অধ্যায়-৬

সিদ্ধান্ত গ্রহণ
(Decission Making)

৬.০ ভূমিকা (Introduction) ঃ

প্রায় সব প্রোগ্রামিং ল্যাংগুয়েজেই সিদ্ধান্ত গ্রহণ (Decision making) এর ব্যাপার আছে। Decision making বা সিদ্ধান্ত গ্রহণ যে-কোনো প্রোগ্রামিং ল্যাংগুয়েজ-এর একটি গুরুত্বপূর্ণ বিষয়। প্রোগ্রামের পরবর্তী অ্যাকশন কার্যকর হবে কি হবে না তা নির্ভর করবে প্রোগ্রামে প্রদত্ত কিছু শর্তের উপর। কন্ডিশন সত্য হলে এক ধরনের সিদ্ধান্ত কার্যকর হবে আর কন্ডিশন মিথ্যা হলে অন্য ধরণের সিদ্ধান্ত কার্যকর হবে। উক্ত অধ্যায়ে আমরা পাইথনে ব্যবহৃত বিভিন্ন সিদ্ধান্ত গ্রহণ (Decision making) সম্পর্কিত স্টেটমেন্ট ও তাদের ব্যবহার সম্পর্কিত বিষয়াদি নিয়ে বিশদ আলোচনা করব।

৬.১ কন্ডিশনাল ও আনকন্ডিশনাল ব্রাঞ্চিং ফ্লো (Conditional & unconditional branching flow) ঃ

প্রোগ্রামে যদি কখনো কোনো স্টেটমেন্ট দুই বা ততোধিক বার সম্পাদনের প্রয়োজন হয় কিংবা কোনো স্টেটমেন্ট কোন শর্ত সাপেক্ষে অথবা কোন স্টেটমেন্টের ফলাফলের ভিত্তিতে নির্বাহ (Execute) করতে হয় কিংবা এক্সিকিউট বন্ধ রাখতে হয় অথবা প্রোগ্রামের নিয়ন্ত্রণ অন্য কোন স্টেটমেন্টে স্থানান্তর করতে হয় তখন প্রোগ্রামের নির্বাহ কোন প্রক্রিয়ায় সম্পন্ন হবে তা নির্ধারণ করার প্রক্রিয়াকে প্রোগ্রাম ফ্লো কন্ট্রোল বলে। ফ্লো কন্ট্রোল প্রক্রিয়া প্রধানত দুই ধরনের হয়, যথা–

- ব্রাঞ্চিং (Branching) ও
- লুপিং (Looping)

ব্রাঞ্চিং আবার দুই প্রকার, যথা–

- কন্ডিশনাল ব্রাঞ্চিং ফ্লো (Conditional branching flow) ও
- আন কন্ডিশনাল ব্রাঞ্চিং ফ্লো (Unconditional branching flow)

কন্ডিশনাল ব্রাঞ্চিং ফ্লো (Conditional branching flow) ঃ যখন কোনো প্রোগ্রাম কন্ডিশনাল স্টেটমেন্টের প্রভাবে এক্সিকিউট হয়, তখন তাকে কন্ডিশনাল ব্রাঞ্চিং ফ্লো বলে। এতে প্রোগ্রামে ব্যবহৃত কন্ডিশন সত্য হলে এক ধরনের ফলাফল আর মিথ্যা হলে অন্য ধরনের ফলাফল পাওয়া যায়। পাইথনে মূলত ,চার ধরনের কন্ডিশনাল ব্রাঞ্চিং ফ্লো স্টেটমেন্ট আছে, যথা–

- if স্টেটমেন্ট
- if else স্টেটমেন্ট
- elif স্টেটমেন্ট ও
- nested if স্টেটমেন্ট

আন-কন্ডিশনাল ব্রাঞ্চিং ফ্লো (Unconditional branching flow) ঃ যখন কোনো প্রোগ্রাম আন কন্ডিশনাল স্টেটমেন্টের প্রভাবে এক্সিকিউট হয়, তখন তাকে আন-কন্ডিশনাল ব্রাঞ্চিং ফ্লো বলে। এতে প্রোগ্রামের কোনো অংশ বার বার আবর্তিত হতে থাকে কিংবা নির্দিষ্ট কোনো স্টেটমেন্টে jump করে।

