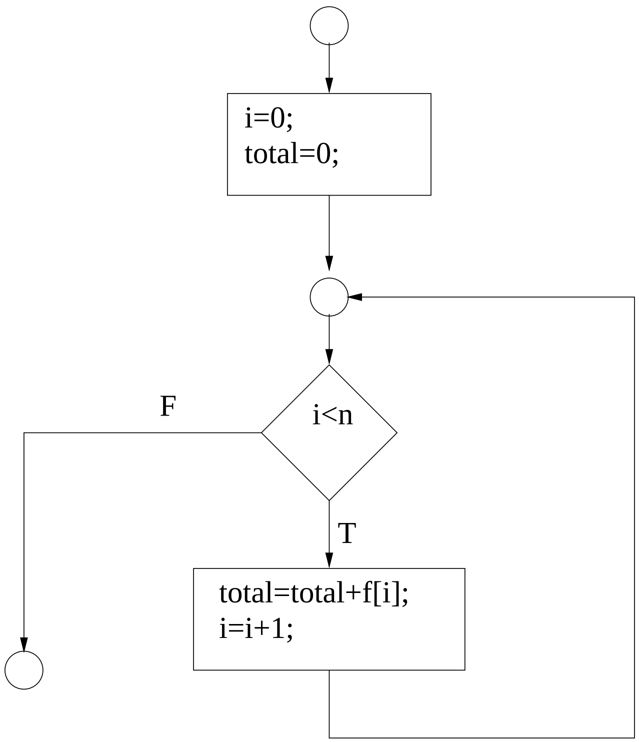
- In the last lecture we:
  - Constructed a recursive definition of a function to Sum to sum the numbers in a list
  - Demonstrated how to apply the recursive definition to calculate the sum of 5 numbers.
  - Wrote a Java method to implement the recursive definition recursively
- In this lecture:
  - Implement the recursive definition of Sum iteratively.
  - Draw a flowchart for this iterative implementation.
  - Annotate the flowchart with assertions.
  - Prove by induction that the iterative implementation of the recursive definition is correct.

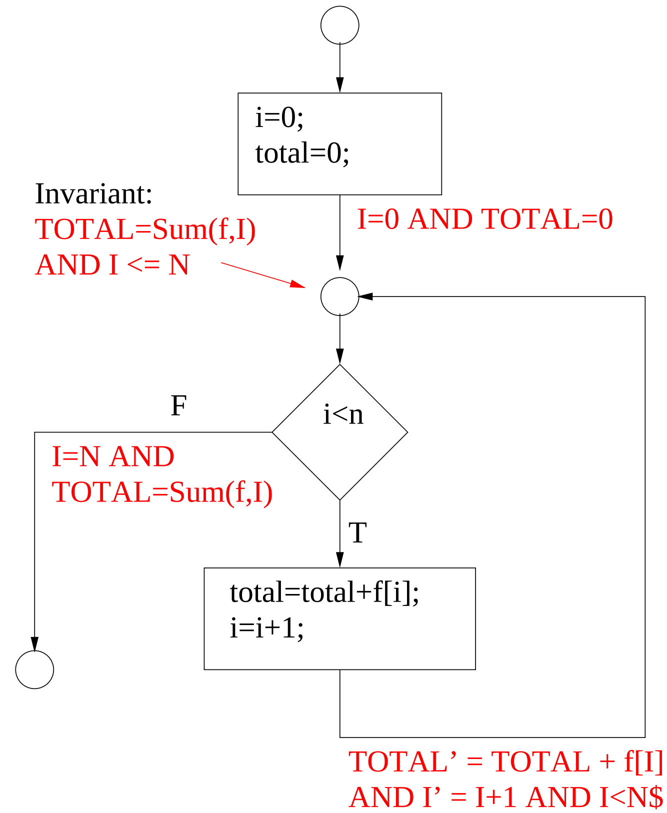
- Reminder: Recursive definition:
  - Base Case: $Sum(f, 0) = 0$
  - Recurrence Relation: $Sum(f, n) = Sum(f, n-1) + f(n-1)$
- An iterative implementation of Sum.

```
int SumMethod(int f[], int n)
{
    int i=0;
    int total=0;
    /* initialisation corresponding
                to the base case */
    while(i<n)
    {
        total=total+f[i];
i=i+1;
        // implementation of recurrence rela
    }
    return total;
}
```

- A flowchart for $SumMethod$, the iterative implementation of Sum.



- The flowchart annotated with assertions



- Now we will use the assertions added to the flowchart to show that the implementation of SumMethod using the while loop is correct.

- This is a proof by induction which proceeds as follows:

  - Basis Step: Show the property P holds for the basis value (This is often 0). i.e. show $P(0)$ holds

  - Assume the inductive hypothesis: Assume $P(i)$ holds.

  - Inductive step: Show that $P(i+1)$ holds.

- The property we wish to prove for the iterative implementation is that when the program terminates the variable $total$ in the method $SumMethod$ holds the sum of the n elements in the list.

- More formally: $TOTAL = Sum(f, N)$. This assertion can be deduced from the assertion on the false branch of the if-statement in the flowchart.

- Proof by Induction

  - Basis step: $P(0)$: Show that if the loop is never executed that $TOTAL = Sum(f, 0)$ Here $N = 0$.

  - After initialising i and $total$ the assertion:
    $I = 0 \wedge TOTAL = 0$ holds
    From the inductive definition (base case):
    $Sum(f, 0) = 0$
    Therefore $TOTAL = Sum(f, 0)$
    $N = 0$ also
    Therefore $TOTAL = Sum(f, 0)$ holds.
    This shows that the base case holds

- Assume inductive hypothesis: In proofs like this one this means that you assume that the invariant holds after i iterations.

- Therefore, assume $TOTAL = Sum(f, I) \wedge I \leq N$

- Inductive Step: Prove that the property holds after $i + 1$ iterations (i.e. one more iteration). Note the variables $total$ and i will change their values after each iteration.

- Note: $TOTAL'$ is the new value for $total$ and I' is the new value for i

- Therefore we must prove:
  $TOTAL' = Sum(f, I') \wedge I' \leq N$

- If the loop is executed one more time then (see assertion in flowchart):
  $TOTAL' = TOTAL + f[I] \wedge I' = I + 1 \wedge I < N$

- From the inductive hypothesis: $TOTAL = Sum(f, I)$

- Therefore $TOTAL' = Sum(f, I) + f[I]$

- From Recurrence relation:
  $Sum(f, I) + f[I] = Sum(f, I+1) = Sum(f, I')$

- Since $I < N$, $I + 1 \leq N$

- Putting this information together:
  $TOTAL = Sum(f, I') \wedge I' \leq N$