

Welcome
to

Data Structure and Algorithm

অধ্যায় ২: অ্যালগরিদম

কম্পিউটার সায়েন্সে অ্যালগরিদম

মৌলিক কেন?

ওয়েল, উত্তর সহজ. প্রোগ্রামগুলি তাদের কী করতে হবে তা না বলে কম্পিউটারগুলি কিছুই করবে না। অ্যালগরিদম কম্পিউটারকে ধাপে ধাপে নির্দেশনা দেয় নির্দিষ্ট কাজ সম্পন্ন করতে এবং সিদ্ধান্ত নিতে।



- ১। অ্যালগরিদমের কী? একটি ভাল অ্যালগরিদমের বৈশিষ্ট্যগুলো বর্ণনা কর।
- ২। Pseudo code বলতে কী বোঝায়? উদাহরণ দাও।
- ৩। Pseudo code এর সাথে Algorithm এর পার্থক্য উদাহরণসহ দেখাও।
- ৪। Algorithm Notation কী? বিভিন্ন ধরনের Algorithm Notation এর বর্ণনা দাও।
- ৫। ফ্লোচার্টে ব্যবহৃত বিভিন্ন ধরনের symbol/প্রতীক গুলোর বর্ণনা দাও।
- ৬। Structured programming বলতে কী বোঝায়? প্রোগ্রামিং এর ক্ষেত্রে বিভিন্ন ধরনের Control structure এর নামসহ ফ্লোচার্ট দেখাও।
- ৭। Subroutine বলতে কী বোঝায়?
- ৮। Complexity of Algorithm বা অ্যালগরিদমের কমপ্লেক্সিটি বলতে কী বোঝায়? বর্ণনা কর।

❖ কোন সমস্যা সমাধানের জন্য সুনির্দিষ্ট ধারাবাহিক নিয়মাবলী কে অ্যালগরিদম বলে।

এটি মূলত সমস্যা সমাধানের জন্য ব্যবহার করা হয়।

মনে করুন যে আপনি চা তৈরি করবেন। তাহলে চা তৈরীর অ্যালগরিদম টা কেমন হবে দেখা যাক।

১. প্রথমে আপনাকে একটি পাত্রে পানি দিতে হবে।
২. পানি ফুটানোর জন্য পাত্রটা চুলার উপরে রাখতে হবে।
৩. চুলায় আগুন দিতে হবে।
৪. পানি গরম হয়ে গেলে চার পাতা দিতে হবে।
৫. প্রয়োজনমতো চিনি দিয়ে পরিবেশন করতে হবে।

আমরা চা বানানোর জন্য উপরের যে স্টেপগুলো ফলো করলাম এটি মূলত একটি চা বানানোর অ্যালগরিদম।

ঠিক এভাবেই কম্পিউটার বিজ্ঞানের অনেক সমস্যা অ্যালগরিদম এর মাধ্যমে সমাধান করা হয়।

2.1 অ্যালগরিদমের বৈশিষ্ট্যগুলি বর্ণনা করুন।

একটি ভাল অ্যালগরিদমের বৈশিষ্ট্য হল:

- * নির্ভুলতা - পদক্ষেপগুলি সুনির্দিষ্টভাবে বলা হয়েছে।
- * স্বতন্ত্রতা - প্রতিটি ধাপের ফলাফল স্বতন্ত্রভাবে সংজ্ঞায়িত করা হয় এবং শুধুমাত্র ইনপুট এবং পূর্ববর্তী ধাপের ফলাফলের উপর নির্ভর করে।
- * সসীমতা - সীমিত সংখ্যক নির্দেশ কার্যকর হওয়ার পরে অ্যালগরিদম বন্ধ হয়ে যায়।
- * ইনপুট - অ্যালগরিদম ইনপুট গ্রহণ করে।
- * আউটপুট - অ্যালগরিদম আউটপুট তৈরি করে।
- * সাধারণতা - অ্যালগরিদম ইনপুটগুলির একটি সেটে প্রযোজ্য।

অ্যালগরিদম এবং সিউডো কোডের মধ্যে পার্থক্য:

একটি অ্যালগরিদম হল
একটি পদ্ধতিগত যৌক্তিক
পদ্ধতি যা একটি কম্পিউটারে
সমস্যা সমাধানের জন্য ব্যবহৃত
হয়। একটি অ্যালগরিদম একটি
সমস্যার সমাধানকে
ধাপ বা নির্দেশের একটি
সুনির্দিষ্ট সেট হিসাবে
উপস্থাপন করে।

যখন ছদ্ম কোড সরল ইংরেজিতে
বিবৃতি।
সিউডো-কোড একটি
নির্দিষ্ট প্রোগ্রামিং
ভাষার
সিনট্যাক্স ব্যবহার করে না।

অ্যালগরিদম

বনাম

সিউডোকোড

অ্যালগোরিদম

একটি সমস্যা কিভাবে
সমাধান করতে হয় তার একটি
দ্ব্যর্থহীন স্পেসিফিকেশন

সমস্যাটি সহজ করতে
এবং বুঝতে সাহায্য করে

সুডোকোড

একটি কম্পিউটার প্রোগ্রাম
বা অন্যান্য অ্যালগরিদমের
অপারেটিং নীতির একটি
অনানুষ্ঠানিক উচ্চ-স্তরের বর্ণনা

একটি অ্যালগরিদম বিকাশের
একটি পদ্ধতি

□□□.□□□□□□□.□□□ দেখুন

Example of Pseudo code

Algorithm:

Step-1: Set a variable as GRADE.

Step-2: If student's GRADE is greater than or equal to 60 then

Print "pass"

otherwise

Print "fail"

Step-3: End Processing.

Pseudo code:

If student_GRADE > or = 60

Print "pass"

else

Print "fail"

Python Code:

```
GRADE=Input (GRADE)
```

```
If GRADE > = 60:
```

```
    print ("Pass")
```

```
else:
```

```
    print ("Fail")
```


অ্যালগরিদম নোশেশন

অ্যালগরিদম স্বরলিপি হিসাবেও পরিচিত, অ্যালগরিদমগুলিকে প্রতিনিধিত্ব করতে ব্যবহৃত প্রমিত স্বরলিপিকে বোঝায়। এই স্বরলিপিটি একটি অ্যালগরিদমের পদক্ষেপ এবং যুক্তিকে একটি সুনির্দিষ্ট, দ্ব্যর্থহীন উপায়ে বর্ণনা করতে ব্যবহৃত হয় যা অন্যদের দ্বারা সহজেই বোঝা এবং প্রয়োগ করা যায়।

2.4 অ্যালগরিদমের জটিলতা বর্ণনা করুন

একটি অ্যালগরিদমের জটিলতা হল একটি প্রদত্ত আকারের ইনপুটের জন্য একটি অ্যালগরিদম দ্বারা প্রয়োজনীয় চলমান সময় এবং/অথবা স্থানের পরিমাপ।

- একটি নির্দিষ্ট ইনপুটের জন্য একটি অ্যালগরিদমের চলমান সময় নির্বাহ করা অপারেশনের সংখ্যার উপর নির্ভর করে। অপারেশনের সংখ্যা যত বেশি, অ্যালগরিদমের চলমান সময় তত বেশি।
- একটি অ্যালগরিদমের জন্য প্রয়োজনীয় স্থান হল ইনপুটের আকারের একটি ফাংশন হিসাবে একটি উদাহরণ সমাধান করার জন্য প্রয়োজনীয় মেমরি স্থানের পরিমাণ। এটি একটি প্রোগ্রাম চালানো এবং আউটপুট উত্পাদন করার জন্য একটি অ্যালগরিদম দ্বারা প্রয়োজনীয় মেমরি।

