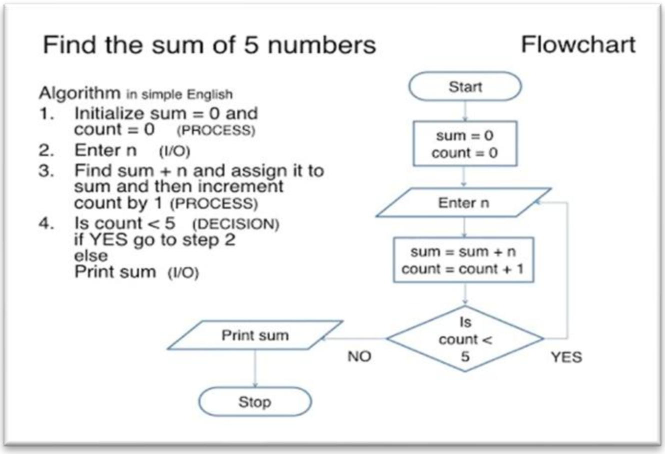


Find the sum of 5 numbers

Algorithm in simple English

1. Initialize sum = 0 and count = 0 (PROCESS)
2. Enter n (I/O)
3. Find sum + n and assign it to sum and then increment count by 1 (PROCESS)
4. Is count < 5 (DECISION)
   if YES go to step 2
   else
   Print sum (I/O)

Example 1: Find the Sum of 5 Numbers

What it does:

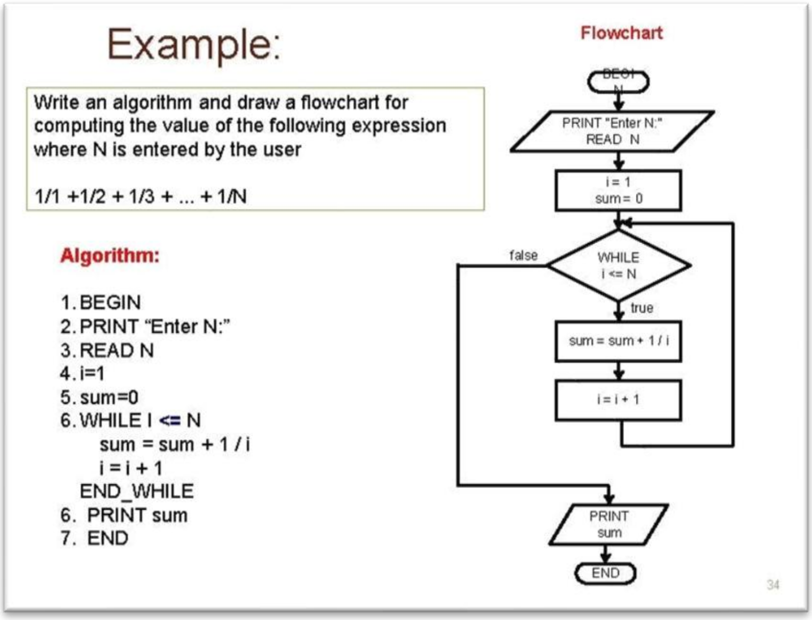
This algorithm takes 5 numbers entered by the user and calculates their sum.

Steps explained:

1. **Initialize** sum and count to 0.
2. **Enter a number (n).**
3. **Add** the number to the sum and increment count by 1.
4. **Check** if count is less than 5:
   - If yes, repeat from step 2.
   - If no, print the sum.

Flowchart explanation:

- Start with sum=0 and count=0.
- Input a number.
- Update sum and count.
- Loop until 5 numbers are entered.
- Print the total sum.



Example:

Write an algorithm and draw a flowchart for computing the value of the following expression where N is entered by the user

1/1 +1/2 + 1/3 + ... + 1/N

Algorithm:

```
1. BEGIN
2. PRINT "Enter N:"
3. READ N
4. i=1
5. sum=0
6. WHILE I <= N
       sum = sum + 1 / i
       i = i + 1
   END_WHILE
6.  PRINT sum
7.  END
```

Example 2: Sum of Series 1 + 1/2 + 1/3 + ... + 1/N

What it does:

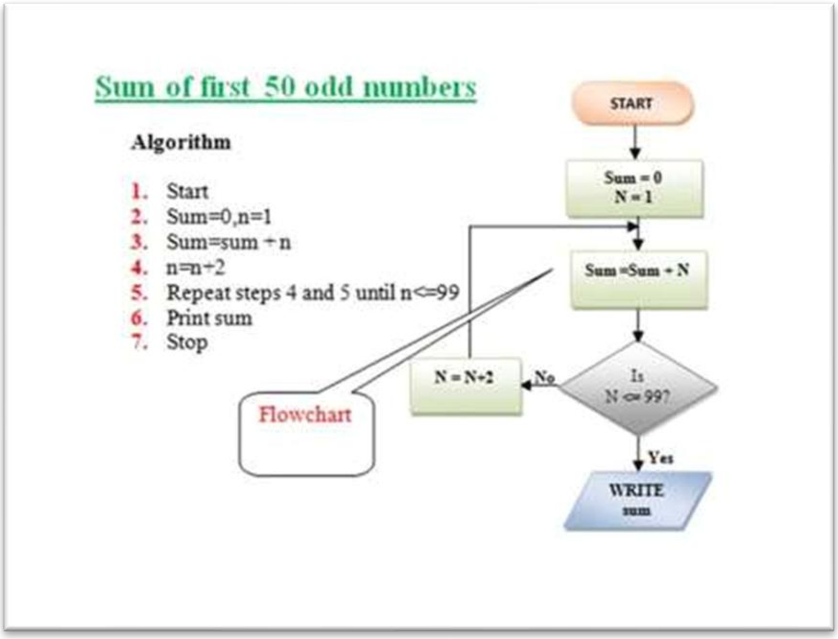
Given an integer N, it computes the sum of the series 1 + 1/2 + 1/3 + … + 1/N.

Steps explained:

1. Start the algorithm.
2. Prompt user to enter N.
3. Initialize i=1 and sum=0.
4. While i <= N, add 1/i to sum and increment i.
5. When the loop ends, print the sum.

Flowchart explanation:

- Read N.
- Loop from i=1 to N.
- In each iteration, add 1/i to sum.
- After loop, print final sum.



Sum of first 50 odd numbers

Algorithm

1. Start
2. Sum=0,n=1
3. Sum=sum +n
4. n=n+2
5. Repeat steps 4 and 5 until n<=99
6. Print sum
7. Stop

Example 3: Sum of First 50 Odd Numbers

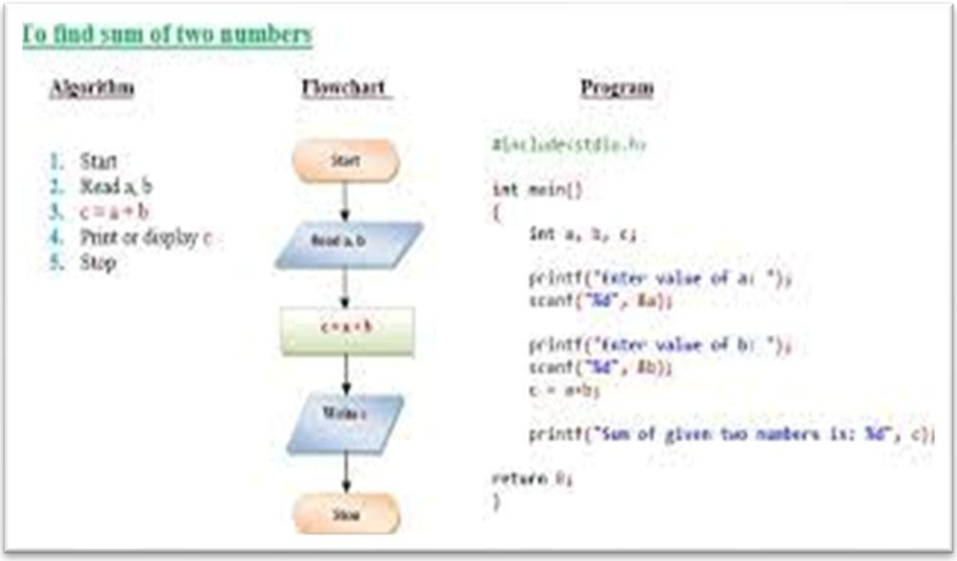
What it does:
Calculates the sum of the first 50 odd numbers (i.e., 1, 3, 5, ..., 99).

Steps explained:

1. Initialize Sum=0 and n=1.
2. Add current odd number (n) to sum.
3. Increment n by 2 to get the next odd number.
4. Repeat until n reaches or exceeds 99 (50th odd number).
5. Print the sum.

