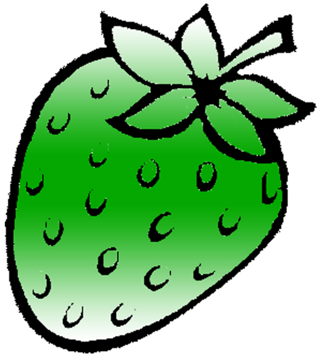
STRAWBERRY



/strawberrydevelopers

/strawberry_app

For more visit:

Strawberrydevelopers.weebly.com

Computer Programming –I

Course Outcome

- After Successful completion of this course, students will be able to
  - Illustrate flowchart and algorithm for a given problem
  - Develop and Execute C programs using basic programming constructs.
  - Implement C programs using various data types and functions
  - Solve problems using Pointers

Algorithms & Flowchart

What is an Algorithm?

- Computer Scientist Niklaus Wirth stated that
  - Program = Algorithms + Data
  - Algorithm : Procedure for solving a problem in finite number of steps.

Example for Algorithm

- Making a pot of tea:
  1. If the kettle does not contain water, then fill the kettle.
  2. Plug the kettle into the power point and switch it on.
  3. If the teapot is not empty, then empty the teapot.
  4. Place tea leaves in the teapot.
  5. If the water in the kettle is not boiling, then go to step 5.
  6. Switch off the kettle.
  7. Pour water from the kettle into the teapot.

Example for Algorithm

- Making a pot of tea:
  1. If the kettle does not contain water, then fill the kettle. **(Decision making)**
  2. Plug the kettle into the power point and switch it on.
  3. If the teapot is not empty, then empty the teapot. **(Decision making)**
  4. Place tea leaves in the teapot.
  5. If the water in the kettle is not boiling, then go to step 5. **(Repetition)**
  6. Switch off the kettle.
  7. Pour water from the kettle into the teapot.

Algorithm Features

- Sequence (also known as process)
- Decision (also known as selection)
- Repetition (also known as iteration or looping)

Sequence

- Each step or process in the algorithm is executed in the specified order.

The decision constructs – if…then..else

- Outcome of a decision – either true or false.
  - If today is Friday then collect pay.
    - It is either true that ‘today is Friday’ or it is false that ‘today is not Friday’.
    - It cannot be both.
- General Form:

```
 If Proposition
     then process1
else
  process 2
```

```
If Friday
   then collect pay
else
    call on next Monday
```

Repetition

- Repeat a process or sequence of process until a condition becomes true.
- General Form:

```
Repeat                          Repeat
    Process1                        Fill water in kettle
    Process2                    Until Kettle is full
    ………
    ……..
    Process N
    Until Proposition
```

Repetition - While

- What happens if the kettle is already full?
- While loop more appropriate.

```
While proposition
 begin

Process1
```

```
While kettle is not full
    fill water in kettle.
```

Repetition –If then goto

1. Fill some water in kettle
2. If kettle not full then goto 1.

Variables

1. Program consists of algorithm (Process) and data.
2. Data needs to be stored.
3. Data is contained(stored) in what is called a ‘variable’.
4. The variable is a container of a value that may change during the execution of the program.

Variables

1. Each variable in a program is given a name
2. Example: Water_Level, Water_Temprature, Num1, Num2

How to use variable in algorithm?

If Water_Level is 0 then fill the kettle.

Rules for algorithm

1. Each algorithm will be logically enclosed by two statements START and STOP.

2. To accept data from user, the INPUT or READ statements are to be used.

3. To display any user message, or the content in a variable PRINT statement will be used.

Rules for algorithm –Arithmetic Operators

1. '←' Assignment operator

x← 6 means the value 6 is assigned to the variable x.

Z←X+Y (Value of X + Value of Y) assigned to Z

Z←X-Y

X←5

Y←6

Z←X*Y

Z←X/Y

Rules for algorithm – Relational Operators

1. ‘>’ greater than

X>Y means if the value contained in X is larger than that in Y then outcome of the expression is true else outcome of expression is false.

‘<=‘ Less than or equal to

X<=Y means that if the value held in X is either less than or equal to the value held in Y then outcome of the expression is true else false.

Other relational operators:

>, >=, =, != (Not equal to)

Rules for algorithm – Logical operators

1. ‘AND’ conjunction

The outcome of the expression is true when both propositions are true otherwise it is false.

```
X←2
Y←1
```

X=2 AND Y=0

X=2 true but Y=0 is false

So outcome of total expression is false.

Rules for algorithm – Logical operators

2. ‘OR’ Disjunction

The outcome of an expression is true when anyone of the propositions is true else it is false.

X$\leftarrow$2

Y$\leftarrow$1

X=2 OR Y=0

X=2 is true but Y=0 is false

But as ‘OR’ is used as one proposition is true result for entire expression will be true.

Algorithm Example

Algorithm for finding the sum of any two numbers.

Solution: Let the two numbers be A and B and let their sum be equal to C. Then, the desired algorithm is given as follows:

```
1.  START
2.  PRINT “ENTER TWO NUMBERS”
3.  INPUT A, B
4.  $C \leftarrow A+B$
5.  PRINT c
6.  STOP
```

Algorithm Example

Algorithm for determining the remainder of a division operation where the dividend and divisor are both integers.

Solution: Let N and D be the dividend and divisor, respectively. Assume Q to be the quotient, which is an integer, and R to be the remainder. The algorithm for the given problem is as follows:

```
1. START
2. PRINT “ENTER DIVIDEND”
3. INPUT N
4. PRINT “ENTER DIVISOR
5. INPUT D
6. Q←N/D (INTEGER DIVISION)
7. R← N-Q*D
8. PRINT R
9. STOP
```

Algorithm Example

Construct the algorithm for interchanging the numeric values of two variables.

Hint: If A and B are two variables with values 5 and 6 respectively. Interchange values of A and B, so that value of A will become 6 and value of B will become A. Use third variable C for it.

Algorithm Example

Solution: Let the two variables be A and B. Consider C to be a third variable that is used to store the value of one of the variables during the process of interchanging the values.

The algorithm for the given problem is as follows:

```
1. START
2. PRINT "ENTER THE VALUE OF A & B"
3. INPUT A, B
4. C←A
5. A←B
6. B←C
7. PRINT A,B
8. END
```

Algorithm Example

Write an algorithm that compares two numbers and prints either the message identifying the greater number or the message stating the both numbers are equal.

Solution: Let A and B be two variables to represent the two numbers that are being compared. The algorithm for this problem is given as follows:

```
1.  START
2.  PRINT "ENTER TWO NUMBERS"
3.  INPUT A,B
4.  IF A>B THEN
        PRINT "A IS GREATER THAN B"
5.  IF B>A THEN
        PRINT "B IS GREATER THAN A"
6.  IF A=B THEN
        PRINT "BOTH ARE EQUAL"
7.  STOP
```

Algorithm Example

Write an algorithm print the largest number among three numbers.

Solution: Let the three numbers be represented by A,B and C. Solution for the algorithm is as follows:

Solution 1

```
1.    START
2.    PRINT "ENTER THREE
      NUMBERS"
3.    INPUT A,B,C
4.    IF A>=B AND B>=C
     THEN PRINT A
5.    IF B>=C AND C>=A
     THEN PRINT B
6.    ELSE
      PRINT C
7. STOP
```

Solution 2

```
1.  START
2.  PRINT "ENTER THREE
    NUMBERS"
3.  INPUT A, B,C
4.  IF A>C THEN
      IF A>C THEN
         PRINT A
      ELSE
         PRINT C
    ELSE IF B>C THEN
         PRINT B
       ELSE
         PRINT C
5. STOP
```

Algorithm Example

Take three sides of a triangle as input and check whether the triangle can be drawn or not. Classify the triangle as equilateral, isosceles, or scalene.