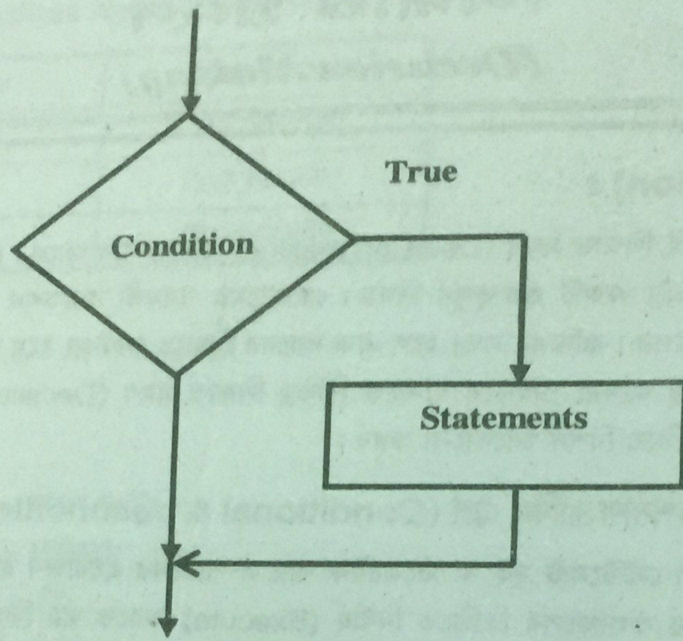
৬.২ if স্টেটমেন্ট, if else স্টেটমেন্ট ও elif স্টেটমেন্ট (if, if else & elif statements in python) ঃ

if স্টেটমেন্ট (if Statement) ঃ প্রোগ্রামে শর্তসাপেক্ষে কোনো স্টেটমেন্ট সম্পাদনের জন্য if স্টেটমেন্ট ব্যবহৃত হয়। পাইথনে if স্টেটমেন্ট ব্যবহার করে নির্দিষ্ট একটি কন্ডিশনের উপর ভিত্তি করে কিছু স্টেটমেন্ট বা কোড ব্লককে রান করানো যায়। যদি একটি কন্ডিশন বা এক্সপ্রেশন সত্য হয় তাহলে এর আওতাভুক্ত স্টেটমেন্ট বা স্টেটমেন্টসমূহ রান হয়।

সিনট্যাক্স (Syntax) ঃ

```
if Condition:
    Statements
```

ফ্লোচার্ট (Flowchart) ঃ



চিত্র ঃ if স্টেটমেন্ট

উদাহরণ-১ ঃ

```
if 10 > 5:
    print("10 greater than 5") # এই স্টেটমেন্টটি if কন্ডিশনের আওতাভুক্ত
    print("IF scope finished") # এই  স্টেটমেন্টটিও if কন্ডিশনের আওতাভুক্ত
print("Program ended")              # এই  স্টেটমেন্টটি if কন্ডিশনের আওতাভুক্ত নয়
```

আউটপুট ঃ

```
10 greater than 5
IF scope finished
Program ended
```

উল্লেখ্য যে, পাইথনের if স্টেটমেন্টে colon (:) খুবই জরুরি। কারণ কোলনের মাধ্যমেই হেডার ও প্রোগ্রাম বডিকে আলাদা করা হয়। কোলনের পরবর্তী লাইনে ইনডেন্টেশন বাধ্যতামূলক। না হলে পাইথনে এরর দেখাবে।

উপরোক্ত প্রোগ্রামে if কন্ডিশন দিয়ে 10, 5 এর চেয়ে বড় কি না চেক করা হচ্ছে । কন্ডিশন সত্য অর্থাৎ বড় হলে এর ভিতরের দুই লাইন এক্সিকিউট হচ্ছে। যেহেতু এটা সত্য কন্ডিশন। তাই 10 greater than 5 এবং IF scope finished লাইন দুটি প্রিন্ট হচ্ছে। আবার, Program ended লাইনটি এসব কন্ডিশন এর বাইরের একটি সাধারণ স্টেটমেন্ট। আর তাই এই লাইনটিও সাধারণভাবেই প্রিন্ট হচ্ছে।

