

CHAPTER-5

ALGORITHM AND FLOWCHARTS

(অ্যালগরিদম এণ্ড ফ্লোচার্ট)

5.1. Algorithm [অ্যালগরিদম] : [NUH-08, NU(Pass)-04, NU(Pre)-04, DUH-11]

The most fundamental concept in computing is that of algorithm. It originates from the word "Algorism" that comes from the name of a famous mathematician Abu Ja'far Mohammed ibn Musa al-Khowarizmi (c. 825 A. D.) in 9th century. A sequence of instructions or a step-by step method or procedure solve a particular problem by computer is called algorithm. The steps may be (i) input, (ii) assignment, (iii) decision, (iv) repetition (or iteration), and (v) output. Algol is one kind of algorithmic language. There are many versions of Algol; such as Algol-58, Algol-60, Algol-68 etc. [অ্যালগরিদম হচ্ছে কম্পিউটিং-এর মৌলিক ধারণা। ইহা "Algorism" শব্দটি হতে উদ্ভাবিত, যা নবম শতাব্দীর আরবের বিখ্যাত গণিতবিদ আবুজাফর মোহাম্মদ ইবনে মুসা আল খোয়ারিজ্মি নামানুসারে নামকরণ করা হয়। কম্পিউটার দ্বারা কোন নির্দিষ্ট সমস্যা সমাধানের জন্য পর্যায়ক্রমিক নির্দেশের পুনরাবৃত্তি এবং (v) আউটপুট। অ্যালগল (Algol) একটি অ্যালগরিদম ভাষা। অ্যালগল -এর বিভিন্ন ভাস্তু রয়েছে। যেমন Algol-58, Algol-60, Algol-68 ইত্যাদি।]

The essential properties of a good algorithms are [ভাল অ্যালগরিদমের শুণাবলি হল]

1. It should be simple. [ইহা খুব সাধারণ হওয়া প্রয়োজন।]
2. The steps must be unambiguous. [ধাপসমূহ দ্ব্যর্থহীন হতে হবে।]
3. It must be effective in solving the problem. [ইহা অবশ্যই একটি সমস্যা সমাধানে কার্যকর হতে হবে।]
4. It should be universal and lead to a unique solution of the problem. [ইহা অবশ্যই সার্বজনীন হবে এবং একটি সমস্যায় একক (অনন্য) সমাধান দিতে সামর্থ হবে।]
5. It should have the capacity to handle some unexpected situations which may arise during the solution of a particular problem such as division by zero. [ইহাতে অবশ্যই অনাকাঙ্খিত পরিস্থিতি সামাল দেওয়ার মত ব্যবস্থা থাকতে হবে। যেমন কোন কিছুকে যেন শূন্য দ্বারা ভাগ না হয়।]

5.2. Flowchart (প্রবাহচিত্র) : [NUH-08, 12, NU(Pass)-07, 16, NU(Pre)-03, DUH-11]

A flowchart is a graphical representation of a specific sequence of steps of an algorithm. John Von Neumann made the first formal flowchart in 1945. Various symbols or boxes are used in the flowchart. Different types of operations are indicated by the following different shaped boxes [ফ্লোচার্ট হল অ্যালগরিদমের পর্যায়ক্রমিক ধাপসমূহের চিত্রিক্ষিত উপস্থাপনা। John Von Neumann ১৯৪৫ সালে সর্বোপ্রথম ফর্মাল ফ্লোচার্ট তৈরী করেন। বিভিন্ন ধরণের প্রতীক বা বক্স সমূহ ফ্লোচার্টের জন্য ব্যবহৃত হয়। বিভিন্ন ধরণের কাজ নির্দেশক বিভিন্ন আকৃতির বক্সসমূহ নিম্নে প্রদত্ত হল।]

SL NO ক্রমিক নং	Symbols [প্রতীক]	Name of symbols [প্রতীকের নাম]	Functions [কাজ]
1.		Ellipse or oval [উপস্থাপনা]	For use to start, stop, exit or pause of a flowchart. [প্রবাহচিত্রের শুরু, শেষ, বের হওয়া বা সাময়িক থামা বুরানোর জন্য ব্যবহৃত হয়।]
2.		Arrow [তীরচিহ্ন] or, Flow lines	For a direction [প্রবাহচিত্রের দিক নির্দেশনার জন্য ব্যবহৃত হয়।]

3.		Card [কার্ড]	For input or Read statement [Input বা Read টেটমেন্ট বুক্ষায়।]
4.		Rectangle [আয়তক্ষেত্র]	For use to calculate or process of data. ডাটা ক্যালকুলেশন বা প্রক্রিয়াকরণ করার জন্য ব্যবহৃত হয়।
5.		Paper cutting	For output print or write statement [Output, print বা write টেটমেন্ট বুক্ষায়।]
6.		Parallelogram [সামন্তরিক]	An input or output operation [ইনপুট বা আউটপুট অপারেশনে ব্যবহৃত হয়।]
7.		Diamond or, Rhombus [রম্বস]	For use to accept a decision by condition i. e. use for IF statement or select case statement. [শর্ত সাপেক্ষে সিদ্ধান্ত গ্রহণের কাজে ব্যবহৃত হয় অর্থাৎ, IF statement বা select case statement এ ব্যবহৃত হয়।]
8.		Hexagon [ষড়ভুজ]	Use for any loop structure, like as DO loop, FOR loop etc. [যেকোন লুপ কাঠামোতে ব্যবহৃত হয়। যেমন DO loop, FOR loop ইত্যাদি।]
9.		Small circle [ছোট বৃত্ত]	On page connector for a connection. [একই পেজে ক্যানেকশন দেওয়ার জন্য ক্যানেক্টর হিসাবে কাজ করে।]
10.		Pentagon [পঞ্চভুজ নীচে শীর্ষ]	Off page connector for a connection of a new page. [পৃষ্ঠা শেষে নতুন পৃষ্ঠায় যাওয়া বুক্ষায়।]
11.		Right open rectangle [ডান পার্শ্বে খোলা আয়তক্ষেত্র]	For use to write a comment. [মন্তব্য লিখার জন্য ব্যবহৃত হয়।]
12.			For use to process of data from subprogram. [সাবপ্রোগ্রাম হতে ডাটা প্রক্রিয়াকরণের জন্য ব্যবহৃত হয়।]
13.		পঞ্চভুজ বাম পার্শ্বে শীর্ষ	Only for use to display a result or results on the monitor. [শুধু ফলাফল মনিটরে প্রদর্শনের জন্য ব্যবহৃত হয়।]

5.3. Difference between Algorithm and Flowchart : The difference between Algorithm and Flowchart are given below [অ্যালগরিদম এবং ফ্লোচার্ট এর মধ্যে পার্থক্য নিম্নরূপ।

Algorithm [অ্যালগরিদম]	Flowchart [ফ্লোচার্ট]
1. Algorithm is the step by step procedure to solve a particular problem. [অ্যালগরিদম হলো কোন নির্দিষ্ট সমস্যা সমাধানের ধারাবাহিক বর্ণনা।]	1. Flowchart is the graphical representation of an Algorithm. [ফ্লোচার্ট বা প্রবাহ চিত্র হচ্ছে অ্যালগরিদমের চিত্ররূপ।]
2. No symbols or boxes are required to express Algorithm. [অ্যালগরিদম বর্ণনা করতে কোন প্রতীক বা বক্স ব্যবহারের প্রয়োজন নাই।]	2. Symbols or boxes are required to express flowchart. [ফ্লোচার্ট বর্ণনা করতে প্রতীক বা বক্স ব্যবহারের প্রয়োজন।]

3. Algorithm is the primary step of programming. [অ্যালগরিদম প্রোগ্রামিং এর প্রথম ধাপ।]

4. Algorithm depends on description. [অ্যালগরিদম বর্ণনার উপর নির্ভরশীল।]

5. Algorithm gives an idea to solve any problem. [অ্যালগরিদম থেকে সমস্যা সমাধানের ধারণা লাভ করা যায়।]

3. Flowchart is the secondary step of programming. [ফ্লোচার্ট প্রোগ্রামিং এর দ্বিতীয় ধাপ।]

4. Flowchart depends on graph. [ফ্লোচার্ট চিত্রের উপর নির্ভরশীল।]

5. Flowchart gives the way to write the correct program. [ফ্লোচার্ট সঠিক প্রোগ্রাম লিখার পথ দেখায়।]

5.4. Different structure of algorithm and flowchart :

[NU(Pre)-2004]

Flowchart is also an algorithm, which is the graphical representation of algorithm. In Fortran Language three types of structure are used in algorithm such as [Flowchart ও অ্যালগরিদম, যা অ্যালগরিদমের চিত্রভিত্তিক উপস্থাপনা। ফর্ট্রান ভাষায় তিনি ধরণের অ্যালগরিদম Structure ব্যবহার করা হয়। যেমন]

(i) Sequential structure (ii) Selective structure (iii) Repetitive structure

(i) **Sequential structure** : The sequential structure consists of one action followed by other [Sequential কাঠামোতে একটির পর একটি Statement কার্যকরী হয়।]

SOLVED PROBLEMS

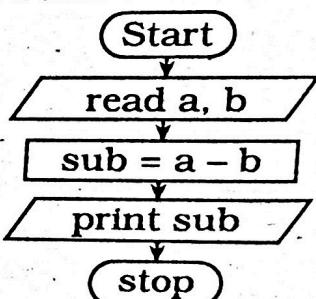
Problem-1. Subtract two real numbers [দুটি বাস্তব সংখ্যার বিয়োগ ফল নির্ণয়ের ক্ষেত্রে]

সংখ্যার বিয়োগ ফল নির্ণয়ের ক্ষেত্রে]

Solution :

Algorithm :

- | | |
|--------------------------|-----------------------------|
| (i) READ A, B | (i) A ও B এর মান পড় |
| (ii) COMPUTE SUB = A - B | (ii) SUB = A - B নির্ণয় কর |
| (iii) PRINT SUB | (iii) SUB এর মান ছাপাও |
| (iv) STOP | (vi) থাম। |



(ii) **Selective structure** : The selective structure consists of a test for a condition followed by two or more alternative paths for the program to flow. The program selects one of the paths depending on the test of the condition in true or false. [Selective কাঠামোতে একটি শর্ত পরীক্ষা করে দুই বা ততোধিক বিকল্পের মধ্যে থেকে শর্ত সত্য বা, মিথ্যার উপর ভিত্তি করে একটি বিকল্প কার্যকরী হয়।]

Problem-2. Find the difference between two real numbers [দুটি বাস্তব সংখ্যার পার্থক্য বের কর।]

Solution :

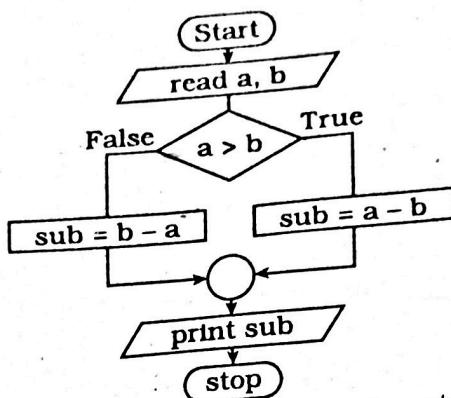
Algorithm :

- | | |
|-------------------------|----------------------------|
| (i) READ A, B | (i) A, B এর মান পড় |
| (ii) IF A > B THEN | (ii) যদি A > B হয় তবে |
| (a) COMPUTE SUB = A - B | (a) SUB = A - B নির্ণয় কর |

Algorithm and Flowcharts-5

- (b) ELSE
 - (c) COMPUTE SUB = B - A
 - (d) END IF
 - (iii) PRINT SUB
 - (iv) STOP
- Flowchart :**

- (b) নতুন
।
- (c) SUB = B - A নির্ণয় কর ।
- (d) যদি শেষ
- (iii) SUB এর মান ছাপাও,
- (iv) থাম ।



(iii) Repetitive Structure : In the repetitive structure, a statement or a set of statements is repeated as long as some condition is satisfied [Repetitive কাঠামোতে একটি শর্ত সিদ্ধ হওয়া সাপেক্ষে একটি বা একগুচ্ছ Statements বারবার কার্যকরী হয় ।]

Problem 3: Find the sum of the series $1 + 3 + 5 + \dots + (2n - 1)$

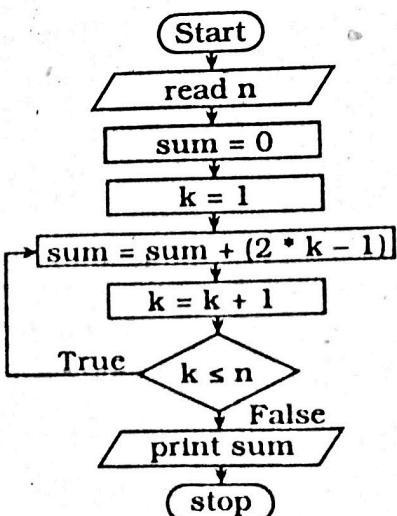
Solution :