Solution: Let the length of three sides of triangle be represented by A,B and C. Solution for the algorithm is as follows:

```
Solution 1
1.     START
2.     PRINT "ENTER LENGTH OF THREE SIDES OF A TRAINGLE"
3.     INPUT A,B, C
4.     IF A+B>C AND B+C>A AND A+C >B
           THEN PRINT"TRAINGLE CAN BE DRAWN"
       ELSE
           PRINT "TRIANBLE CANNOT BE DRAWN"  :GO TO 6.
5.   IF A=B AND B=C THEN
          PRINT "EQUILATERAL"
     ELSE
        IF A!=B AND B!=C THEN
          PRINT "SCALENE"
     ELSE
        PRINT "ISOSCELES"
6. STOP
```

Algorithm Example

Construct an algorithm for incrementing the value of a variable that starts with an initial value of 1 and stops when the value becomes 5.

Solution: Let the variable be represented by C. The algorithm for the said problem is given as follows:

```
1. START
2. C← 1
3. WHILE C<=5
4. BEGIN
5. PRINT C
6. C← C+1
7. END
```

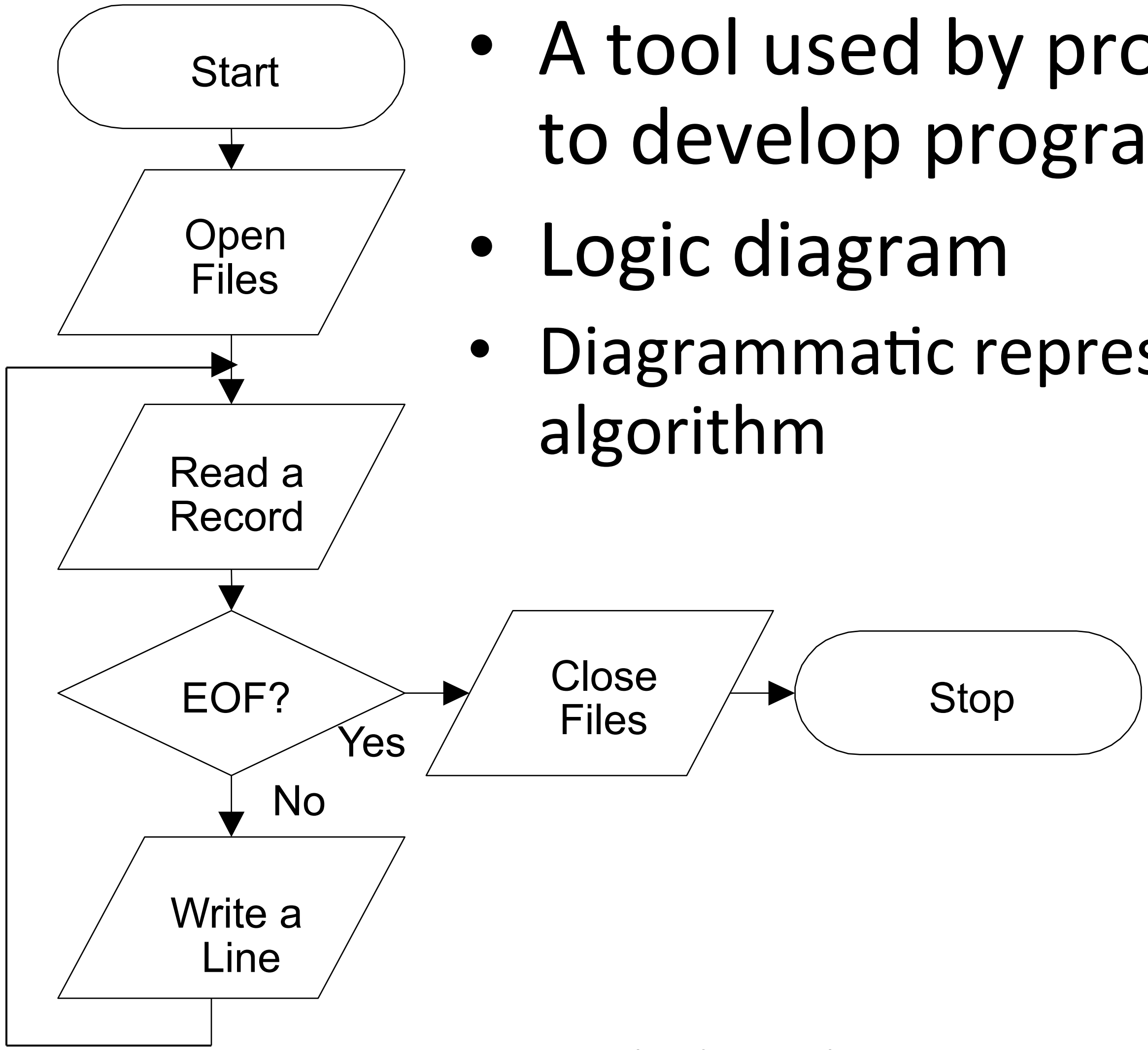
Algorithm Example

Write an algorithm for the addition of N given numbers.

Solution: Let the sum of N given numbers be represented by S.

```
1.  START
2.  PRINT “HOW MANY NUMBERS?”
3.  INPUT N
4.  S← 0
5.  C←1
6.  PRINT “ENTER NUMBER”
7.  INPUT A
8.  S← S+A
9.  C←C+1
10. IF C<=N THEN GOTO 6
11. PRINT S
12. STOP
```

What is a program flowchart?