আর হ্যাঁ, এই যে বলা হচ্ছে if স্টেটমেন্টের আওতাভুক্ত বা আওতাভুক্ত নয়, এটি নির্ধারণ হয় indentation তথা স্টেটমেন্টের সামনে যথাযথ হোয়াইট স্পেস ব্যবহার করে। ইনডেন্টেশন হলো কিছু বিশেষ স্টেটমেন্টের পর কোড একটু ডানে সরিয়ে লেখা যাতে বুঝা যায় কোন ব্লকটি কার অন্তর্গত। সহজ কথায় ইন্ডেন্টেশন হলো সাজিয়েগুছিয়ে কোড লেখা। পাইথনে ইন্ডেন্টেশন নানাভাবেই করা যায়। তবে স্টান্ডার্ড হলো চারটি স্পেস চেপে কোড ইনডেন্ট করা। তবে আমরা চাইলে দুইটি বা তিনটি স্পেস চেপেও কাজ করতে পারি। শুধু মাথায় রাখতে হবে, সব জায়গায় যেন একই সংখ্যক স্পেস চেপে কোড ইনডেন্ট করা হয়। ট্যাব চেপেও কিন্তু কোড ইনডেন্ট করা যায়।

উপরের উদাহরণে if 10 > 5: হচ্ছে কন্ডিশন বা এক্সপ্রেশন। এর নিচের দুই লাইন কিন্তু if এর আওতাভুক্ত। কারণ এর সামনে indentation বা স্পেস ব্যবহার করে ডান দিকে সরিয়ে নেয়া হয়েছে। অন্যান্য প্রোগ্রামিং ল্যাংগুয়েজে এই কাজটি করা হয়ে থাকে { } ব্যবহার করে।

উদাহরণ-২ ঃ

```
>>> a = 10
>>> a
10
>>> if a < 20:
... print('a is less than 20.')
    File "<stdin>", line 2
        print('a is less than 20.')
            ^
IndentationError: expected an indented block
```

উপরোক্ত প্রোগ্রামটি রান করলে পাইথন একটি এরর থ্রো করেছে। এটা হচ্ছে Indentation error। পাইথনে ইনডেন্টেশন বাধ্যতামূলক। তাই যখনই কোথাও ইনডেন্টেশন না করা হয় তখনই পাইথন এরকম এরর থ্রো করে। সঠিক ইনডেন্টেশনের মাধ্যমে এই ধরনের এরর দূর করা যায়।

উদাহরণ-৩ ঃ

```
>>> a=30
>>> b=10
>>> if a>b:
        print("a is greater than b")


a is greater than b
>>>
```

উদাহরণ-৪ ঃ

```
>>> a=30
>>> b=50
>>> if a>b:
        print("a is greater than b")

>>>
```

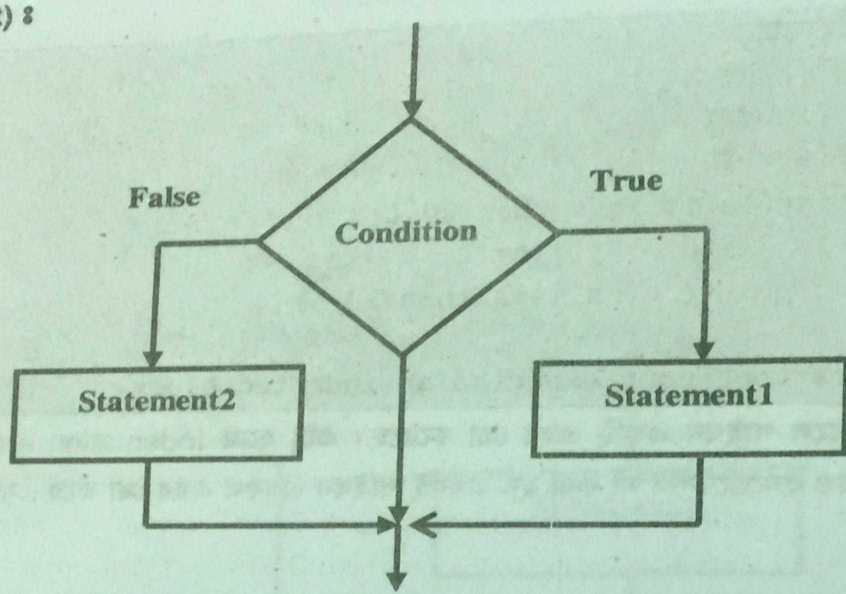
উদাহরণ-৩ ও উদাহরণ-৪ এর আউটপুট লক্ষ করলে দেখা যাবে যে, উদাহরণ-৩ এর ক্ষেত্রে কন্ডিশন সত্য হওয়ায় প্রিন্ট করছে a is greater than b কিন্তু উদাহরণ-৪ এর ক্ষেত্রে কন্ডিশন সত্য না হওয়ায় কিছুই প্রিন্ট করছে না।