Algorithm :

- (i) READ N
- (ii) LET SUM = 0
- (iii) LET K = 1
- (iv) COMPUTE SUM = SUM + (2 * K - 1)
- (v) COMPUTE K = K + 1
- (vi) IF K ≤ N GO TO STEP (IV) *(উৎপর্য)*
- (vii) PRINT SUM
- (viii) STOP

Flowchart :

- (i) N এর মান পড়
- (ii) SUM = 0 ধর
- (iii) K = 1 ধর
- (iv) SUM = SUM + (2 * K - 1) নির্ণয় কর
- (v) K = K + 1 নির্ণয় কর
- (vi) যদি K ≤ N হয় তবে (iv) ধাপে যাও
- (vii) SUM এর মান ছাপাও
- (viii) থাম ।

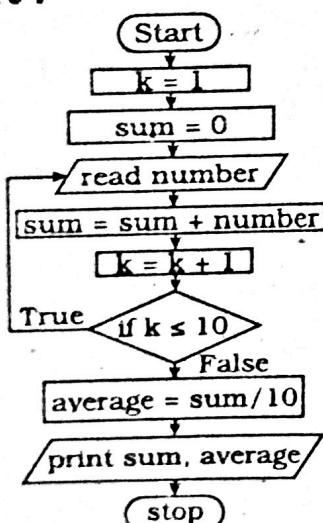


Problem-4. Write an algorithm to find the sum and average of 10 numbers draw a flowchart and implement it into Fortran. [দশটি সংখ্যার যোগফল ও গড় নির্ণয় কর এবং তা Fortran প্রোগ্রামে প্রয়োগ কর।]

Solution : Algorithm :

- (i) LET K = 1
- (ii) LET SUM = 0
- (iii) READ NUMBER
- (iv) CALCULATE SUM = SUM + NUMBER
- (v) CALCULATE K = K + 1
- (vi) IF K ≤ 10 GO TO STEP (III)
- (vii) CALCULATE AVERAGE = SUM / 10
- (viii) PRINT THE VALUE OF SUM
- (ix) PRINT THE VALUE OF AVERAGE
- (x) STOP

Flowchart :



- (i) K = 1 ধর
- (ii) SUM = 0 ধর
- (iii) NUMBER পড়
- (iv) SUM = SUM + NUMBER কর
- (v) K = K + 1 নির্ণয় কর
- (vi) যদি K ≤ 10 হয় তবে (iii) ধাপে যাও
- (vii) AVERAGE = SUM / 10 নির্ণয় কর
- (viii) SUM এর মান ছাপাও
- (ix) AVERAGE এর মান ছাপাও
- (x) থাম।

Fortran program :

```

REAL NUMBER
K = 1
SUM = 0.0
7 READ*, NUMBER
SUM = SUM + NUMBER
K = K + 1
IF(K.LE. 10) GOTO 7
AVERAGE = SUM / 10.0
PRINT*, SUM, AVERAGE
STOP
END

```

Problem-5. Write an algorithm and draw a flowchart to find the sum of the following series and implement it into Fortran : $1 + 2 + 3 + \dots + 100$ [$1 + 2 + 3 + \dots + 100$ ধারাটির যোগফল নির্ণয়ের জন্য অ্যালগরিদম ও ফ্লোচার্ট তৈরী কর এবং তা Fortran প্রোগ্রামে প্রয়োগ কর।]

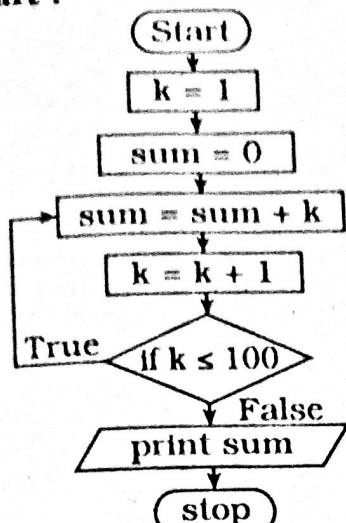
Solution :

Algorithm :

- (i) LET K = 1
- (ii) LET SUM = 0
- (iii) CALCULATE SUM = SUM + K
- (iv) CALCULATE K = K + 1
- (v) IF K ≤ 100 GOTO STEP (iii)
- (vi) PRINT THE VALUE OF SUM
- (vii) STOP

- (i) K = 1 ধর
- (ii) SUM = 0 ধর
- (iii) SUM = SUM + K নির্ণয় কর
- (iv) K = K + 1 নির্ণয় কর
- (v) যদি K ≤ 100 হয় তবে (iii) ধাপে যাও
- (vi) SUM এর মান ছাপাও
- (vii) থাম।

[NU(Pre)-2005]

Flowchart :**Fortran program :**

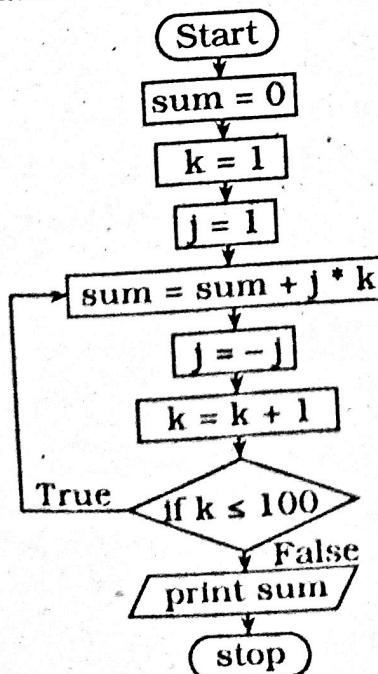
```

INTEGER SUM
K = 1
SUM = 0
10 SUM = SUM + K
      K = K + 1
      IF(K .LE. 100) GOTO 10
      PRINT*, 'SUM = ', SUM
      STOP
      END
  
```

Problem-6. Write an algorithm and draw a flowchart to find the sum of the following series and Implement it into Fortran: $1 - 2 + 3 - \dots - 100$ [$1 - 2 + 3 - \dots - 100$ ধারাটির যোগফল নির্ণয়ের জন্য একটি অ্যালগরিদম ও ফ্লোচার্ট তৈরী কর এবং তা Fortran প্রোগ্রামে প্রয়োগ কর।]

Solution :**Algorithm :**

- (i) LET SUM = 0
- (ii) LET K = 1
- (iii) LET J = 1
- (iv) CALCULATE SUM = SUM + J * K
- (v) Calculate j = -j
- (vi) Calculate k = k + 1
- (vii) If k ≤ 100 go to step (iv)
- (viii) print the value of sum
- (ix) Stop

Flowchart :

- (i) SUM = 0 ধর
- (ii) K = 1 ধর
- (iii) J = 1 ধর
- (iv) SUM = SUM + J * K নির্ণয় কর
- (v) J = -J নির্ণয় কর
- (vi) K = K + 1 নির্ণয় কর
- (vii) যদি K ≤ 100 হয় তবে (iv) ধাপে যাও
- (viii) SUM এর মান ছাপাও
- (ix) থাম।

Fortran program :

```

INTEGER SUM
SUM = 0
K = 1
J = 1
10 SUM = SUM + J * K
      J = -J
      K = K + 1
      IF(K .LE. 100) GOTO 10
      PRINT*, 'SUM = ', SUM
      STOP
      END
  
```

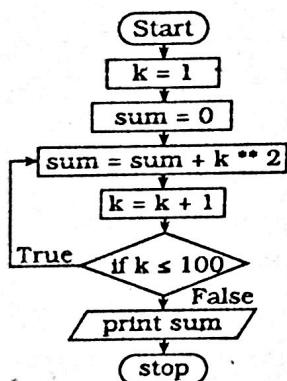
Problem-7. Write an algorithm and draw a flowchart to find the sum of following series and implement this into Fortran : $1^2 + 2^2 + 3^2 + \dots + 100^2$ [১০০ টি $1^2 + 2^2 + 3^2 + \dots + 100^2$ ধীরাটির যোগফল নির্ণয়ের জন্য একটি আলগোরিদম লিখ ও একটি শর্করা ফোর্ট্রেন প্রোগ্রাম প্রয়োগ কর]

Solution :

Algorithm :

- (i) LET SUM = 0
- (ii) LET K = 1
- (iii) CALCULATE SUM = SUM + K ** 2
- (iv) CALCULATE K = K + 1
- (v) IF K ≤ 100 GOTO STEP (iii)
- (vi) PRINT THE VALUE OF SUM
- (vii) STOP

Flowchart :



- (i) $\text{SUM} = 0$
- (ii) $K = 1$
- (iii) $\text{SUM} = \text{SUM} + K ** 2$
- (iv) $K = K + 1$
- (v) যদি $K \leq 100$ হয় তবে (iii) পথে পুনরাবৃত্তি
- (vi) SUM এর মান দ্বাপারে
- (vii) থাম !

Fortran program :

```

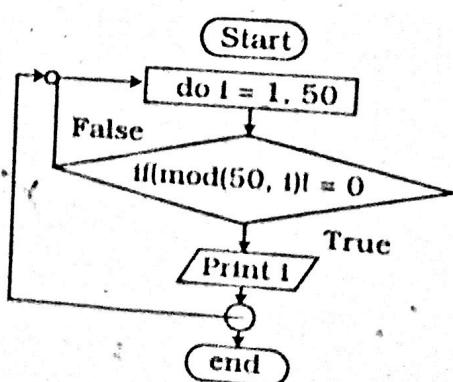
INTEGER SUM
K = 1
SUM = 0
5   SUM = SUM + K ** 2
      K = K + 1
      IF(K .LE. 100) GOTO 5
      PRINT*, 'SUM = ', SUM
      STOP
END
  
```

Problem-8. Draw a flowchart to print integers between 1 to 50 that are not divisible by 5. Then write a FORTRAN program according to the flowchart. [১ থেকে ৫০ পর্যন্ত যে সকল পূর্ণসংখ্যা ৫ দ্বারা বিভাজ্য নয় তাদেরকে প্রিন্ট করার প্রবাহ চির আঁক। প্রবাহ চির অনুযায়ী ফোর্ট্রেন প্রোগ্রাম লিখ]

```

DO I = 1, 50
IF(MOD(50, I) != 0
  PRINT*, I
ENDDO
END
  
```

[DUH (Aff. Coll.)-2017]



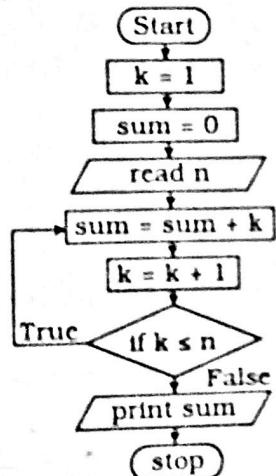
Problem-9. Write an algorithm and draw a flowchart to find the sum of the following series and implement this into Fortran : $1 + 2 + 3 + \dots + n$ [$1 + 2 + 3 + \dots + n$ ধৰণীর যোগফল নিৰ্ণয়ের জন্য একটি অ্যালগৱিদম লিখ ও একটি প্ৰবাহ চিৰ আৰু এবং তা Fortran প্ৰোগ্ৰামে প্ৰযোগ কৰ।]

Solution :

Algorithm :

- (i) LET SUM = 0
- (ii) LET K = 1
- (iii) READ N
- (iv) CALCULATE SUM = SUM + K
- (v) CALCULATE K = K + 1
- (vi) IF K ≤ N GO TO STEP (IV)
- (vii) PRINT THE VALUE OF SUM
- (viii) STOP

Flowchart :



- (i) SUM = 0 ধৰ
- (ii) K = 1 ধৰ
- (iii) N এৰ মান পড়
- (iv) SUM = SUM + K নিৰ্ণয় কৰ
- (v) K = K + 1 নিৰ্ণয় কৰ
- (vi) যদি K ≤ N হয় তবে (iv) ধাপে যাও
- (vii) SUM এৰ মান ছাপাও
- (viii) ধাৰ্ম।

Fortran program :

```

INTEGER SUM
K = 1
SUM = 0
READ*, N
7   SUM = SUM + K
      K = K + 1
      IF(K.LE. N) GOTO 7
      PRINT*, 'SUM = ', SUM
      STOP
      END
  
