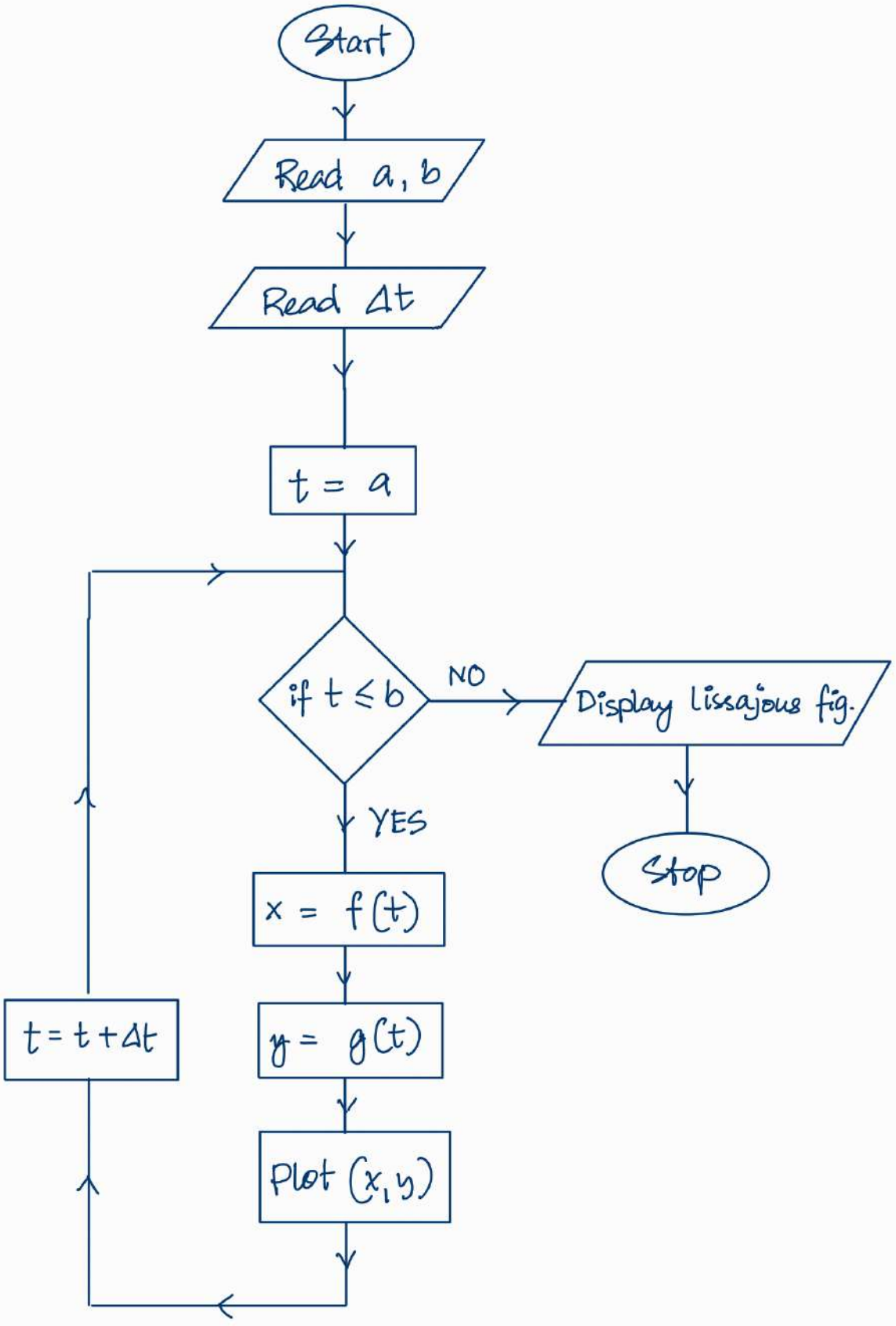
$$= \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!}$$



$t \in [a,b]$; interval Δt

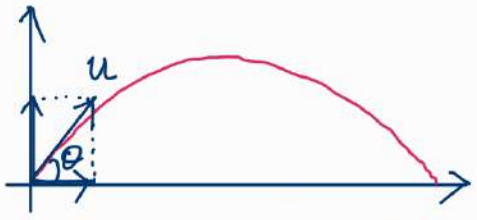
Example 6:

Draw a flowchart for plotting a lissajous figure.



Example 7:

Draw a flowchart to plot the trajectory of a projectile thrown at an angle with the horizontal.



$$x(t) = u\cos(\theta)\,t$$

$$y(t) = u\sin\theta\, t - \frac{1}{2}9.81\,t^2$$

$t \in [a, b]$; interval Δt