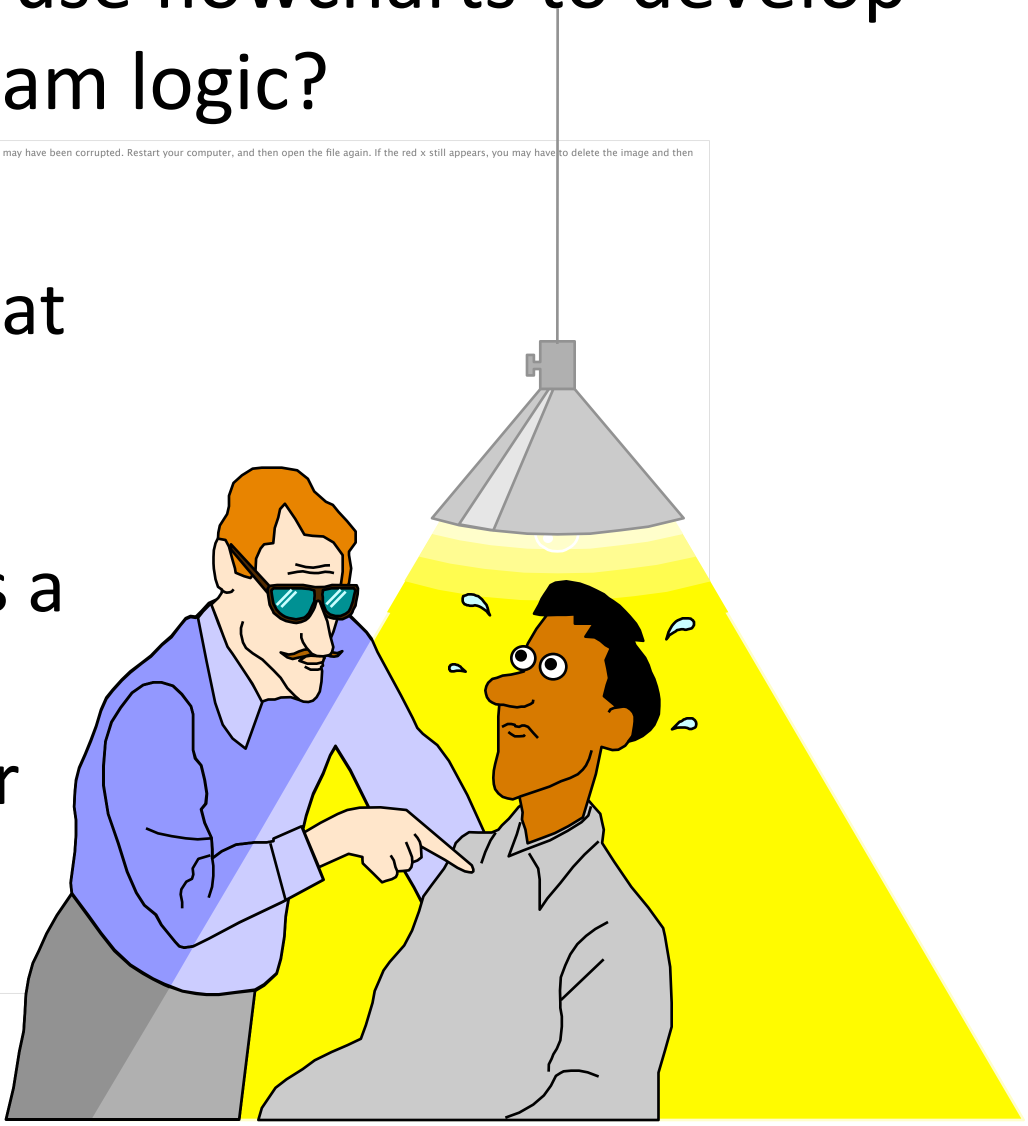


- A tool used by programmers to develop program logic
- Logic diagram
- Diagrammatic representation of algorithm

Why do programmers use flowcharts to develop program logic?

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

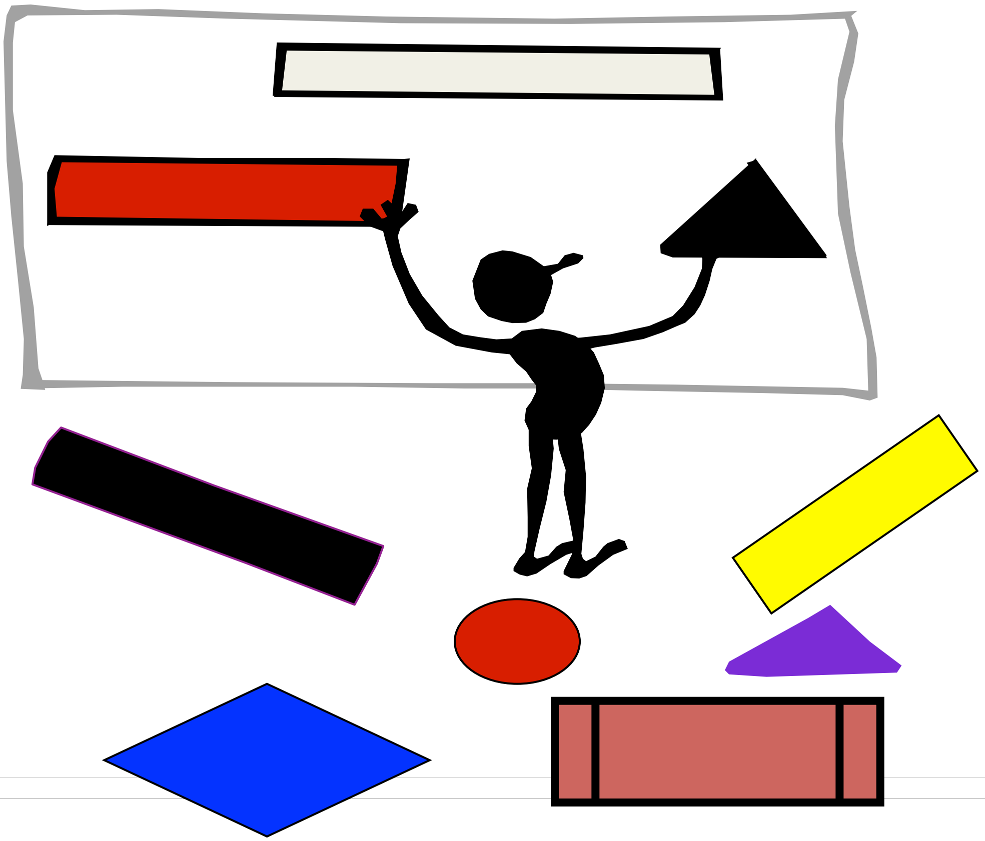
- Flowcharts are a good visual reference of what the program will do
- Serve as program documentation and as a communications tool
- Allow the programmer to test alternative solutions



What are the flowchart symbols?