```

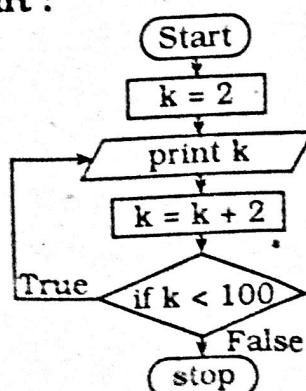
Problem-10. Write an algorithm and draw a flowchart to print the positive even numbers less than 100 and implement this into Fortran. [100 এৰ চেয়ে ছোট ধনাত্মক জোড় সংখ্যাগুলি ছাপানোৰ জন্য একটি অ্যালগৱিদম লিখ ও একটি প্ৰবাহ চিৰ আৰু এবং তা Fortran প্ৰোগ্ৰামে প্ৰযোগ কৰ।]

Solution :

Algorithm :

- (i) LET K = 2
- (ii) PRINT THE VALUE OF K
- (iii) Calculate k = k + 2
- (iv) IF K < 100 GOTO STEP (ii)
- (v) STOP

Flowchart :



- (i) K = 2 ধৰ
- (ii) K এৰ মান ছাপাও
- (iii) K = K + 2 নিৰ্ণয় কৰ
- (iv) যদি K < 100 হয় তবে (ii) ধাপে যাও
- (v) ধাৰ্ম।

Fortran program :

```

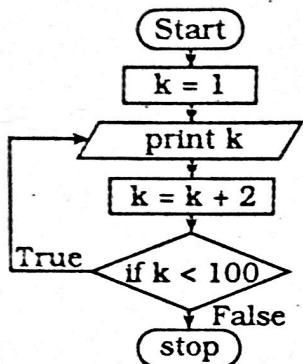
K = 2
5   PRINT*, K
      K = K + 2
      IF(K.LT. 100) GOTO 5
      STOP
      END
  
```

Problem-11. Write an algorithm and draw a flowchart to print the positive odd numbers less than 100 and implement this into Fortran. [100 এর চেয়ে ছোট ধনাত্মক বিজড়িত সংখ্যাগুলি ছাপানোর জন্য একটি অ্যালগরিদম লিখ ও একটি প্রবাহ চিত্র আঁক এবং তা Fortran প্রোগ্রামে প্রয়োগ কর।]

Solution : Algorithm :

- (i) LET K = 1
- (ii) PRINT THE VALUE OF K
- (iii) CALCULATE K = K + 2
- (iv) IF K < 100 GOTO STEP (ii)
- (v) STOP

Flowchart :



- (i) K = 1 ধর
- (ii) K এর মান ছাপাও
- (iii) K = K + 2 নির্ণয় কর
- (iv) যদি K < 100 হয় তবে (ii) ধাপে যাও
- (v) থাম।

Fortran program :

```

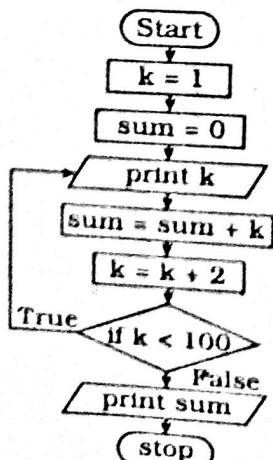
      K = 1
      5   PRINT*, K
           K = K + 2
           IF(K .LE. 100) GOTO 5
           STOP
           END
  
```

Problem-12. Write an algorithm and draw a flowchart to print the positive odd numbers less than 100 and to compute their sum. [100 এর চেয়ে ছোট ধনাত্মক বিজড়িত সংখ্যাগুলি ছাপানো ও তাদের যোগফল নির্ণয়ের জন্য একটি অ্যালগরিদম ও প্রবাহ চিত্র আঁক এবং তা Fortran প্রোগ্রামে প্রয়োগ কর।]

Solution : Algorithm :

- (i) LET K = 1
- (ii) LET SUM = 0
- (iii) PRINT THE VALUE OF K
- (iv) CALCULATE SUM = SUM + K
- (v) CALCULATE K = K + 2
- (vi) IF K < 100 GOTO STEP (iii)
- (vii) PRINT THE VALUE OF SUM
- (viii) STOP

Flowchart :



- (i) K = 1 ধর
- (ii) SUM = 0 ধর
- (iii) K এর মান ছাপাও
- (iv) SUM = SUM + K নির্ণয় কর
- (v) K = K + 2 নির্ণয় কর
- (vi) যদি K < 100 হয় তবে (iii) ধাপে যাও
- (vii) SUM এর মান ছাপাও
- (viii) থাম।

Fortran program :

```

      INTEGER SUM
      K = 1
      SUM = 0
      5   PRINT*, K
           SUM = SUM + K
           K = K + 2
           IF(K .LT. 100) GOTO 5
           PRINT*, SUM
           STOP
           END
  
```

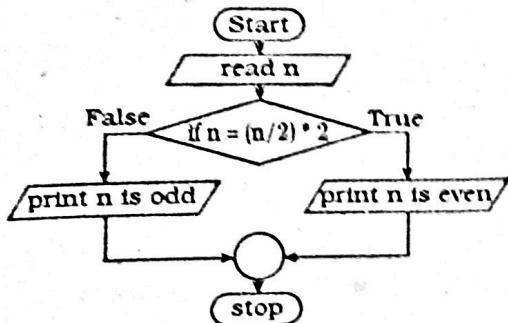
Problem-13. Write an algorithm and draw a flowchart to read an integer number N and check it is even or odd and implement this into Fortran. [একটি পূর্ণসংখ্যা N পড়ে এবং তা জোড় কি বিজ্ঞেড় তা নির্ণয়ের জন্য একটি আলগোরিদম সিখ ও একটি শুধুমাত্র চিত্র আৰু অপুৰ তা Fortran প্রয়োগ কৰ ।]

Solution :

Algorithm :

- (i) READ N
- (ii) IF $N = (N/2) * 2$ THEN
PRINT N 'IS EVEN'
ELSE
PRINT N 'IS ODD'
ENDIF
- (iii) STOP

Flowchart :



- (i) N পড়ু
- (ii) যদি $N = (N/2) * 2$ হয় তবে
ছাপাও N এর মান জোড়
নতুন
ছাপাও N এর মান বিজোড়
ধারণ
- (iii) ধারণ ।

Fortran program :

```

READ*, N
IF(N .EQ. (N/2)*2) THEN
  PRINT*, N, 'IS EVEN'
ELSE
  PRINT*, N, 'IS ODD'
ENDIF
STOP
END
  
```

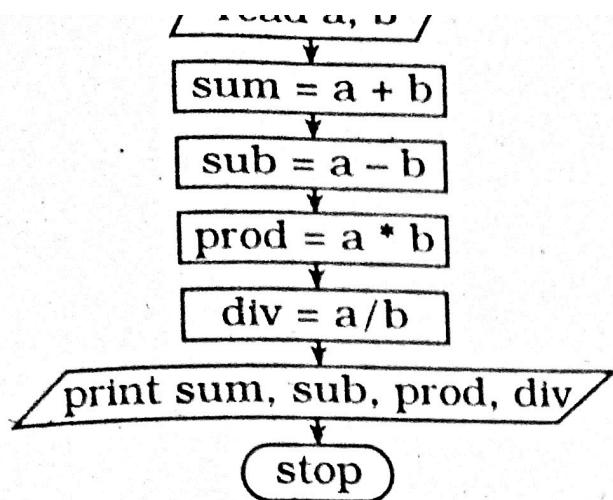
Problem-14. Write an algorithm and draw a flowchart to read an integer N and determine whether the integer number is positive, negative or zero. [একটি আলগোরিদম লিখ ও একটি প্রবাহ চিত্র আৰু যা একটি পূর্ণসংখ্যা N পড়বে এবং তা ধনাত্মক, ঋণাত্মক কি শূন্য তা নির্ণয় কৰবে ।]

Solution :

Algorithm :

- (i) READ N
- (ii) (a) IF $N > 0$ THEN
(b) PRINT N IS POSITIVE
(c) ELSE
(d) IF $N = 0$ THEN
(e) PRINT N IS ZERO
(f) ELSE
(g) PRINT N IS NEGATIVE
(h) END IF
(i) END IF
- (iii) STOP

- Or,**
- (i) READ N
 - (ii) (a) IF $N > 0$ THEN PRINT N IS POSITIVE
(b) IF $N < 0$ THEN PRINT N IS NEGATIVE
(c) IF $N = 0$ THEN PRINT N IS ZERO
 - (iii) STOP



SUB = A - B
 PROD = A * B
 DIV = A/B
 PRINT*, SUM, SUB, PROD, DIV
 STOP
 END

Example-18. Write an algorithm and draw a flowchart to find the sum of the following series and implement this into Fortran : $1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \dots + \frac{x^{10}}{10!} \cdot \left[1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \dots + \frac{x^{10}}{10!} \right]$

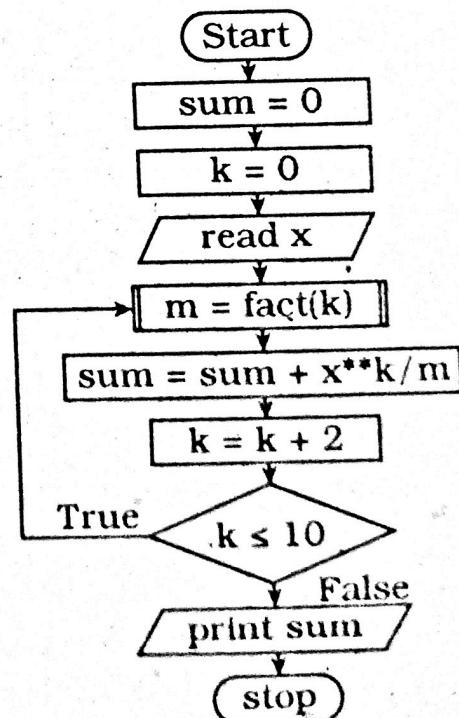
+ $\frac{x^{10}}{10!}$ ধৰাটিৰ যোগফল নিৰ্ণয়েৰ জন্য/একটি অ্যালগৱিদম লিখ ও একটি প্ৰণালী আবশ্যিক এবং তা Fortran প্ৰোগ্ৰাম লিখতে প্ৰয়োগ কৰ।]

Solution :

Algorithm :

- (i) LET SUM = 0
- (ii) LET K = 0
- (iii) READ X
- (iv) M = FACT(K) [CALCULATE FROM FUNCTION SUB-PROGRAM]
- (v) CALCULATE SUM = SUM + X ** K / FLOAT(M)
- (vi) CALCULATE K = K + 2
- (vii) IF K ≤ 10 GOTO STEP (IV)
- (viii) PRINT THE VALUE OF SUM
- (ix) STOP
- (x) DEFINE FUNCTION FACT (N)
- (xi) LET FACT = 1
- (xii) IF N = 0 THEN GO TO STEP (XVII)
- (xiii) LET J = 1
- (xiv) CALCULATE FACT = FACT * J
- (xv) CALCULATE J = J + 1
- (xvi) IF J ≤ N GO TO STEP (XIV)
- (xvii) RETURN

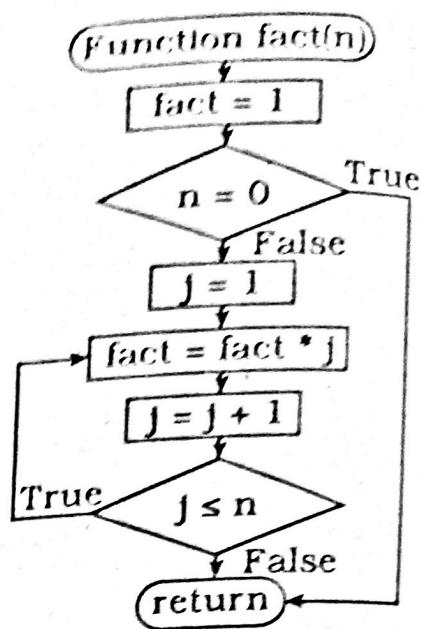
Flowchart :



Fortran program :

```

    INTEGER FACT
    SUM = 0.0
    K = 0
    READ*, X
    10 M = FACT(K)
    SUM = SUM + X**K/FLOAT(M)
    K = K + 2
    IF(K.LE.10) GOTO 10
    PRINT*, SUM
    END
    INTEGER FUNCTION FACT(N)
    FACT = 1
    IF(N.EQ.0) GOTO 30
    J = 1
    20 FACT = FACT * J
    J = J + 1
    IF(J.LE.N) GOTO 20
    30 RETURN
    END
  
```



Example-19. Write an algorithm and draw a flowchart to find the sum of the following series and implement this into Fortran: $x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots - \frac{x^9}{9!} \left[x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots - \frac{x^9}{9!} \right]$

$\frac{x^9}{9!}$ দ্বারা টির যোগফল নির্ণয়ের জন্য একটি অ্যালগরিদম লিখ ও একটি প্রবাহ চির আঁক এবং তা Fortran প্রেরণ