Flowchart explanation:

- Start.
- Initialize variables.
- Loop to add odd numbers.
- Check if n > 99.
- If yes, print sum.



To find sum of two numbers

Algorithm

1. Start
2. Read a, b
3. c = a + b
4. Print or display c
5. Stop

Flowchart

Program

```
#include<stdio.h>

int main()
{
    int a, b, c;

    printf("Enter value of a: ");
    scanf("%d", &a);

    printf("Enter value of b: ");
    scanf("%d", &b);
    c = a+b;

    printf("Sum of given two numbers is: %d", c);

return 0;
}
```

Example 4: Sum of Two Numbers

What it does:
Reads two numbers a and b from the user and computes their sum.

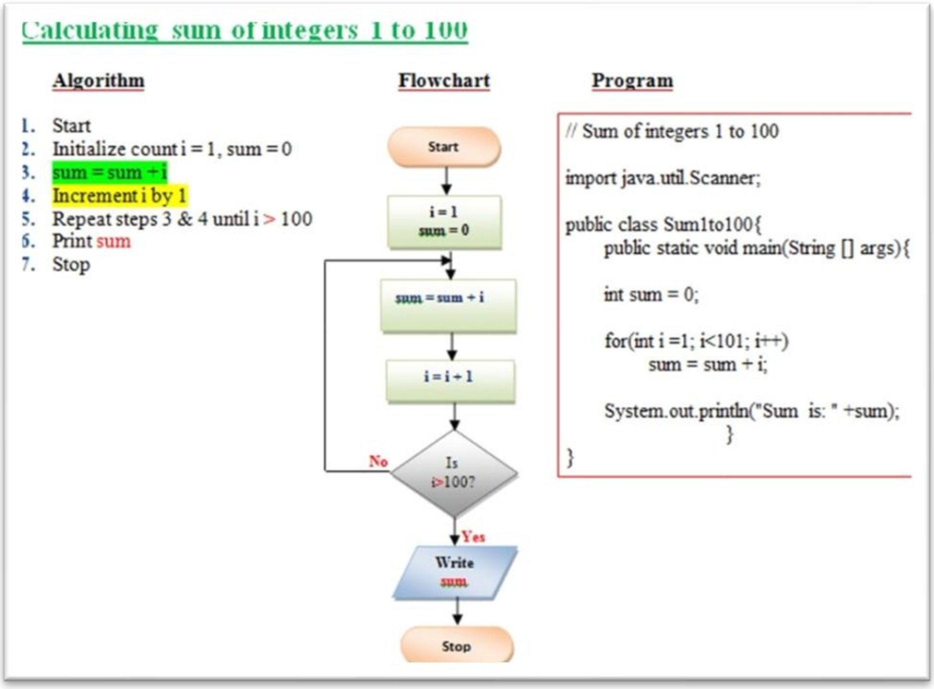
Steps explained:

1. Start.
2. Read two numbers.
3. Add them (c = a + b).
4. Print or display the result.
5. Stop.

Flowchart explanation:

- Start.
- Read values of a and b.
- Compute their sum.
- Output the result.
- End.

Program snippet shown:
Basic C program that reads two integers and prints their sum.



Calculating sum of integers 1 to 100

Algorithm

1. Start
2. Initialize count i = 1, sum = 0
3. sum = sum + i
4. Increment i by 1
5. Repeat steps 3 & 4 until i > 100
6. Print sum
7. Stop

Flowchart

Program

```
// Sum of integers 1 to 100

import java.util.Scanner;

public class Sum1to100{
      public static void main(String [] args){

      int sum = 0;

      for(int i =1; i<101; i++)
             sum = sum + i;

      System.out.println("Sum  is: " +sum);
                        }
}
```

Example 5: Sum of Integers 1 to 100

What it does:
Calculates the sum of integers from 1 to 100.

Steps explained:

1. Initialize i=1 and sum=0.
2. Add i to sum.
3. Increment i by 1.
4. Repeat steps 2 and 3 until i exceeds 100.
5. Print the sum.

Flowchart explanation:

- Start.
- Initialize i and sum.
- Loop from i=1 to 100, adding i to sum each time.
- After i>100, output the sum.

Program snippet shown:
Java program doing the same calculation with a loop.