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

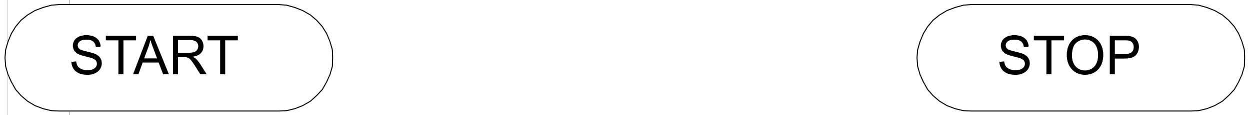


Terminal Symbol

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

Designates the beginning or end of a program or routine

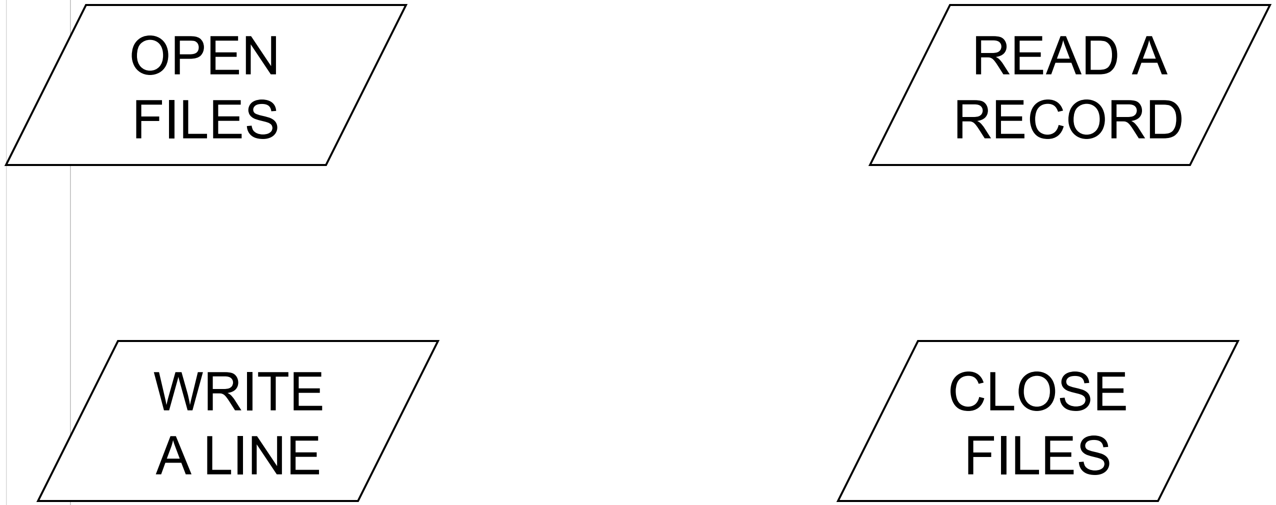


Input/Output (I/O) Symbol

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

Denotes an operation involving an input or output device

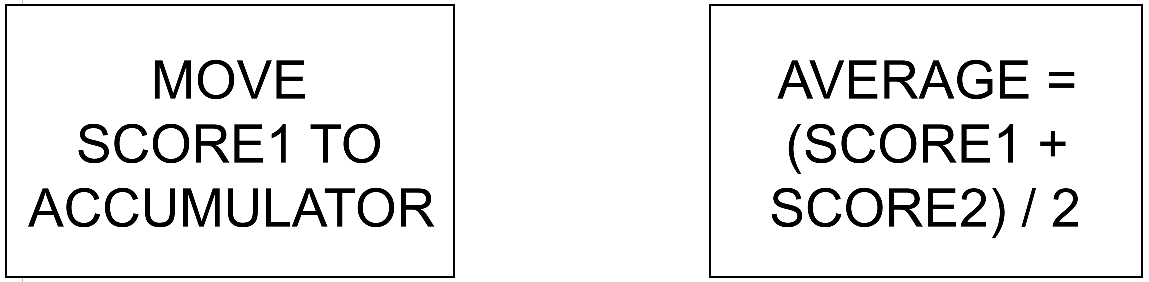


Process Sym

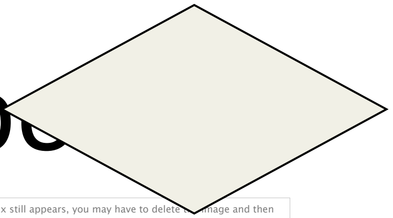
The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

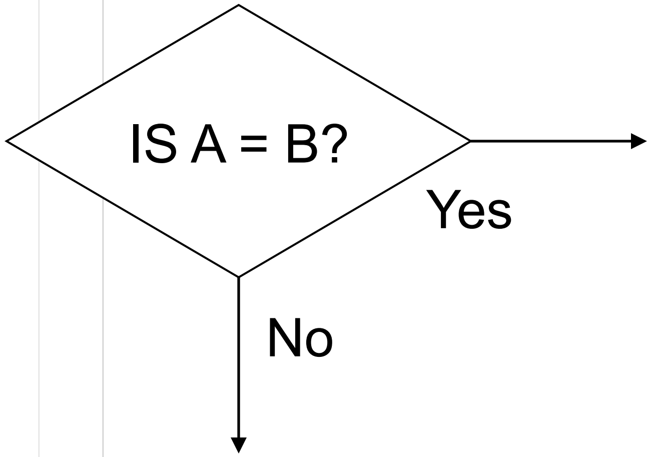
Used for move instructions and arithmetic instructions

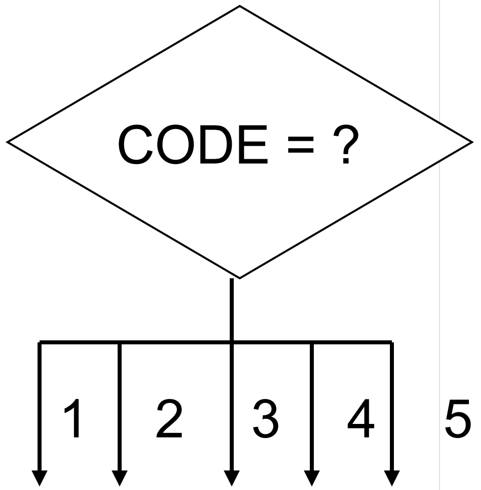


Decision Symb

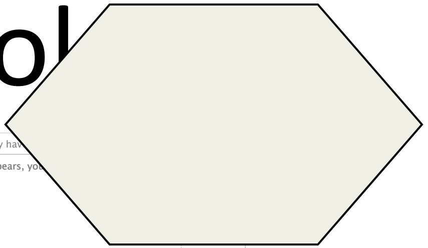


Designates that different logic paths will be followed based on a condition occurring in a program

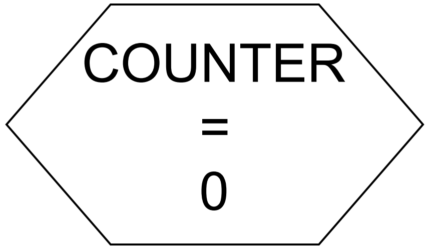




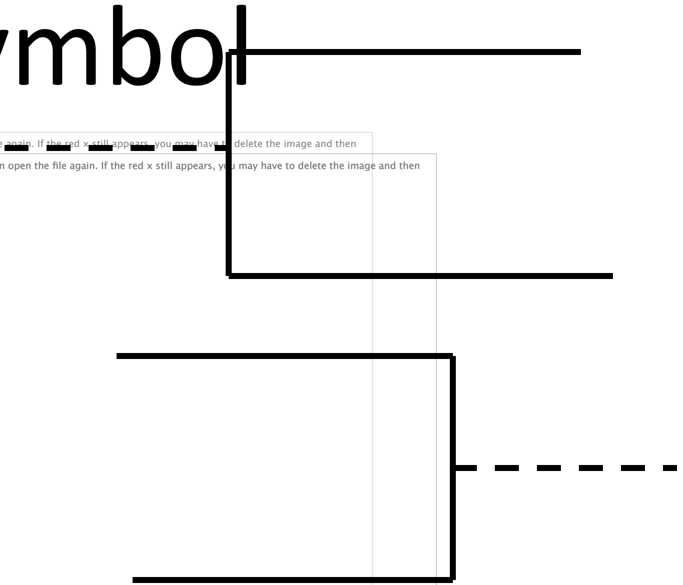
Preparation Symbol



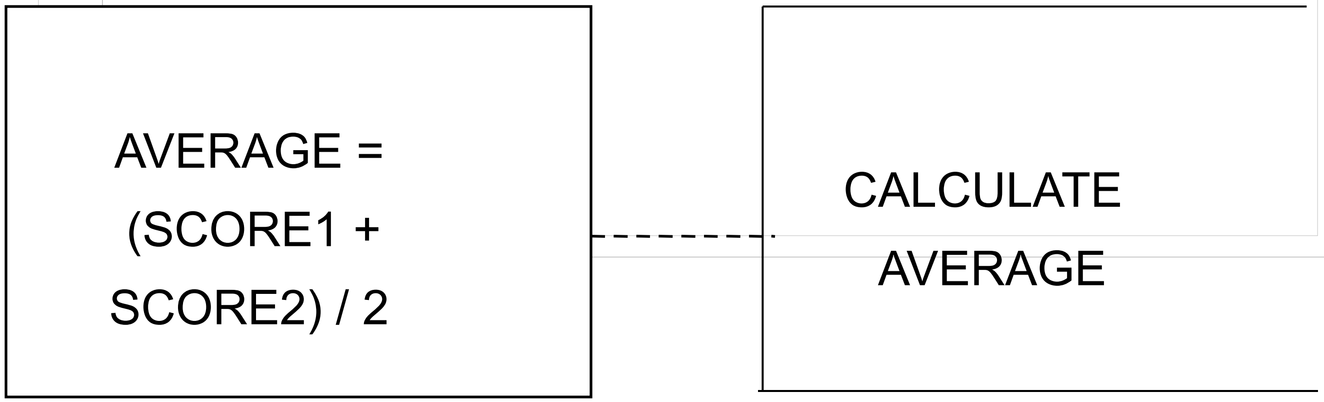
Assigns initial values to areas in storage and does whatever else is necessary to get ready for processing