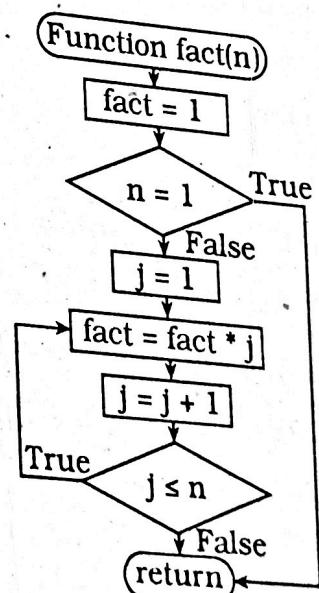
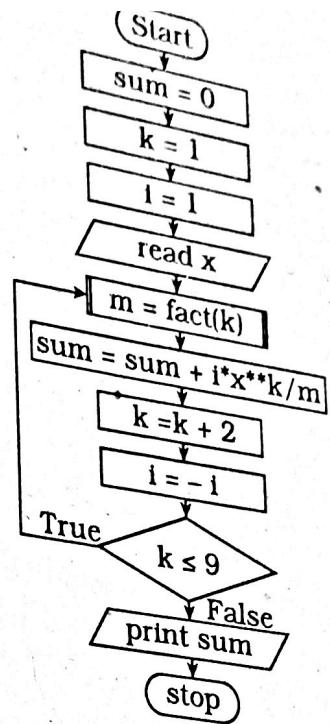
পদ্ধতি অযোগ কর।]

Solution :

Algorithm :

- (i) LET SUM = 0
- (ii) LET K = 1
- (iii) LET I = L
- (iv) READ X
- (v) M = FACT(K) [CALCULATE FROM FUNCTION SUB PROGRAM]
- (vi) CALCULATE SUM = SUM + I * X ** K / FLOAT(M)
- (vii) CALCULATE K = K + 2
- (viii) CALCULATE I = - I
- (ix) IF K ≤ 9 GO TO STEP (V)
- (x) PRINT THE VALUE OF SUM
- (xi) STOP
- (xii) DEFINE FUNCTION FACT(N)
- (xiii) LET FACT = 1
- (xiv) IF N = 1 THEN GO TO STEP (xix)
- (xv) LET J = 1
- (xvi) CALCULATE FACT = FACT * J
- (xvii) CALCULATE J = J + 1
- (xviii) IF J ≤ N GOTO STEP (xvi)
- (xix) RETURN

$m = \text{FACT}(k)$
 $\text{SUM} = \text{SUM} + i * (x * k / \text{FLOAT}(m))$
 $K = K + 2$
 $i = -i$
 $\text{IF}(K .LE. 9) \text{ GOTO } 10$
 $\text{PRINT} *, \text{SUM}$
 END
INTEGER FUNCTION FACT(N)
 $\text{FACT} = 1$
 $\text{IF}(N .EQ. 1) \text{ GOTO } 30$
 $J = 1$
20 $\text{FACT} = \text{FACT} * J$
 $J = J + 1$
 $\text{IF}(J .LE. N) \text{ GOTO } 20$
 RETURN
 END



Example-20. Write an algorithm and draw a flowchart to find the product of the following series and implement this into Fortran : $\frac{1}{2} \cdot \frac{3}{4} \cdot \frac{5}{6} \cdots \frac{21}{22}$ [$\frac{1}{2} \cdot \frac{3}{4} \cdot \frac{5}{6} \cdots \frac{21}{22}$] এর ফলাফল নির্ণয়ের জন্য একটি অ্যালগরিদম লিখ ও একটি প্রবাহ চিত্র আঁক এবং তা Fortran প্রোগ্রাম লিখতে প্রয়োগ কর।

Solution :

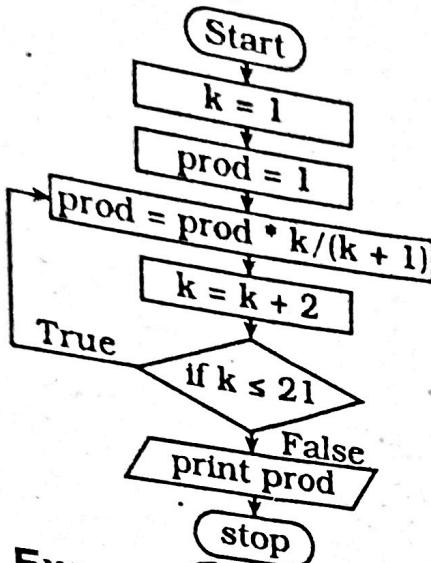
Algorithm :

- (i) LET $K = 1$
- (ii) LET $\text{PROD} = 1$
- (iii) CALCULATE $\text{PROD} = \text{PROD} * K / (K + 1)$
- (iv) CALCULATE $K = K + 2$

- | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> (i) $K = 1$ ধর (ii) $\text{PROD} = 1$ ধর (iii) $\text{PROD} = \text{PROD} * K / (K + 1)$ নির্ণয় কর (iv) $K = K + 2$ নির্ণয় কর |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

STOP THE VALUE OF PROD

Flowchart :



১৮ অক্টোবর (৩) পর্যন্ত

(vi) PROD এর মান ছাপাও
(vii) থাম।

Fortran program :

```

K = 1
PROD = 1.0
10 PROD = PROD * FLOAT(K) / (FLOAT(K)
K = K + 2
IF(K .LE. 21) GOTO 10
PRINT*, 'PRODUCT = ', PROD
STOP
END
  
```

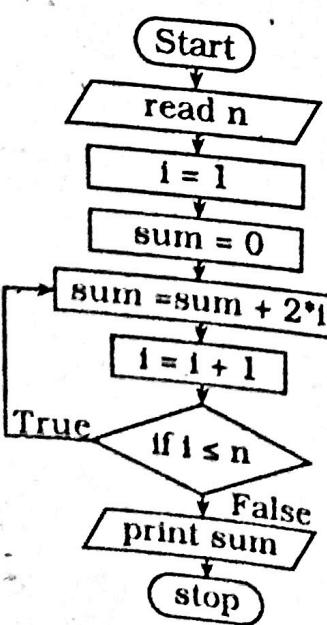
Example - 21. Given a positive integer N, obtain an algorithm and a flowchart to find the sum of the series : $2 + 4 + 6 + 8 + \dots + 2N$ [একটি অ্যালগরিদম লিখ ও প্রবাহ চির প্রকরণে ধনাত্মক পূর্ণসংখ্যা N পড়বে এবং $2 + 4 + 6 + 8 + \dots + 2N$ ধারাটির যোগফল নির্ণয় করবে।]

Solution :

Algorithm :

- (i) READ N
- (ii) LET SUM = 0
- (iii) LET I = 1
- (iv) CALCULATE SUM = SUM + 2 * I
- (v) LET I = I + 1
- (vi) IF I ≤ N GOTO STEP (IV)
- (vii) PRINT THE VALUE OF SUM
- (viii) STOP

Flowchart :



(i) N এর মান পড়ুন
(ii) SUM = 0 ধরুন
(iii) I = 1 ধরুন
(iv) SUM = SUM + 2 * I নির্ণয় করুন
(v) I = I + 1 নির্ণয় করুন
(vi) যদি I ≤ N হয় তবে (iv) ধরে যাও
(vii) SUM এর মান ছাপাও
(viii) থাম।

Fortran program :

```

INTEGER SUM
PRINT*, 'ENTER THE VALUE OF N'
READ(5, *) N
I = 1
SUM = 0
10 SUM = SUM + 2 * I
I = I + 1
IF(I .LE. N) GOTO 10
WRITE(6, 20) SUM
FORMAT(2X, 'SUM = ', I8)
STOP
  
```

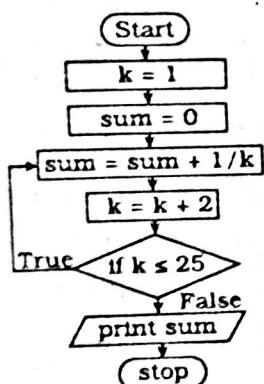
Example-22. Draw a flowchart and write a Fortran program which calculates the sum with four decimal places of the following series [নিচের দাবার যোগফল দের করার জন্য একটি ফ্লোচার্ট ও এক এবং সেই অনুযায়ী একটি প্রোগ্রাম লিখ যাহা চার দশমিক স্থান পর্যন্ত যোগফল দিবে।]

$$\frac{1}{1} + \frac{1}{3} + \frac{1}{5} + \dots + \frac{1}{25}$$

[NUH-2013]

Solution :**Algorithm :**

- (i) LET SUM = 0
- (ii) LET K = 1
- (iii) CALCULATE SUM = SUM + 1/K
- (iv) LET K = K + 2
- (v) IF K ≤ 25 GOTO STEP (III)
- (vi) PRINT THE VALUE OF SUM
- (vii) STOP

Flowchart :**Fortran program :**

```

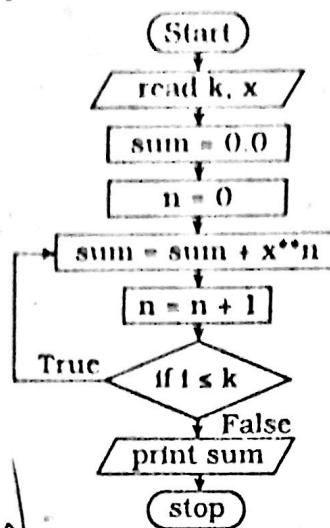
SUM = 0.0
K = 1
7  SUM = SUM + 1.0/FLOAT(K)
      K = K + 2
      IF(K .LE. 25) GOTO 7
      WRITE(6, 20) SUM
20  FORMAT(5X, 'SUM =' , F10.4)
      STOP
      END
  
```

Example-23. Construct a flowchart of the following series and implement it into Fortran : Sum = 1 + x + x² + ... + x^k. [Sum = 1 + x + x² + ... + x^k ধারাটির যোগফল নির্ণয়ের জন্য একটি অ্যালগরিদম লিখ ও একটি প্রবাহ চির আঁক এবং সে অনুযায়ী একটি প্রোগ্রাম লিখ ।] [NU(Pre)-2005]

Solution :**Algorithm :**

- (i) READ K AND X
- (ii) LET SUM = 0
- (iii) LET N = 0
- (iv) CALCULATE SUM = SUM + X * N
- (v) LET N = N + 1
- (vi) IF N ≤ K GOTO STEP (IV)
- (vii) PRINT THE VALUE OF SUM
- (viii) STOP

- (i) K এবং X এর মান পড়
- (ii) SUM = 0 ধর
- (iii) n = 0 ধর
- (iv) SUM = SUM + X * N নির্ণয় কর
- (v) N = N + 1 নির্ণয় কর
- (vi) যদি N ≤ K হয় তবে (iv) ধাপে যাও
- (vii) SUM এর মান ছাপাও
- (viii) থাম ।

Flowchart :**Fortran program :**

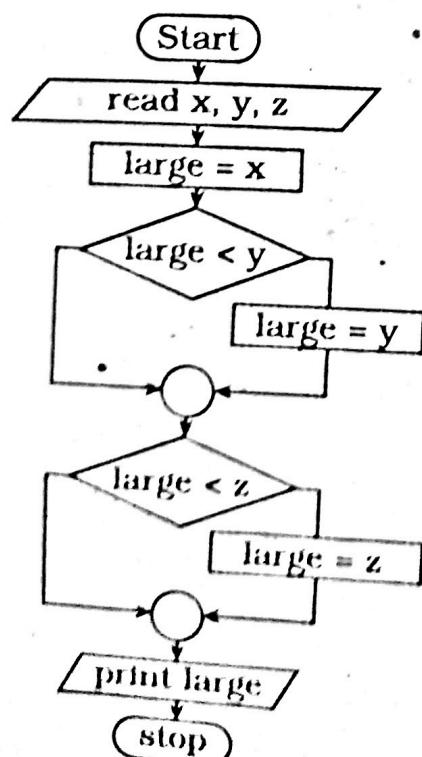
```

READ(5, *) K, X
SUM = 0.0
N = 0
9   SUM = SUM + X**N
      N = N + 1
      IF(N .LE. K) GOTO 9
      WRITE(6, 20) SUM
20   FORMAT(3X, 'SUM = ', P
      STOP
END
  
```

Example 24. Write a flowchart for choosing the largest of three digits numbers x, y and z. [ভিন্ন ভিন্ন তিনটি সংখ্যা x, y, z এর মধ্যে সবচেয়ে বড় সংখ্যাটি বাহির করার জন্য ফোর্ম্যাট তৈরী কর।] [NUH- 2010, 2012, 2016, NU(Pass)-2007, NU(Pre)- 2014, RUH- 2014]