অ্যালগরিদমের প্রকার



স্মার্ট ইঞ্জিন
অ্যালগরিদম



এনক্রিপশন
অ্যালগরিদম



লোগী
অ্যালগরিদম



রিকার্সিভ
অ্যালগরিদম



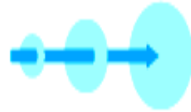
ব্যাকট্র্যাকিং
অ্যালগরিদম



বিভক্ত
এবং-জয়
অ্যালগরিদম



ডায়নামিক
প্রোগ্রামিং
অ্যালগরিদম



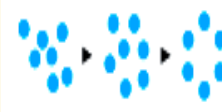
ক্রট-ফোর্স
অ্যালগরিদম



বাছাই
অ্যালগরিদম



হ্যাশিং
অ্যালগরিদম



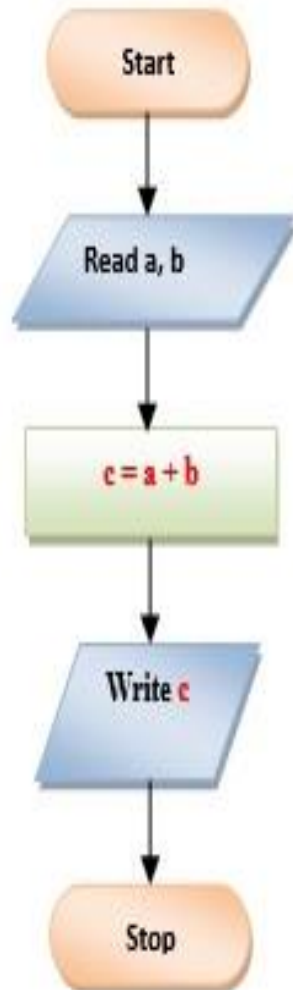
র্যান্ডমাইজড
অ্যালগরিদম

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;
}
```

Finding Area of the square

Algorithm

1. Start
2. Read length, L
3. $area = L * L$
4. Print or display **area**
5. Stop

Flowchart



Program

```
#include<stdio.h>

int main()
{
    int L, area;

    printf("Enter length of square L: ");
    scanf("%d", &L);

    area = L*L;

    printf("Area of square is: %d", area);

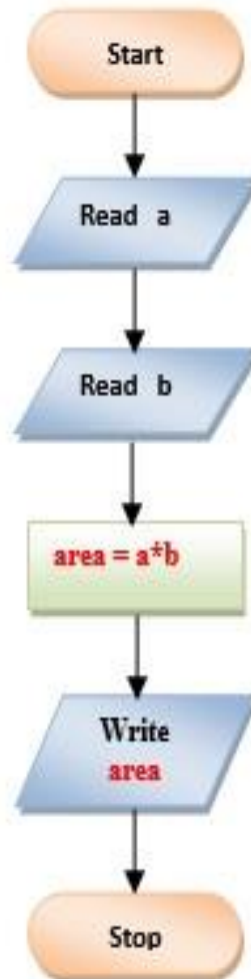
    return 0;
}
```

Finding Area of the rectangle

Algorithm

1. Start
2. Read side length, a
3. Read side length b
4. $area = a * b$
5. Print or display **area**
6. Stop

Flowchart



Program

```
#include<stdio.h>

int main()
{
    int a, b, area;
    printf("Enter side length a: \n");
    scanf("%d", &a);

    printf("Enter side length b: \n");
    scanf("%d", &b);

    area = a*b;

    printf("Area of rectangle is: %d ", area);

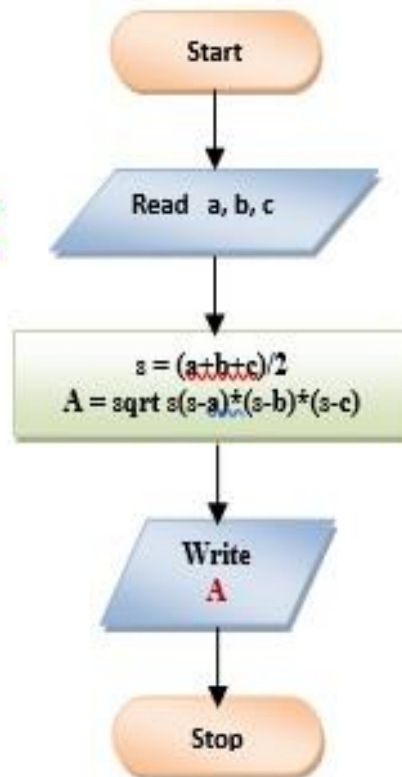
    return 0;
}
```


Area of a triangle where three sides are given

Algorithm

1. Start
2. Read a, b, c
3. $s = (a+b+c)/2$
4. $A = \sqrt{s(s-a)(s-b)(s-c)}$
5. Print or display A
6. Stop

Flowchart



Program