Annotation Symbol

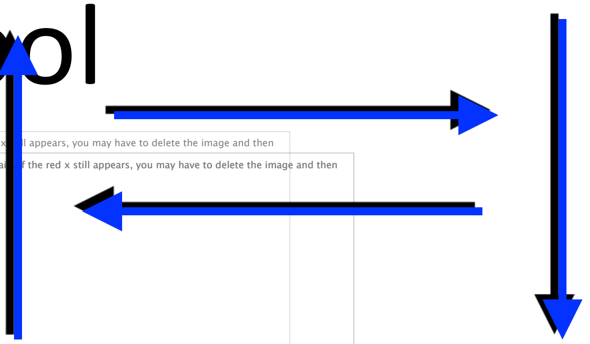


Describes or shows comments on the flowchart



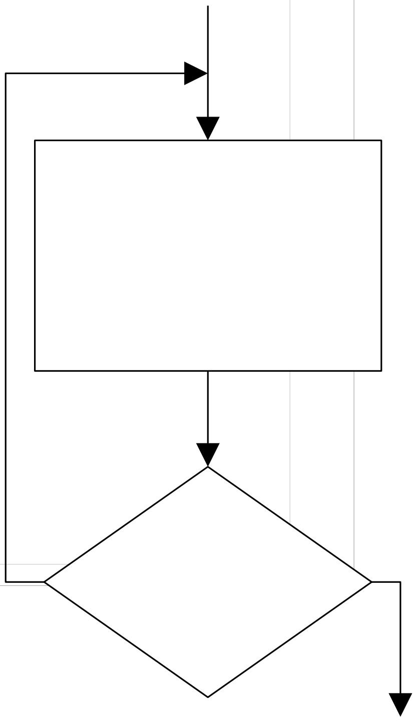
Flowline Symbol

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file agai[illegible]f the red x still appears, you may have to delete the image and then insert it again.

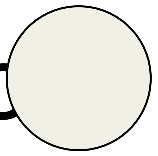


Links other symbols and indicates the sequence of operations

- Normal flow is top to bottom and left to right
- Arrowheads must be used when the flow is other than the normal flow

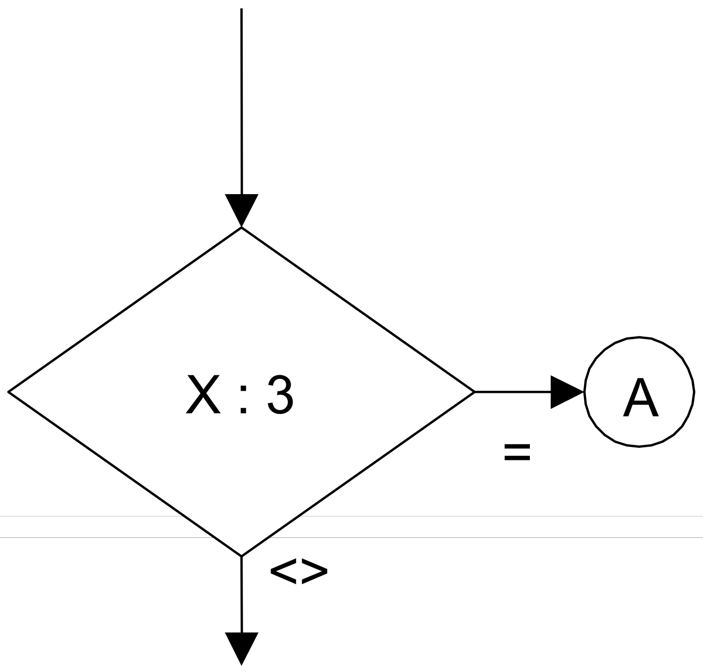


Connector Symbo



The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

Connects parts of the flowchart when the use of flowlines would be confusing or otherwise undesirable

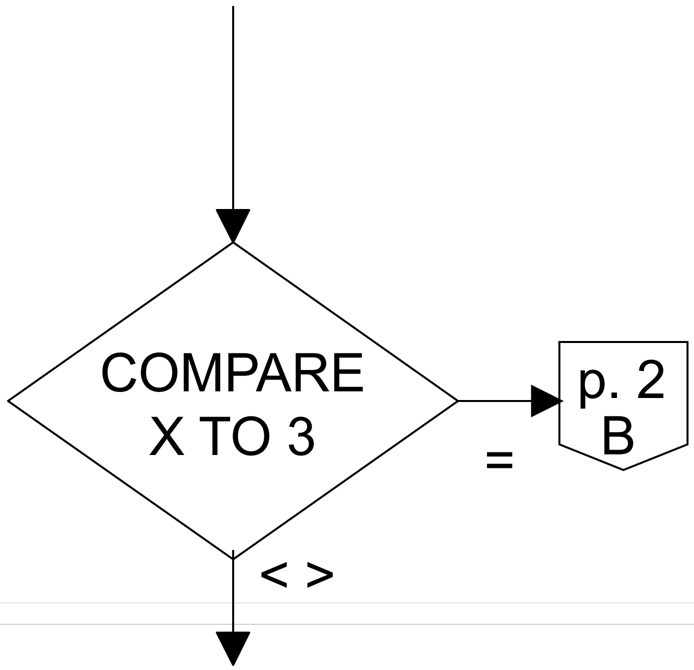


Off-page Connector Symb

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

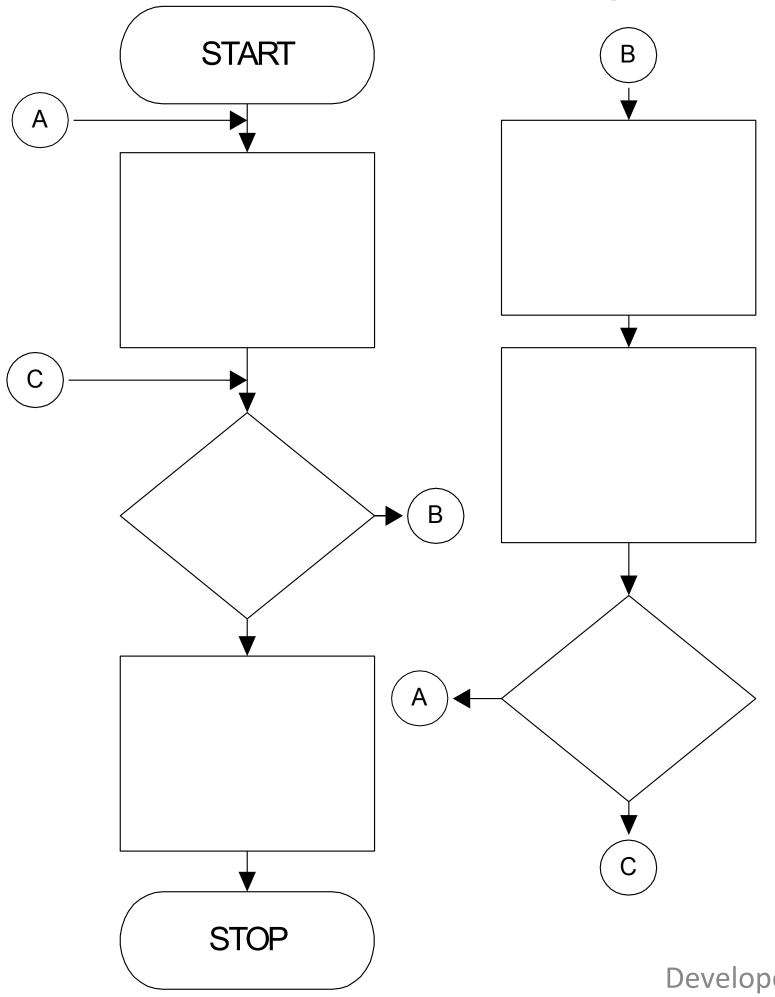
The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