Solution :**Algorithm :**

- (i) READ X, Y, Z
- (ii) LET LARGE = X
- (iii) IF LARGE < Y SET LARGE = Y
- (iv) IF LARGE < Z SET LARGE = Z
- (v) PRINT THE VALUE OF LARGE
- (vi) STOP

Flowchart :

(i) x, y, z এর মান পড়

(ii) large = x ধর

(iii) যদি large < y হয় তবে large = y ধর

(iv) যদি large < z হয় তবে large = z ধর

(v) large এর মান ছাপাও।

(vi) থাম।

Fortran program :

```

REAL LARGE
READ(5, *) X, Y, Z
LARGE = X
IF(LARGE .LT. Y) LARGE = Y
IF(LARGE .LT. Z) LARGE = Z
WRITE(6, 10) LARGE
10   FORMAT(2X, 'LARGEST NUMBER = ', FS.2)
      STOP
END
  
```

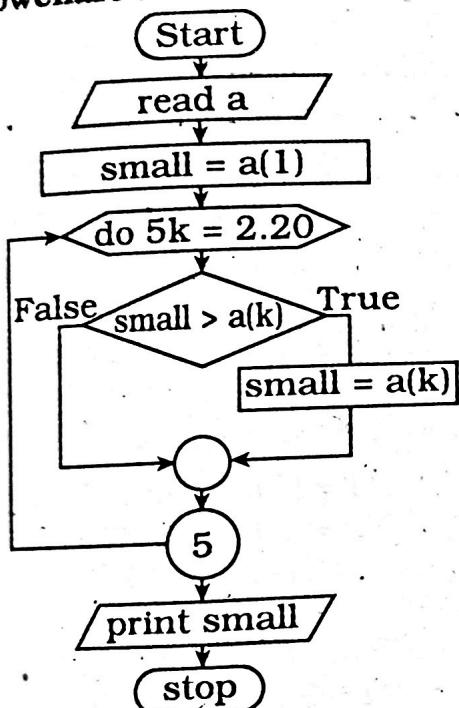
Example-25 Write a flowchart to find the smallest of a set of twenty numbers.
 ২০টি সংখ্যার একটি গুচ্ছের মধ্যে শুন্দিমতি বাহির করার জন্য একটি ফ্লোচার্ট আব।।

Solution :

Algorithm :

- Argu
(i) DEFINE ARRAY A(20)
(ii) READ A
(iii) LET SMALL = A(1)
(iv) DO FOR K = 2 TO 20
 (a) IF(SMALL > A(K)) SMALL = A(K)
 (b) ENDDO
(v) PRINT THE VALUE OF SMALL
(vi) STOP

(V) Flowchart :



[NUH-2003, NU(Pass)-2010]

- (i) A(20) ARRAY সংজ্ঞায়িত কর
 - (ii) A এর সকল মান পড়
 - (iii) SMALL = A(1) দ্বা
 - (iv) DO যখন K = 2 থেকে 20
 - (a) যদি SMALL > A(K) হয় তবে SMALL = A(K) দ্বা
 - (b) END DO
 - (v) SMALL এর মান ছাপাও
 - (vi) থাম।

Fortran program :

DIMENSION A(20)

READ *, (A(I), I = 1, 20)

SMALL = A(1)

DO 5 K = 2, 20

IF(SMALL .GT. A(K)) THEN

SMALL = A(K)

END IF

5 CONTINUE

WRITE(6, 10) SMALL

10 FORMAT(2X, 'SMALLEST NUMBER')

= ', F8.2)

END

stop
Example-26. Develop an algorithm and write a program in FORTRAN that reads values for the three sides of a triangle. Calculate its perimeter and its area and output these values. [একটি অ্যালগরিদম লিখ ও একটি প্রবাহ চিত্র আঁক যা একটি ত্রিভুজের তিনটি বাহুর মান পড়বে এবং তার পরিসীমা ও ক্ষেত্রফল নির্ণয় করবে। সে অনুযায়ী একটি FORTRAN প্রোগ্রাম লিখ।]

Solution :

Algorithm :

- (i) READ THE THREE SIDES OF THE TRIANGLE AND STORES IN A, B, C

(ii) CALCULATE $P = A + B + C$

(iii) CALCULATE $S = P/2$

(iv) AREA = $\text{SQRT}(S*(S - A)*(S - B)*((S - C)))$

(v) PRINT THE VALUE OF PERIMETER P

- (i) A, B, C এর মান পড়

(ii) $P = A + B + C$ নির্ণয় কর

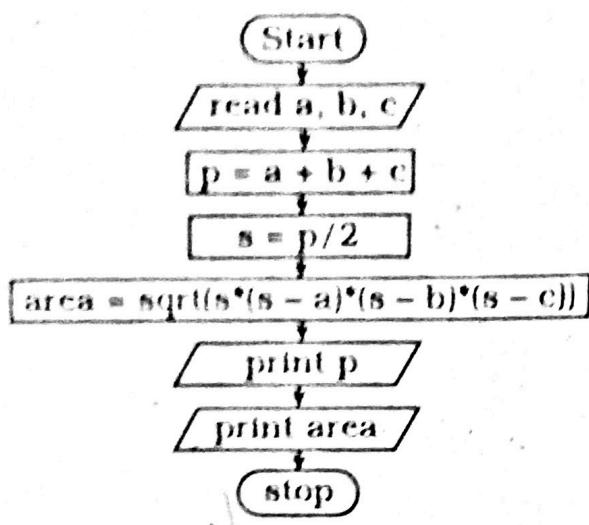
(iii) $S = P/2$ নির্ণয় কর

(iv) $\text{AREA} = \text{SQRT}(S*(S - A)*(S - B)*(S - C))$ নির্ণয় কর

(v) P এর মান ছাপাও

(vi) PRINT THE VALUE OF AREA

(vii) STOP

Flowchart :

Example-27. Write an algorithm and draw a flowchart to find all the roots of the quadratic equation $Ax^2 + Bx + C = 0$. [$Ax^2 + Bx + C = 0$ দিঘাত সমীকরণটি সকল মূল দেব চল্য একটি অ্যালগরিদম ও ফ্রোচার্ট তৈরী কর।] [NUH-08, 10, 11, 12, 17, NU(Pass)-09, 11 16, NU(Pre)-08, DUH-

Solution :**Fortran program :**

```

READ*, A, B, C
D = B**2 - 4.0*A*C
IF(D .GE. 0.0) THEN
  X1 = (-B + SQRT(D)) / (2.0*A)
  X2 = (-B - SQRT(D)) / (2.0*A)
  PRINT*, X1, X2
ELSE
  P = -B / (2.0*A)
  Q = SQRT(ABS(D)) / (2.0*A)
  PRINT*, '1ST ROOT = ', P, '+ I ', Q
  PRINT*, '2ND ROOT = ', P, '- I ', Q
ENDIF
STOP
END
  
```

Or,**Algorithm :**

- (i) READ A, B, C
- (ii) CALCULATE $D = B^2 - 4AC$
- (iii) CALCULATE $P = \frac{-b}{(2a)}$
- (iv) CALCULATE $Q = \sqrt{\frac{(ABS(D))}{(2A)}}$
- (v) IF $D \geq 0$ THEN

(vi) AREA এর মান হাপাও

(vii) গাম।

Fortran program :

PRINT*, 'ENTER THE SIDES OF THE TRIANGLE'

```

READ*, A, B, C
P = A + B + C
S = P/2.0
AREA = SQRT(S*(S-A)*(S-B)*(S-C))
WRITE(6, 10) P
10 FORMAT(2X, 'PERIMETER = ', F8.2)
WRITE(6, 20) AREA
20 FORMAT(2X, 'AREA = ', F8.2)
STOP
END
  
```

Or, COMPLEX CX1, CX2

```

READ*, A, B, C
D = B**2 - 4.0*A*C
IF(D .GE. 0.0) THEN
  X1 = (-B + SQRT(D)) / (2.0*A)
  X2 = (-B - SQRT(D)) / (2.0*A)
  PRINT*, X1, X2
ELSE
  P = -B/(2.0*A)
  Q = SQRT(ABS(D))/2.0*A
  CX1 = (P, Q)
  CX2 = (P, -Q)
  PRINT*, '1ST ROOT ', CX1
  PRINT*, '2ND ROOT ', CX2
ENDIF
STOP
END
  
```

অ্যালগরিদম :

- (i) A, B, C এর মান পড়
- (ii) $D = B^2 - 4AC$ নির্ণয় কর
- (iii) $P = -B/(2A)$ নির্ণয় কর
- (iv) $Q = \sqrt{ABS(D)/(2A)}$ নির্ণয় কর
- (v) যদি $D \geq 0$ হয় তবে

- (a) $X_1 = P + Q$
- (b) $X_2 = P - Q$
- (c) PRINT X_1, X_2
- (d) ELSE
- (e) PRINT $P, + I, Q$
- (f) PRINT $P, - I, Q$
- (g) ENDIF
- (vi) STOP

Program :

```

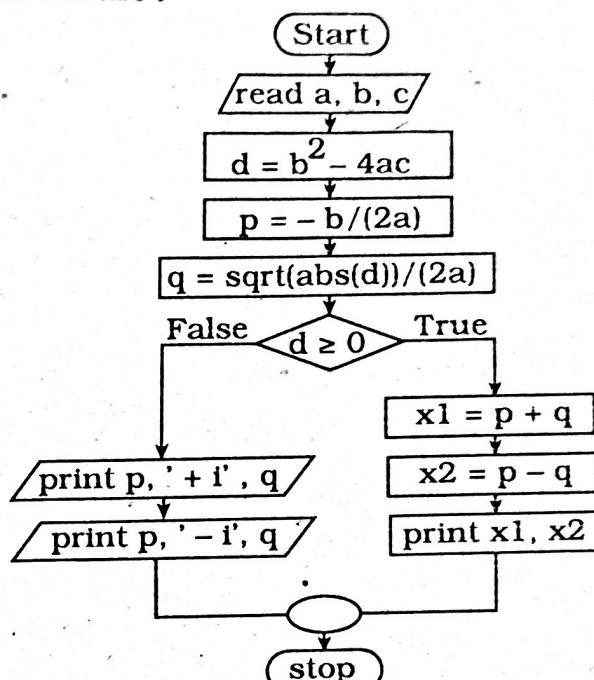
READ *, A, B, C
D = B ** 2 - 4.0 * A * C
P = - B / (2.0 * A)
Q = SQRT(ABS(D)) / (2.0 * A)
IF (D, GE, 0.0) THEN
    X1 = P + Q
    X2 = P - Q
    PRINT *, '1ST ROOT = ', X1
    PRINT *, '2ND ROOT = ', X2
ELSE
    PRINT *, '1ST ROOT = ', P, '+ I, Q
    PRINT *, '2ND ROOT = ', P, '- I, Q
ENDIF
END

```

- (a) $X_1 = P + Q$ নির্ণয় কর
- (b) $X_2 = P - Q$ নির্ণয় কর
- (c) X_1, X_2 ঢাপাও
- (d) ELSE (নতুবা)
- (e) $P, + I, Q$ ঢাপাও
- (f) $P, - I, Q$ ঢাপাও
- (g) ENDIF (যদি শেষ)

(vi) থাম।

Flowchart :



Example-28. Write an algorithm and draw a flowchart to find the real roots of a quadratic equation. [একটি দ্বিঘাত সমীকরণের বাস্তব মূল নির্ণয় করার জন্য একটি algorithm লিখ।]

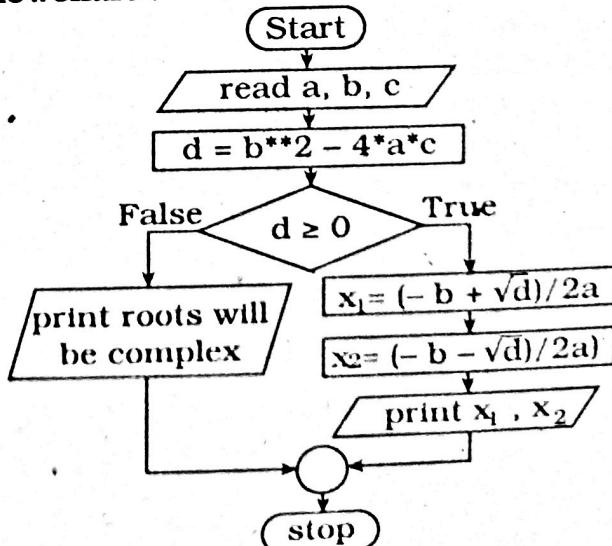
[NUH-2005, DUH-2004, 2009]

Solution :

Algorithm :