```
#include<stdio.h>
#include<math.h>

int main()
{
    int a, b, c;
    float s, A;

    printf("Enter values of a, b, c: \n");
    scanf("%d %d %d", &a, &b, &c);

    s = (a+b+c)/2;
    A = sqrt (s*(s-a)*(s-b)*(s-c));

    printf("Area of triangle is: %f", A);

    return 0;
}
```

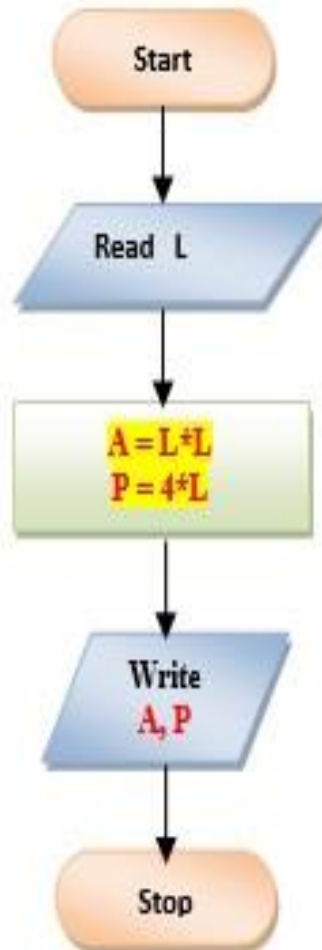
```
E:\WACHEMO\Sept 2016-Jan 2017\p
Enter values of a, b, c:
5
8
10
Area of triangle is: 14.071247
```


Find the area & perimeter of a square

Algorithm

1. Start
2. Read length L
3. Area $A = L * L$
4. Perimeter $P = 4 * L$
5. Print or display A, P
6. Stop

Flowchart



Program

```
#include<stdio.h>

int main()
{
    int L, A, P;

    printf("Enter length of a square L: ");
    scanf("%d", &L);

    A = L * L;
    P = 4 * L;

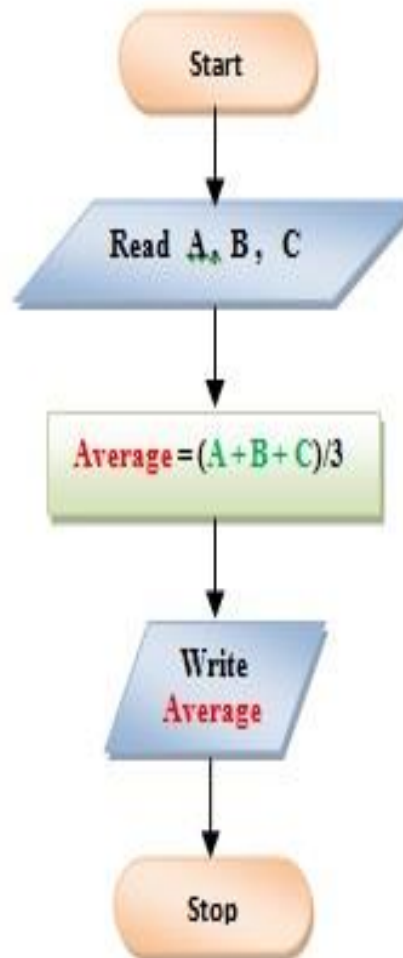
    printf("Area = %d\n", A);
    printf("Perimeter = %d", P);
    return 0;
}
```

Calculating the average for 3 numbers

Algorithm

1. Start
2. Read 3 numbers A, B, C
3. Calculate the average by the equation:
 $Average = (A + B + C)/3$
4. Print average
5. Stop

Flowchart



Program