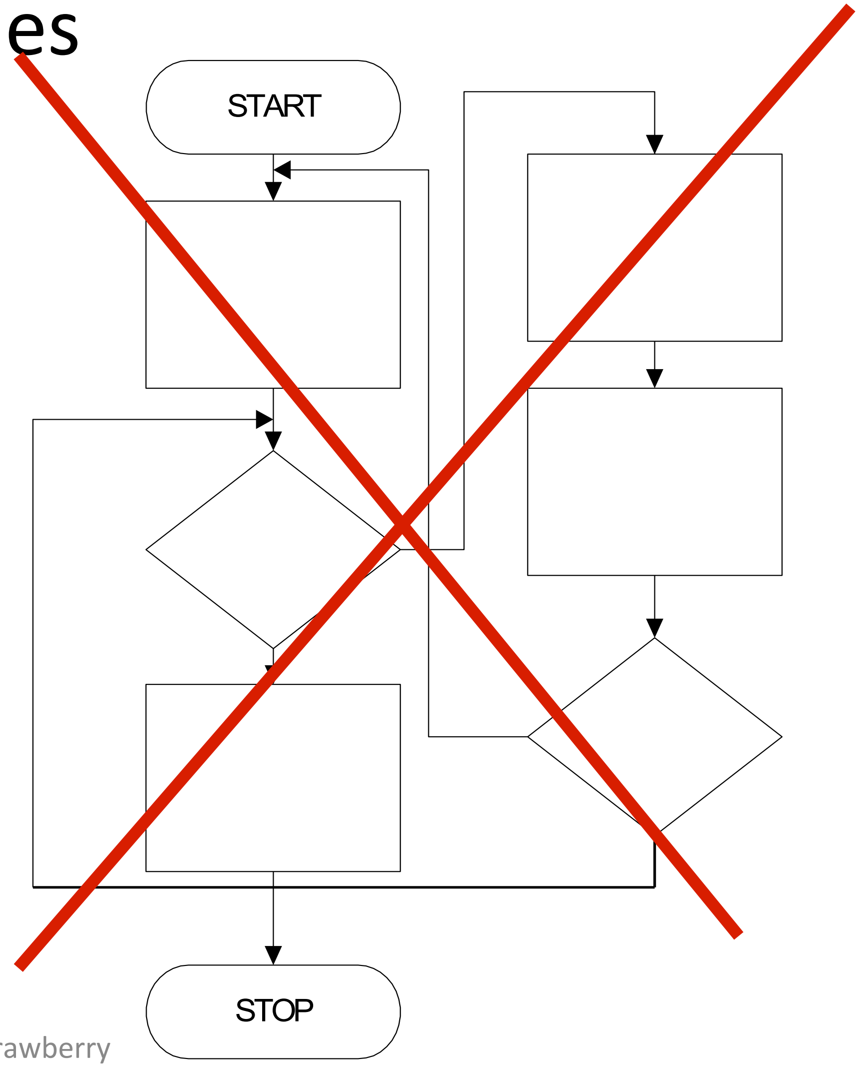
Connects parts of the flowchart from one page to another



More Rules...