- (i) READ A, B, C
- (ii) CALCULATE $D = B^2 - 4AC$
- (iii) IF $D \geq 0$ THEN
 - (a) $X_1 = (-B + \sqrt{D}) / (2A)$
 - (b) $X_2 = (-B - \sqrt{D}) / (2A)$
 - (c) PRINT X_1, X_2
- (d) ELSE
 - (e) PRINT THE MESSAGE
 - (f) ROOTS WILL BE COMPLEX
- (g) END IF
- (iv) STOP

Flowchart :



Fortran program :

```

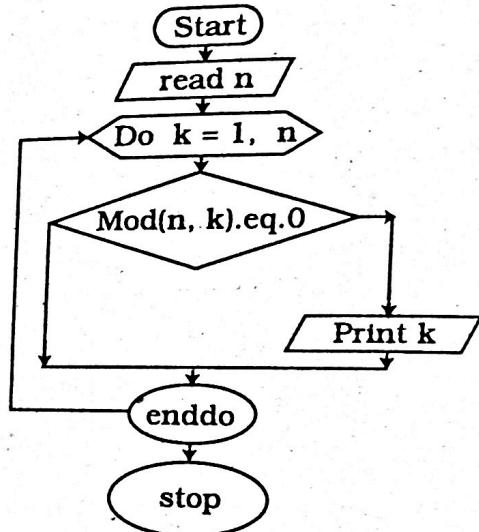
        READ*, A, B, C
        D = B**2 - 4.0*A*C
        IF(D .GE. 0.0) THEN
          X1 = (-B + SQRT(D))/(2.0*A)
          X2 = (-B - SQRT(D))/(2.0*A)
          PRINT20, X1, X2
          FORMAT(2X, 'X1 = ', F8.3, 2X, 'X2 = ', F8.3)
20      ELSE
          PRINT*, 'ROOTS WILL BE COMPLEX'
        ENDIF
        END
    
```

Example-29 Draw a flowchart to find the divisors of an integer number N
write a FORTRAN program according to the Flowchart. [একটি পূর্ণসংখ্যা N এর উণ্ঠিত
করার জন্য প্রবাহ চির্ত আঁক এবং প্রবাহ চির্ত অনুযায়ী ফোর্ট্রান প্রোগ্রাম লিখ।] [DUH (Aff. Coll.)-20]

Solution :

```

READ*, N
PRINT*, 'LIST OF DIVISORS OF', N
DO K = 1, N
  IF(MOD(N, K) .EQ. 0) PRINT*, K
ENDDO
END
    
```



Example-30 Draw a flowchart to print how many prime numbers between M and N (Not complete program) (একটি ফ্লোচার্ট আঁক যা M ও N (M < N) এর মধ্যবর্তী কতটি মৌলিক
সংখ্যা আছে তা নির্ণয় করবে।)

Solution :**Algorithm :**

- (i) LET P = 0
- (ii) READ M, N
- (iii) IF M = 1 THEN M = M + 1
- (iv) IF M = 2 THEN
 - (a) P = P + 1
 - (b) M = M + 1
 - (c) END IF
- (v) IF M > 2 THEN
 - (a) CALCULATE J = M / 2
 - (b) K = 2
 - (c) IF M = (M/K)K GOTO SUB-STEP (G)

- (i) P = 0 ধর
- (ii) M ও N এর মান পড়
- (iii) যদি M = 1 হয় তবে M = M + 1 নির্ণয় কর
- (iv) যদি M = 2 হয় তবে
 - (a) P = P + 1 নির্ণয় কর
 - (b) M = M + 1 নির্ণয় কর
 - (c) END IF
- (v) যদি M > 2 হয় তবে
 - (a) J = M/2 নির্ণয় কর
 - (b) K = 2 ধর
 - (c) যদি M = (M/K) * K হয় তবে (G) ধাপে ফি

- | | | | |
|-------|----------------------------|-------|--------------------------------|
| (d) | CALCULATE K = K + 1 | (d) | K = K + 1 নির্ণয় কর |
| (e) | IF K ≤ J GOTO SUB-STEP (C) | (e) | যদি K = J হয় তবে (C) ধাপে যাও |
| (f) | CALCULATE P = P + 1 | (f) | P = P + 1 নির্ণয় কর |
| (g) | CALCULATE M = M + 1 | (g) | M = M + 1 নির্ণয় কর |
| (h) | IF M ≤ N GOTO SUB-STEP (A) | (h) | যদি M ≤ N হয় তবে (A) ধাপে যাও |
| (vi) | PRINT THE VALUE OF P | (vi) | P এর মান ছাপাও |
| (vii) | STOP | (vii) | থাম। |

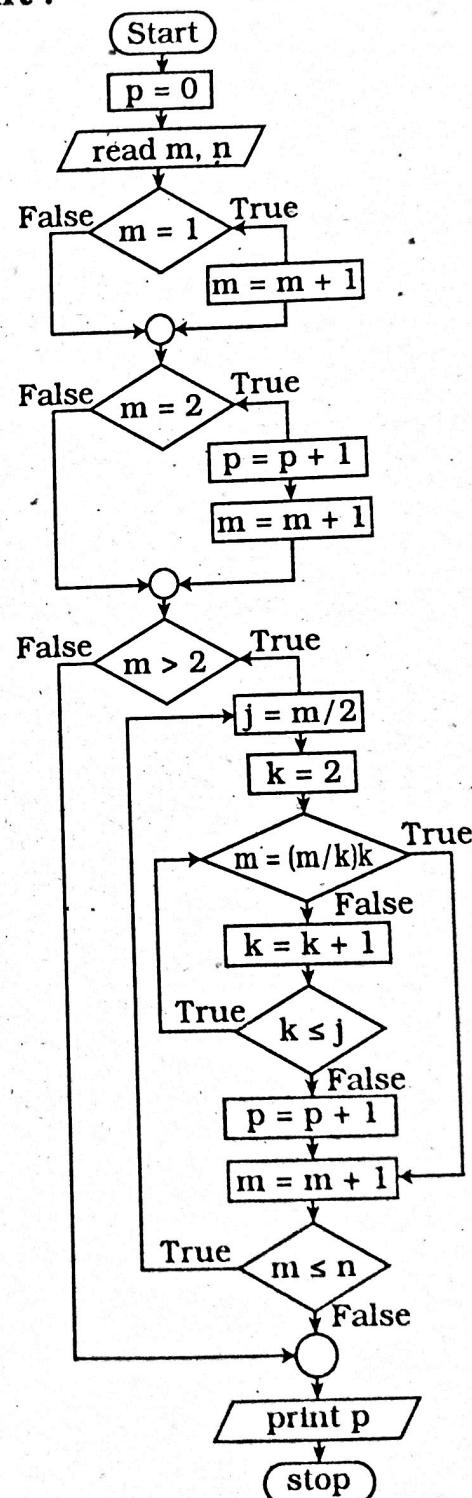
Fortran program :

```

INTEGER P
P = 0
READ(5, *) M, N
IF(M .EQ. 1) M =
IF(M .EQ. 2) THE
P = P + 1
M = M + 1
END IF
IF(M .GT. 2) THE
5 J = M / 2
K = 2
10 IF(M .EQ. (M/K))
    K = K + 1
    IF(K ≤ J) GOTO 1
    P = P + 1
20 M = M + 1
    IF(M .LE. N) GO TO 10
ENDIF
PRINT30, P
30 FORMAT(2X, 'N')
1 PRIME NUMBER
END

```

Flowchart :



Example-31. Write an algorithm and a FORTRAN program, which reads integer N and determines whether N is prime or not. [একটি অঙ্গ প্রোগ্রাম লিখ যা একটি পূর্ণ সংখ্যা N পড়বে এবং N মৌলিক কিনা তা নির্ণয় করবে।]

[NUH-13, 17, 18, NU(Pass)-11, 16, 18, DUH (Aff. Coll.)]

Solution :

Algorithm :

- (i) READ N
- (ii) IF $N \leq 1$ GOTO STEP (XI)
- (iii) IF $N = 2$ GOTO STEP (IX)
- (iv) CALCULATE $J = N / 2$
- (v) LET $K = 2$
- (vi) IF $N = (N/K) * K$ GOTO STEP (XI)
- (vii) CALCULATE $K = K + 1$
- (viii) IF $K \leq J$ GOTO STEP (VI)
- (ix) PRINT N IS A PRIME
- (x) GOTO STEP (XII)
- (xi) PRINT N IS NOT A PRIME
- (xii) STOP

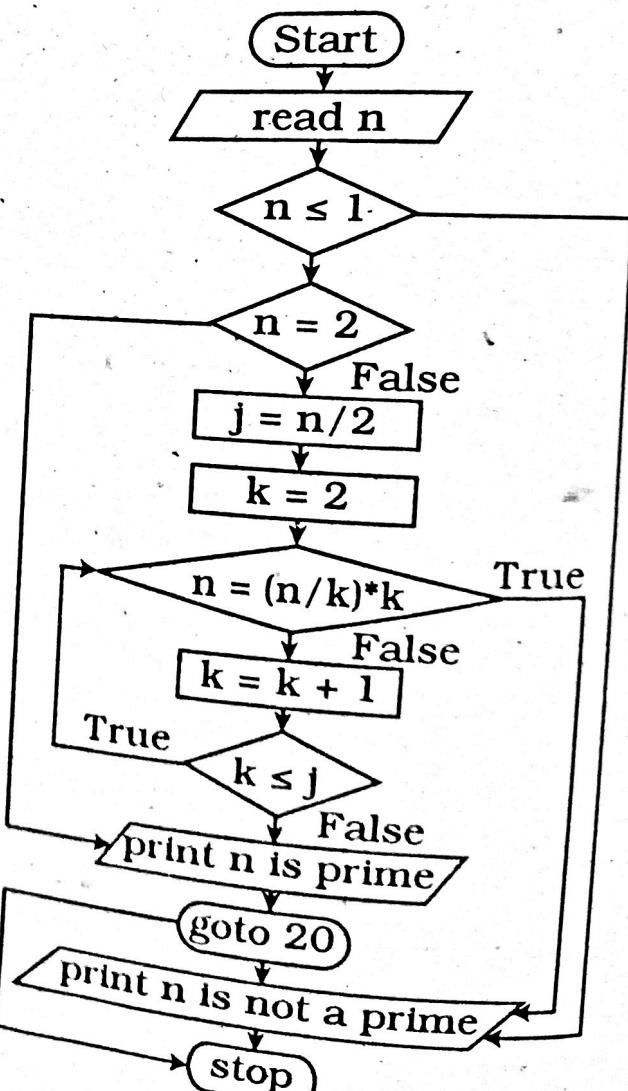
Fortran program :

```

READ*, N
IF(N .LE. 1) GOTO 10
IF(N .EQ. 2) GOTO 7 .
J = N / 2
K = 2
5   IF(N .EQ. (N/K) * K) GOTO 10
      K = K + 1
      IF(K .LE. J) GOTO 5
7   PRINT*, N, 'IS A PRIME'
      GOTO 20
10  PRINT*, N, 'IS NOT A PRIME'
20  STOP
END
  
```

- N এর মান পড়
- (i) যদি $N \leq 1$ হয় তবে (XI) ধাপে যাও
 - (ii) যদি $N = 2$ হয় তবে (IX) ধাপে যাও
 - (iii) $J = N/2$ নির্ণয় কর
 - (iv) $K = 2$ ধর
 - (v) যদি $N = (N/K) * K$ হয় তবে (XI) ধাপে যাও
 - (vi) $K = K + 1$ নির্ণয় কর।
 - (vii) যদি $K \leq J$ হয় তবে (VI) ধাপে যাও
 - (viii) ছাপাও N মৌলিক
 - (ix) (XII) ধাপে যাও
 - (x) ছাপাও N মৌলিক নয়
 - (xi) থাম।
 - (xii) থাম।

Flowchart :



Example 3. Write an algorithm to determine if an integer $N > 2$ is a prime number. If N is not a prime, print N and all its divisors. Implement this FORTRAN to find all prime numbers < 500 . [একটি পূর্ণসংখ্যা $N > 2$ মৌলিক কিনা তা নির্ণয় করে। একটি algorithm লিখ। যদি N মৌলিক না হয়, তবে N এর সকল ভাজক ছাপবে। 500 এর নীচে সকল সংখ্যা নির্ণয়ের জন্য উক্ত algorithm কে FORTRAN এ রূপান্তর করে।] [NUH-2004, 2]

Solution :

Algorithm :