```
#include<stdio.h>

int main()
{
    int A, B, C;
    float Average;

    printf("Enter values of A, B, C: \n");
    scanf("%d %d %d", &A, &B, &C);

    Average = (A+B+C)/3;

    printf("Average of given 3 numbers is: %f", Average);

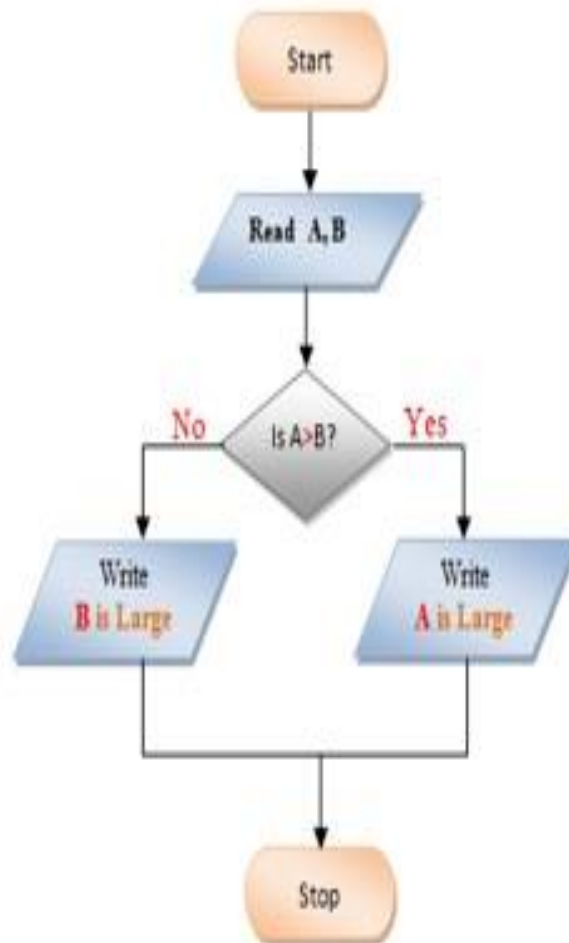
    return 0;
}
```

Greatest of two numbers

Algorithm

1. Start
2. Read A, B
3. If $A > B$ then
 Print A is large
 else
 Print B is large
4. Stop

Flowchart



Program

```
#include<stdio.h>

int main()
{
    int A, B;

    printf("Enter values of A, B: ");
    scanf("%d %d", &A, &B);

    if (A>B)
        printf("A is Larger");
    else
        printf("B is Larger");

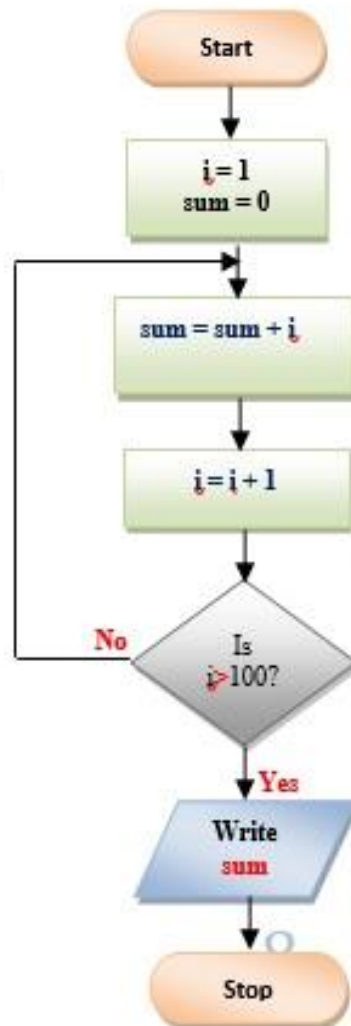
    return 0;
}
```

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

```
#include<stdio.h>

int main()
{
    int i, sum;

    sum = 0;
    for(i=1; i<101;i++)
    {
        sum = sum + i;
    }

    printf("Sum of integers from 1 to 100 is: %d", sum);

    return 0;
}
```



The End