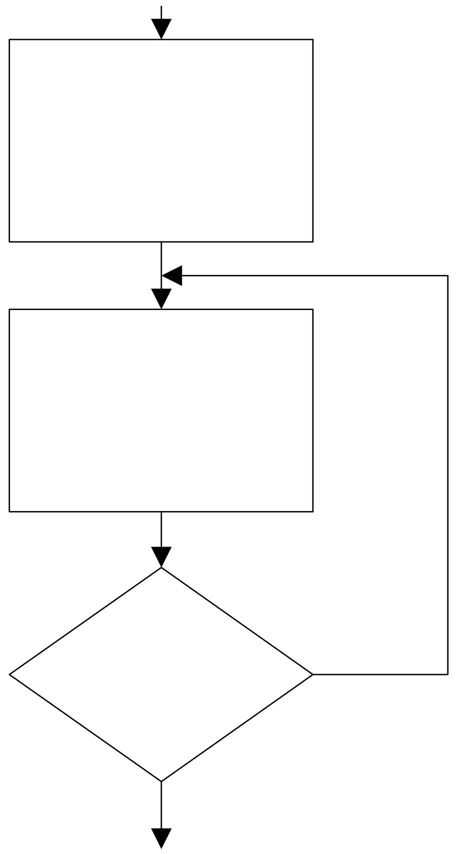
- Avoid intersecting flowlines

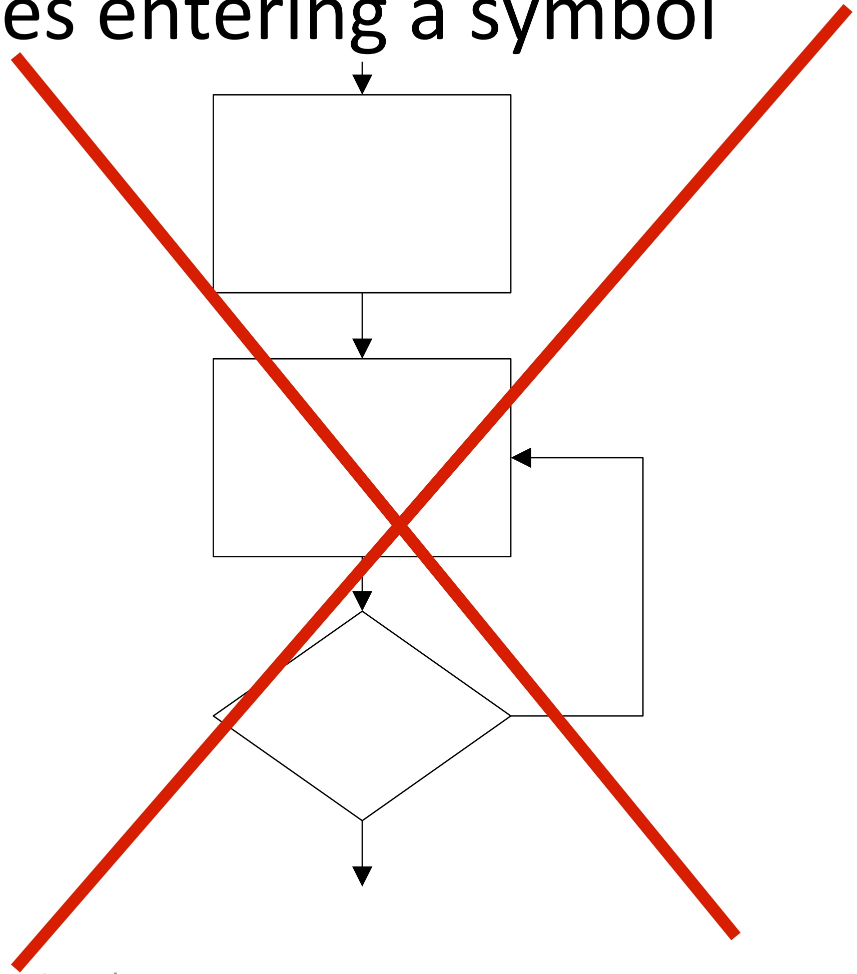




More rules

- Avoid multiple flowlines entering a symbol

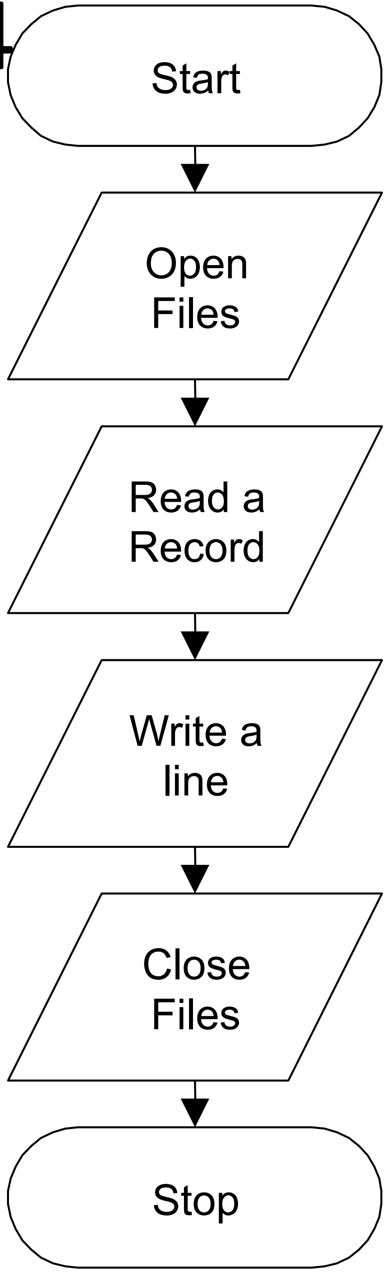




Example in Section 2.4

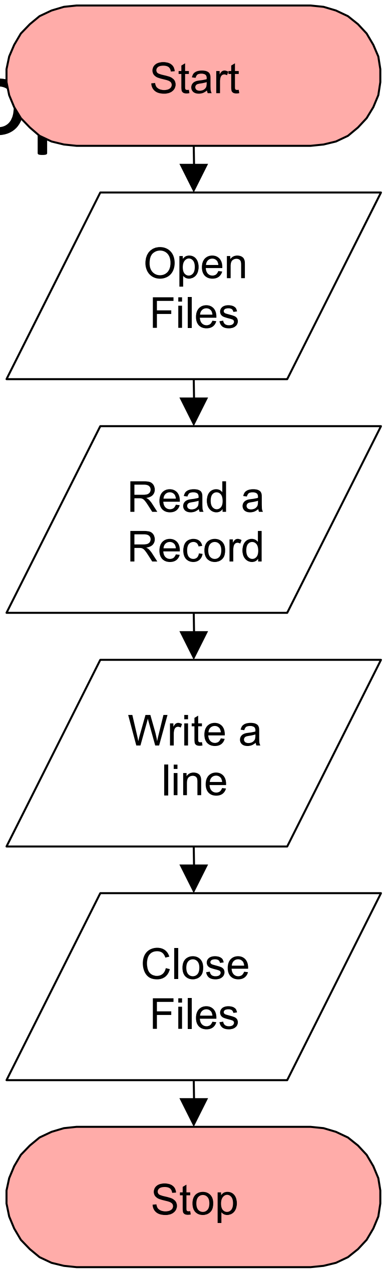
(page 34)

- Instructions
  - Open
  - Read
  - Move
  - Write
  - Close
  - Stop
- Areas in storage
  - Input areas
  - Output areas
  - Work areas

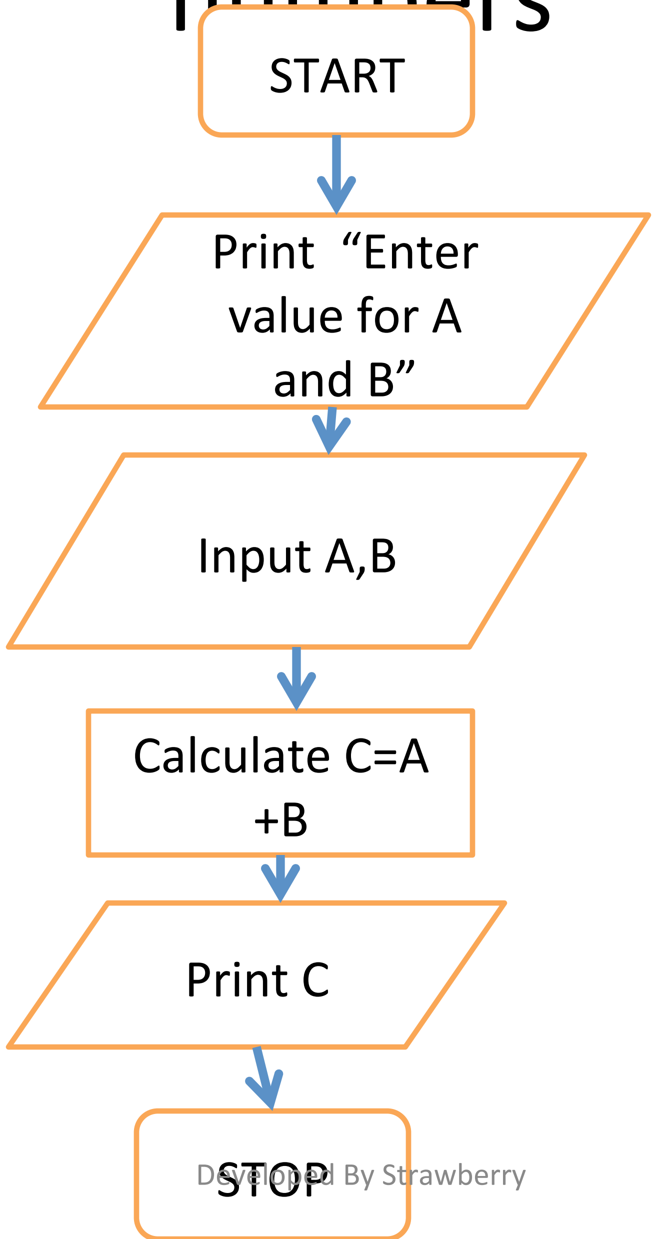


Start and Sto

- Start
  - First instruction of flowchart
- Stop
  - Last instruction executed
- Start and Stop instructions are in terminal symbols