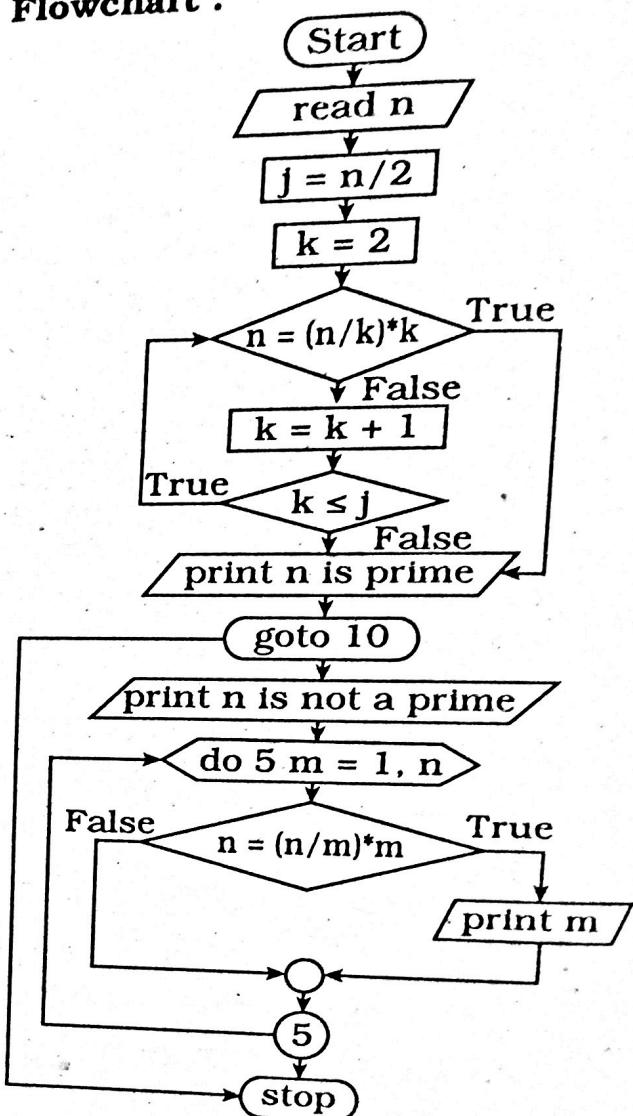
- (i) READ N
- (ii) LET J = N / 2
- (iii) LET K = 2
- (iv) IF $N = (N/K)*K$ GOTO STEP (IX)
- (v) LET K = K + 1
- (vi) IF $K \leq J$ GOTO STEP (IV)
- (vii) PRINT N IS A PRIME
- (viii) GOTO STEP (XI)
- (ix) PRINT N IS NOT A PRIME
- (x) DO FORM M = 1, N
 - (1) IF $N = (N/M)*M$ THEN PRINT M
 - (2) END DO
- (xi) STOP

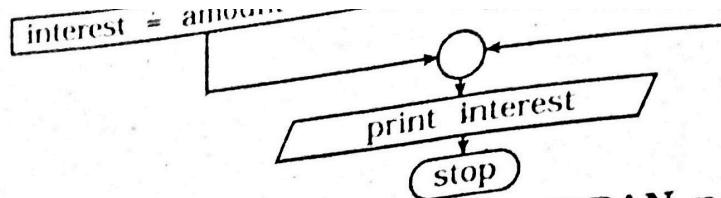
Fortran program :

```

I = 2
PRINT*, I
N = 3
7   J = N/2
     K = 2
10  IF(N .EQ. (N/K) * K) GOTO 20
     K = K + 1
     IF(K .LE. J) GOTO 10
     PRINT*, N
20  N = N + 2
     IF(N .LE. 500) GOTO 7
     STOP
END
  
```

Flowchart :





Example-36. Write an algorithm and a FORTRAN program to read a five digit number and to compute the sum of the digits of the number. [একটি অ্যালগরিদম লিখ, যা পাঁচ অংকবিশিষ্ট একটি সংখ্যা পড়বে এবং তার সংখ্যার অংকগুলোর যোগফল করবে।]

[NUH-2009, 2016, NU(Pre)]

Solution :

Algorithm :

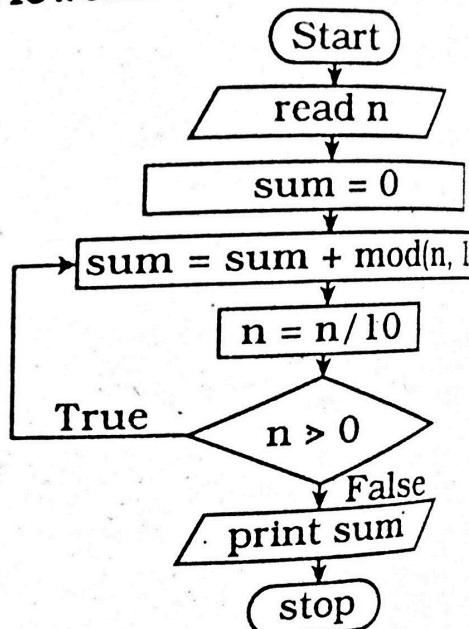
- (i) READ A FIVE DIGIT NUMBER N
- (ii) LET SUM = 0
- (iii) CALCULATE SUM = SUM + MOD (N, 10)
- (iv) CALCULATE N = N/10
- (v) IF N > 0 GOTO STEP (III)
- (vi) PRINT SUM
- (vii) STOP

Fortran program :

```

INTEGER SUM
READ*, N
SUM = 0
10 SUM = SUM + MOD(N, 10)
      N = N/10
      IF(N .GT. 0) GOTO 10
      WRITE(6, 20) SUM
20 FORMAT(2X, 'SUM = ', I5)
      END
  
```

Flowchart :



Example-37. Suppose A, B, C contains 3 positive numbers. Write an algorithm to determine whether A, B, C can form the sides of a triangle. If no, print the message 'NOT A TRIANGLE'; if yes, determine whether A, B, C form the sides of (i) an equilateral triangle, (ii) an isosceles triangle, (iii) a right-angled triangle. Implement this into FORTRAN. [মনে কর A, B, C তিনটি ধনাত্মক সংখ্যা ধারণ করে। A, B, C তিনজোর তিনবাহু হতে পারে কিনা, তা নির্ণয়ের জন্য একটি algorithm লিখ। যদি হয়, তবে A, B, C (i) সমবাহু ত্রিভুজ, (ii) সমদ্বিবাহু ত্রিভুজ, (iii) সমন্তব্ধ ত্রিভুজ।]

INTEGER SUM

```

PRINT*, ID      AVERAGE
DO 7 N = 1, 50
    READ*, ID, N1, N2, N3
    SUM = N1 + N2 + N3
    AVE = FLOAT(SUM) / 3.0
    PRINT*, ID, AVE

```

7 CONTINUE

STOP

END

Example-42 The temperature is found 5° C in a beating colded night at R. U. Write a FORTRAN program to find that temperature in a Fahrenheit scale using the relation $\frac{C}{5} = \frac{F - 32}{9}$. [কোন এক তীব্র শীতের রাতে R. U. তে তাপমাত্রা পাওয়া গেল 5° C এই তাপমাত্রাকে $\frac{C}{5} = \frac{F - 32}{9}$ সম্পর্ক ব্যবহার করে ফাহেনহাইট তাপমাত্রায় রূপান্তরের জন্য একটি অ্যালগরিদম ও পদার্থ চির আঁক করে সে অনুযায়ী একটি FORTRAN প্রোগ্রাম লিখ।]

[RUH(Stat)-2000]

Hence given a Flowchart for above FORTRAN program.

Solution :

Algorithm :

- (i) GIVEN C = 5° CELSIUS
- (ii) CALCULATE F = $\frac{9}{5} C + 32$
- (iii) WRITE THE VALUE OF F
- (iv) STOP

Fortran program :

```

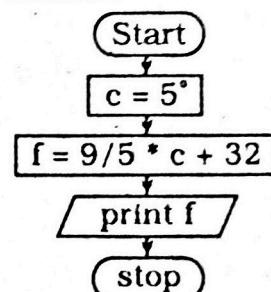
C = 5.0
F = (9.0/5.0) * C + 32.0
PRINT20, F
20 FORMAT(2X, 'FAHRENHEIT TEMPERA
TURE =', F5.2)
END

```

অ্যালগরিদম :

- (i) C = 5 ধর (দেওয়া আছে)
- (ii) F = $\frac{9}{5} * C + 32$ নির্ণয় কর
- (iii) F এর মান ছাপাও
- (iv) থাম

Flowchart :



Example-43. In a class test the scores out of 80 marks of 150 students are scored in a data file in the following format [একটি ক্লাস পরীক্ষায় 150 ছাত্রের মোট 80 নম্বরের মধ্য থেকে নম্বর একটি ডাটা ফাইলে নিম্নরূপে সন্নিবেশিত আছে।]

Roll NO.

Marks obtained

Draw a flowchart for making three lists of three following categories of students : [নিম্নলিখিত তিনটি ক্যাটাগরীভুক্ত তিনটি লিষ্ট তৈরীর জন্য একটি Flowchart আঁক।]

Category

Marks obtained range

1st

Marks ≥ 48

2nd

$32 \leq \text{Marks} < 48$

3rd

Marks < 32

Each list must contain the following information [প্রতি লিষ্টে নিম্নলিখিত তথ্য থাকতে হবে।]

Roll No.	Marks obtained	% of marks obtained
----------	----------------	---------------------

Write a Fortran program for the above problem. [উপরোক্ত সমস্যা সমাধানের একটি Fortran প্রোগ্রাম লিখ।]

Example-44. Write an algorithm and a FORTRAN Program, which prints all positive integers less than 100 omitting those integers divisible by 7. [একটি এলগরিদম এবং একটি FORTRAN প্রোগ্রাম লিখ, যা 100 অপেক্ষা ক্ষুদ্রতম সকল ধনাত্মক পূর্ণসংখ্যা ছাপবে যারা 7 দ্বারা বিভাজ্য নহ।]

Solution :

Algorithm :

- (i) LET K = 1
- (ii) IF MOD (K, 7) ≠ 0 THEN
PRINT THE VALUE OF K
ENDIF
- (iii) CALCULATE K = K + 1
- (iv) IF K < 100 GOTO STOP ((II))

আলগরিদম :

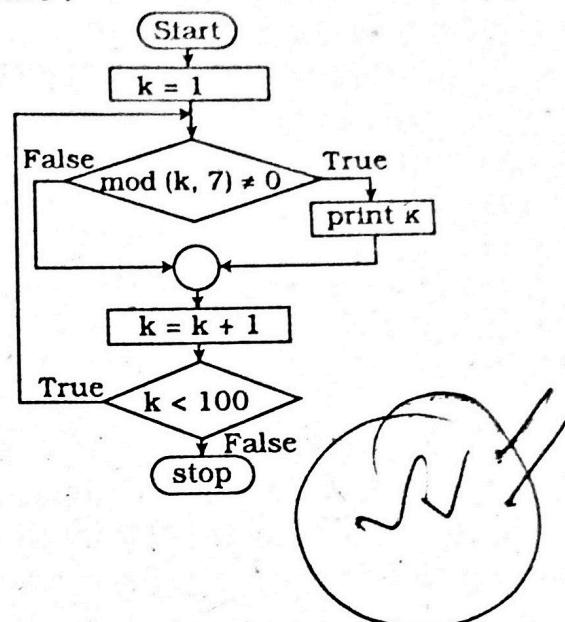
- (v) STOP
- (i) K = ১
- (ii) যদি $\text{MOD}(K, 7) \neq 0$ হয় তবে K এর মান ছাপাও END IF (যদি মেষ)
- (iii) K = K + 1 নির্ণয় কর
- (iv) যদি $K \leq 100$ হয় তবে (II) ধাপ যাও
- (v) থাম।

Fortran Program :

```

K=1
20 IF(MOD(K, 7) .NE. 0) THEN
    PRINT *, K
  ENDIF
  K=K+1
  IF(K .LT. 100) GOTO 20
  STOP
END
  
```

Flowchart :



Example-45. Write an algorithm and draw a flowchart to determine whether a given integer $K > 2$ is prime or not. Implement this into FORTRAN. [প্রদত্ত একটি পূর্ণসংখ্যা $K > 2$ কে পৰিকল্পিত কিনা তা নির্ণয়ের জন্য একটি algorithm লিখ এবং একটি ফ্লোচটি আঁক। এবং তা FORTRAN এ ট্রান্সলেট কর।]

Solution :

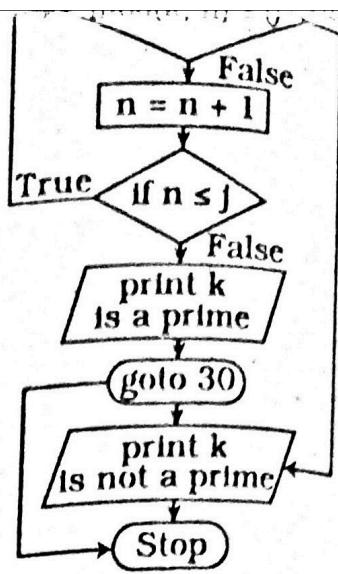
Algorithm :