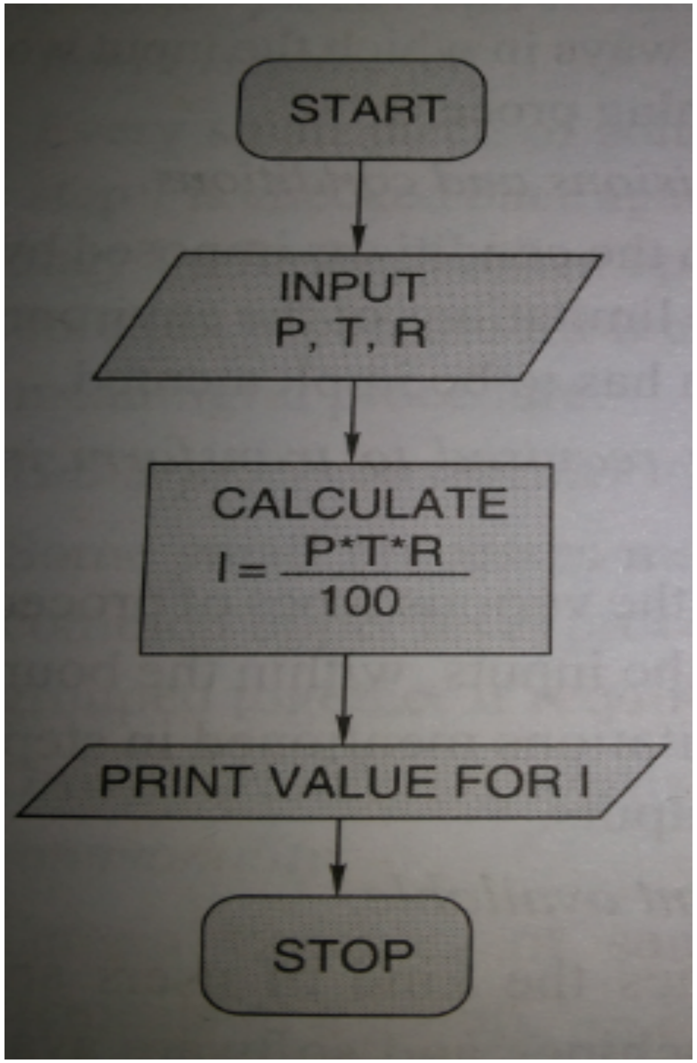


Flow chart example for adding two numbers



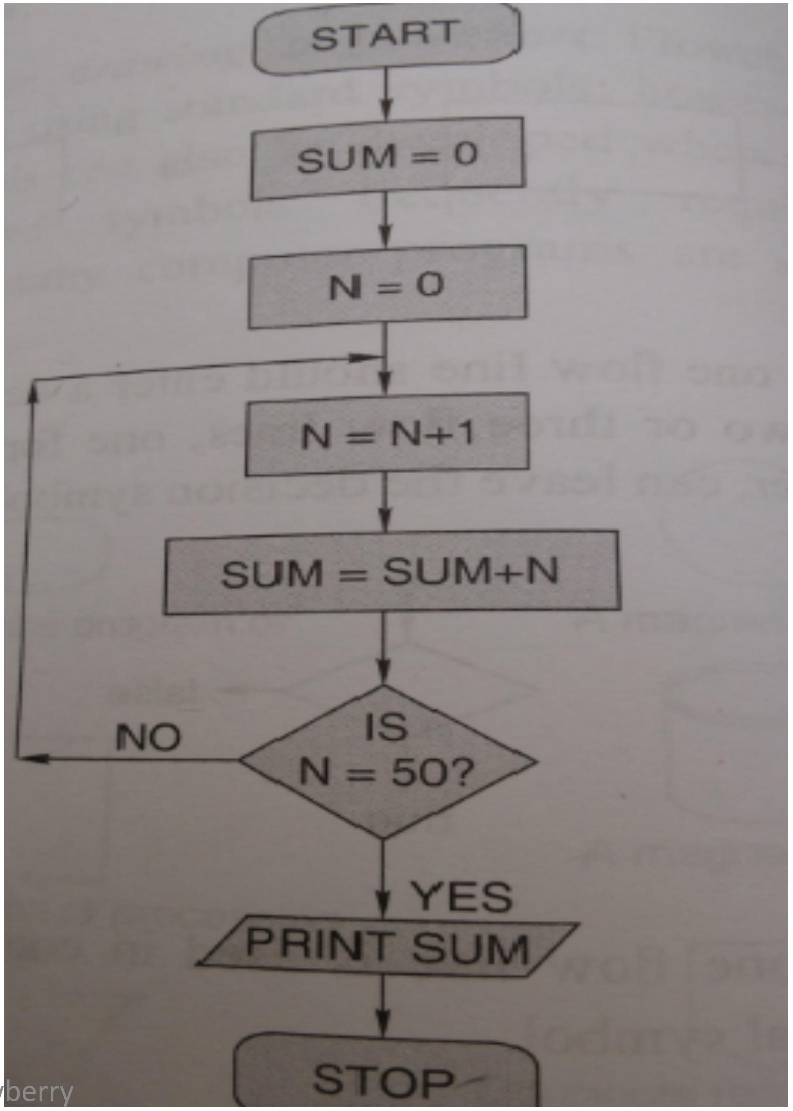
Flow chart Example

- Draw a flowchart for calculating the simple interest using the formula
- SI= (P*T*R)/100
- Where P denotes the principal amount, T time and R rate of interest.



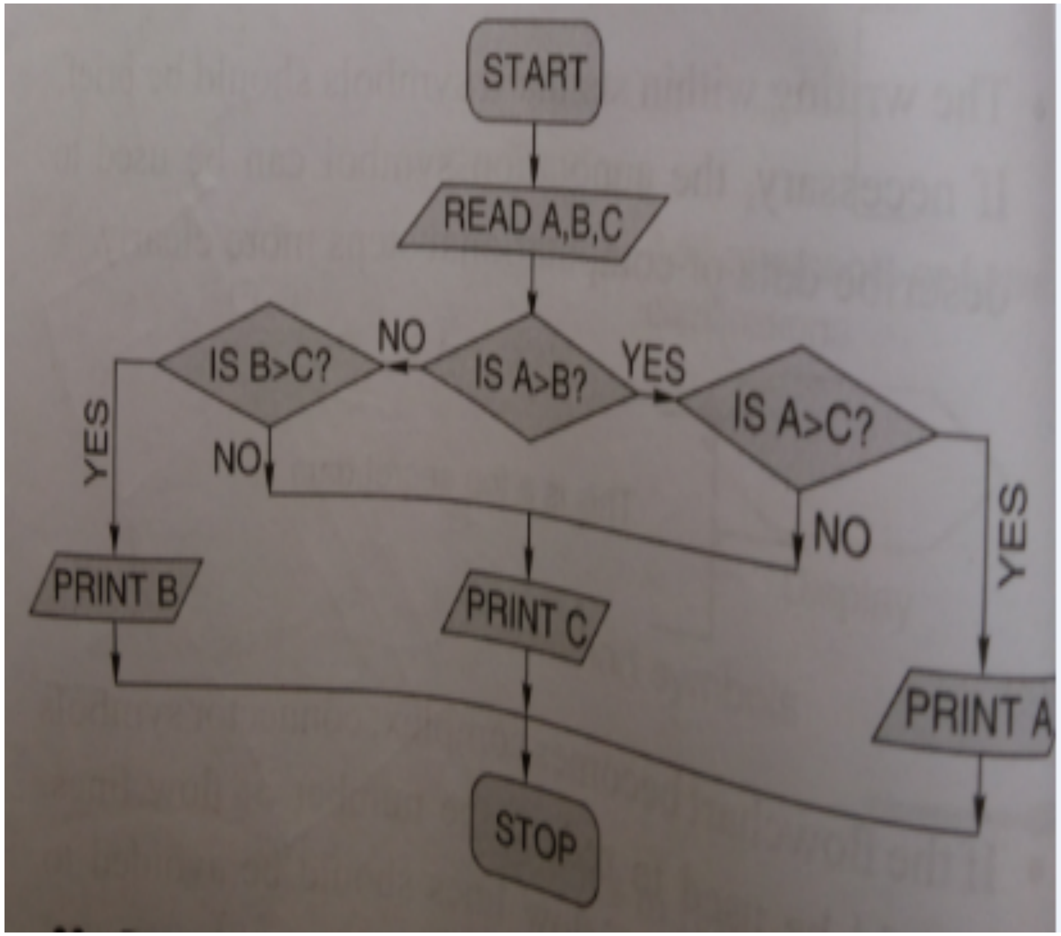
Flow chart Example

- Draw a flowchart to find the sum of the first 50 natural numbers.

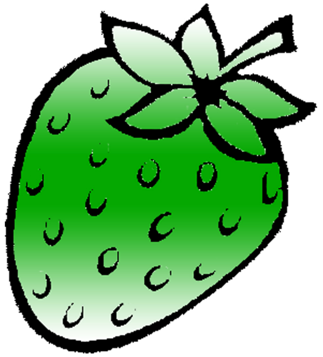


Flow chart Example

- Draw a flowchart to find the largest of three numbers A, B and C.



STRAWBERRY



/strawberrydevelopers

/strawberry_app

For more visit:

Strawberrydevelopers.weebly.com