- (i) READ K
- (ii) CALCULATE J = K/2
- (iii) LET N = 2
- (iv) IF MOD (K, N) = 0 GOTO STEP (IX)
- (v) CALCULATE N = N+1
- (vi) IF N ≤ J GOTO STOP (IV)
- (vii) PRINT K IS A PRIME

```

10   N = 2
      IF(MOD(K, N) .EQ. 0) GOTO 10
      N = N + 1
      IF(N .LE. J) GOTO 10
      PRINT *, K, ' IS A PRIME'
      GOTO 30
      PRINT *, K, ' IS NOT A PRIME'
20
30   STOP
      END

```



Example-46. Write an algorithm to determine an integer N is prime Implement this into FORTRAN to find all prime numbers less than 500. [2]
পূর্ণসংখ্যা মৌলিক কি না তা নির্ণয়ের জন্য একটি এলগরিদম লিখ। 500 এর নীচে সকল মৌলিক সংখ্যা নির্ণয় কর।

[NUH-2004, 2007, NU(Pre)

Solution :

Algorithm :

- (i) READ N
- (ii) IF N ≤ 1 GOTO STEP (XI)
- (iii) IF N = 2 GOTO STEP (IX)
- (iv) CALCULATE K = N/2
- (v) LET J = 2
- (vi) IF MOD (N, J) = 0 GOTO STEP (XI)
- (vii) CALCULATE J = J + 1
- (viii) IF J ≤ K GOTO STEP (VI)
- (ix) PRINT N IS A PRIME
- (x) GOTO STEP (XII)
- (xi) PRINT N IS NOT A PRIME
- (xii) STOP

Fortran Program :

```

N = 2
IF(N .EQ. 2) GOTO 10
40  K = N/2
    J = 2
20  IF(MOD(N, J), EQ.0) GOTO 30
    J = J + 1
    IF(J .LT. K) GOTO 20
10   PRINT *, N
30   N = N + 1
    IF(N .LT. 500) GOTO 40
    STOP
    END

```

(i)

(ii)

(iii)

(iv)

(v)

(vi)

(vii)

(viii)

(ix)

(x)

(xi)

(xii)

N এর মান পড়

যদি $N \leq 1$ হয় তবে (XI) ধাপে যাও

যদি $N = 2$ হয় তবে (IX) ধাপে যাও

$K = N/2$ নির্ণয় কর

$J = 2$ ধর

যদি $MOD(N, J) = 0$ হয়, তবে (XI)

$J = J + 1$ নির্ণয় কর

যদি $J = K$ হয় তবে (VI) ধাপে যাও

ছাপাও N মৌলিক

(XII) ধাপে যাও

ছাপাও N মৌলিক নয়

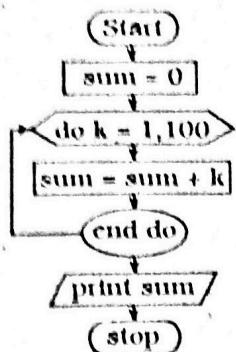
থাম।

Example 37. Write an algorithm and flowchart to compute the sum of $1 + 2 + 3 + \dots + 100$ using Do loop, and implement it in to fortran program. [$1 + 2 + 3 + \dots + 100$ এর যোগফল নির্ণয় করে একটি অ্যালগরিদম ও প্রবাহ চির আৰু এবং তা Fortran লিখে সম্পূর্ণ কৰা]

Solution :

Algorithm :

- (i) DEF SUM = 0
- (ii) DO FOR K = 1 TO 100 STEP 1
- (iii) COMPUTE SUM = SUM + K
- (iv) END DO
- (v) PRINT SUM
- (vi) STOP



অ্যালগরিদম :

- (i) SUM = 0 ধৰ
- (ii) DO যখন K = 1 থেকে 100 বৃদ্ধি 1
- (iii) SUM = SUM + K নির্ণয় কৰ।
- (iv) DO শেষ
- (v) SUM এর মান ছাপাও
- (vi) থাম

Program :

INTEGER SUM

SUM = 0

DO K = 1, 100

SUM = SUM + K

END DO

PRINT *, SUM

END

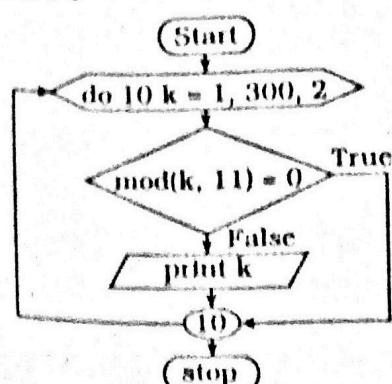
Example 38. Write an algorithm and draw a flowchart to print the positive odd numbers which less than 300 and not divisible by 11. [একটি অ্যালগরিদম লিখ ও একটি প্রবাহ চির থাক যা 11 দ্বাৰা বিভাজ্য সংখ্যা ব্যতি তেওঁ ধনাত্মক বিজোড় সংখ্যা ছাপাবে।]

Solution :

Algorithm :

- (i) DO FOR K = 1 TO 300 STEP 2
- (ii) IF MOD(K, 11) = 0 GOTO STEP (IV)
- (iii) PRINT K
- (iv) END DO
- (v) STOP

Flowchart :



অ্যালগরিদম :

- (i) DO যখন K = 1 থেকে 300 বৃদ্ধি 2
- (ii) যদি MOD(K, 11) = 0 হয় তবে (IV) ধাপে যাও
- (iii) K ছাপাও
- (iv) DO শেষ
- (v) থাম

Program :

PROGRAM ODD NUMBERS

DO 10 K = 1, 300, 2

IF (MOD (K, 11) EQ. 0) GOTO 10

PRINT *, K

10 CONTINUE

END

Computer Techniques Programming with Fortran
Example-49. Write an algorithm and draw a flowchart which read a integer n and test is prime or not using do loop. [Do loop ব্যবহার করে একটি অ্যালগরিদম একটি প্রোগ্রাম তৈরি কর যা একটি ধনাত্মক পূর্ণসংখ্যা n পড়বে এবং তা মৌলিক কিনা তা যাচাই করবে।]

Solution :

Algorithm :

- (i) READ N
- (ii) IF $N \leq 1$ GO TO STEP (VIII)
- (iii) DO FOR $K = 2$ TO $N/2$ STEP 1
- (iv) IF MOD (N, K) = 0 GOTO STEP (VIII)
- (v) END DO
- (vi) PRINT N IS A PRIME
- (vii) GOTO STEP (IX)
- (viii) PRINT N IS NOT A PRIME
- (ix) STOP

অ্যালগরিদম :

- (i) N এর মান পড়
- (ii) যদি $N \leq 1$ হয় তবে (VIII) ধাপে যাও
- (iii) DO যখন $K = 2$ থেকে $N/2$ বৃদ্ধি 1
- (iv) যদি $\text{MOD}(N, K) = 0$ হয় তবে (VIII) ধাপে যাও
- (v) DO শেষ
- (vi) ছাপাও N মৌলিক
- (vii) (IX) ধাপে যাও
- (viii) ছাপাও n মৌলিক নয়।
- (ix) থাম।

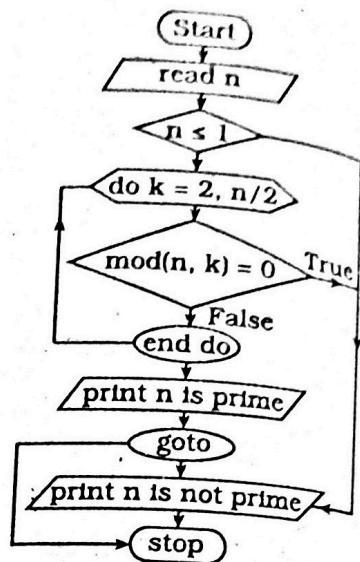
Program :

```

READ *, N
IF (N, LE, 1) GOTO 10
DO K = 2, N/2
IF(MOD (N, K), EQ, 0) GOTO 10
END DO
PRINT *, N, 'IS A PRIME'
GOTO 20
10 PRINT *, N, 'IS NOT A PRIME'
20 END

```

Example-50. Draw a flowchart to find the smallest of three given integers [তিনটি প্রদত্ত পূর্ণসংখ্যার মধ্যে ক্ষুদ্রতমটি বের করার জন্য একটি ফ্লোচার্ট অঙ্কন কর।]



EXERCISE-5

Part-A : Brief Questions [অতি সংক্ষিপ্ত প্রশ্ন]

1. Algorithm কি?

[NUH-2011, 2013, 2016, NU(Pass)-2016,
DUH (Aff. Coll.)-2018, NU(Pre)-2012, 2014]

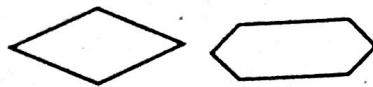
Ans : কম্পিউটার দ্বাৰা কোন নির্দিষ্ট সমস্যা সমাধানেৰ জন্য পর্যায়ক্রমিক নির্দেশাবলী বা প্ৰযোগৰিদম বলে। ধাপগুলো হতে পাৰে : (i) ইনপুট, (ii) এ্যাসাইনমেন্ট, (iii) পুনৰাবৃত্তি এবং (v) আউটপুট। অ্যালগৱিদম ভাষা।

2. Flowchart প্ৰবাহ চিত্ৰ কি?

[NUH-11, 12, 14, 16, NU(Pass)-17, NU(Pre)-12, 14]

Ans : ফ্ৰোচার্ট হল অ্যালগৱিদমেৰ পৰ্যায়ক্রমিক ধাপসমূহেৰ চিত্ৰভিত্তিক উপস্থাপনা। John Von Neumann ১৯৪৫ সালে সৰ্বোপৰ্যায় ফৰ্মাল ফ্ৰোচার্ট তৈৰী কৰেন। বিভিন্ন ধৰণেৰ প্ৰতীক বা বক্স সমূহ ফ্ৰোচার্ট ব্যবহৃত হয়।

3. IF Statement ও Loop এৰ জন্য ব্যবহৃত চিত্ৰ লিখ।

**Ans :**

4. কে সৰ্বোপৰ্যায় ফৰ্মাল ফ্ৰোচার্ট তৈৰী কৰেন?

Ans : ফ্ৰোচার্ট হল অ্যালগৱিদমেৰ পৰ্যায়ক্রমিক ধাপসমূহেৰ চিত্ৰভিত্তিক উপস্থাপনা। John von Neumann ১৯৪৫ সালে সৰ্বোপৰ্যায় ফৰ্মাল ফ্ৰোচার্ট তৈৰী কৰেন। বিভিন্ন ধৰণেৰ প্ৰতীক বা বক্স সমূহ ফ্ৰোচার্টেৰ জন্য ব্যবহৃত হয়।

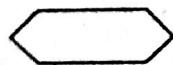
5. প্ৰবাহ চিত্ৰে দিক নির্দেশনাৰ জন্য কোন চিহ্ন ব্যবহৃত হয়?

Ans : →

6. Write the symbol for Do loop?

**Ans :**

7. Write the symbol for IF Statement.

**Ans :**

8. When a hexagon used in flow -chart? [ফ্ৰোচার্টে কখন ষড়ভুজ ব্যবহৃত হয়?] [NUH]

Ans : লুপ কাঠামো প্ৰকাশেৰ জন্য।

9. ফৱট্ৰান ভাষায় কত প্ৰকাৰ Structure ব্যবহাৰ কৰা হয়।

Ans : Flowchart ও অ্যালগৱিদম, বা অ্যালগৱিদমেৰ চিত্ৰভিত্তিক উপস্থাপনা। ফৱট্ৰান ভাষায় তিনি অ্যালগৱিদম Structure ব্যবহাৰ কৰা হয়। যেমন

(i) Sequential structure

(ii) Selective structure

(iii) Repetitive structure

10. Sequential Structure কি?

Ans : Sequential কাঠামোতে একটিৰ পৰ একটি Statement কাৰ্য্যকৰী হয়।

11. Selective Structure কি?

Ans : Selective কাঠামোতে একটি শৰ্ত পৰীক্ষা কৰে দুই বা ততোধিক বিকল্পেৰ মধ্যে থেকে শৰ্ত সত্ৰিয়াৰ উপৰ ভিত্তি কৰে একটি বিকল্প কাৰ্য্যকৰী হয়।

12. Repetitive Structure কি?

Ans : Repetitive কাঠামোতে একটি শৰ্ত সিঙ্ক ইওয়া সাপেক্ষে একটি বা একগুচ্ছ Statements