if else স্টেটমেন্ট (if else statement) ঃ প্রোগ্রামে 'অন্যথায়' অর্থে if স্টেটমেন্টের সাথে else স্টেটমেন্ট ব্যবহৃত হয়। শর্তসাপেক্ষে কোনো স্টেটমেন্ট সম্পাদনের জন্য এটি ব্যবহৃত হয়। পাইথনে if স্টেটমেন্ট ব্যবহার করে নির্দিষ্ট একটি কন্ডিশনের উপর ভিত্তি করে কিছু স্টেটমেন্ট বা কোড ব্লককে রান করানো যায়। যদি কন্ডিশনটি বা এক্সপ্রেশনটি সত্য হয় তাহলে এর আওতাভুক্ত স্টেটমেন্ট বা স্টেটমেন্টসমূহ রান হয়, আর মিথ্যা হলে else স্টেটমেন্টের পরবর্তী স্টেটমেন্ট বা স্টেটমেন্টসমূহ রান হয়।

সিনট্যাক্স (Syntax) ঃ

```
if Condition:
        Statements
else:
        Statements
```

ফ্লোচার্ট (Flowchart) :



চিত্রঃ if else স্টেটমেন্ট এর ফ্লোচার্ট

উদাহরণ-১ :

```
>>> a=10
>>> b=20
>>> if a>b:
	print("a is larger than b")
else:
	print ("b is larger than a")

b is larger than a
>>>
```

উদাহরণ-২ :

```
>>> number=int(input("Enter your Number"))
Enter your Number 21
>>> if number%2==0:
	print("Even Number")
else:
	print("Odd Number")

Odd Number
>>>
```

উদাহরণ-৩ :

```
>>> Number=int(input("Enter the Number:"))
Enter the Number:-3
>>> if Number>=0:
	print("This is an Positive Number or Zero")
else:
	print("This is a Negative Number")

This is a Negative Number
>>>
```

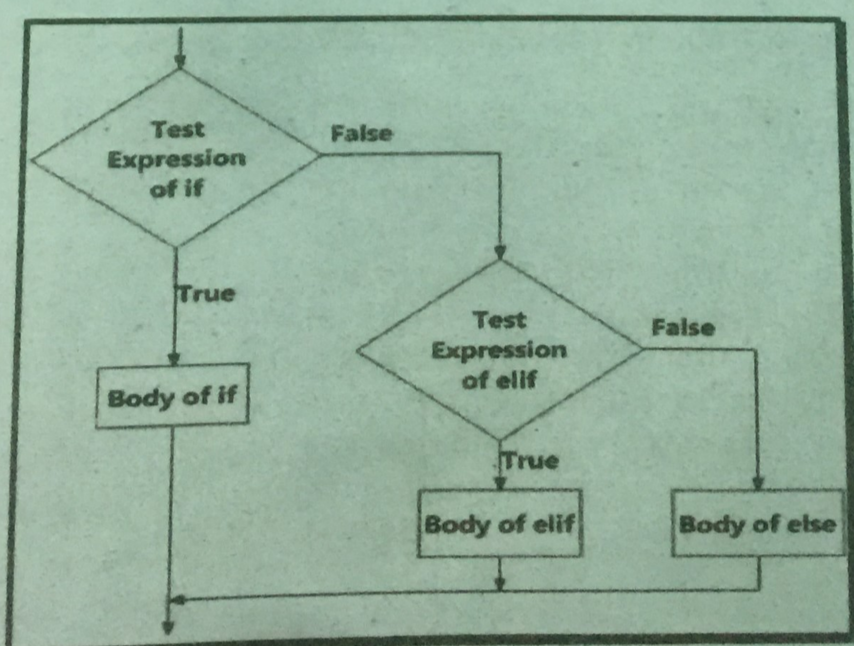
```
import cmath
a=int(input("Enter the Value of a:"))
b=int(input("Enter the Value of b:"))
c=int(input("Enter the Value of c:"))
if (a+b)>c and (b+c)>a and (c+a)>b:
    S=(a+b+c)/2
    Area=cmath.sqrt(S*(S-a)*(S-b)*(S-c))
    print (Area)
else:
    print("Triangle is not Possible")
```

elif স্টেটমেন্ট (elif Statement) ঃ else if এর সংক্ষিপ্ত রূপ হলো elif স্টেটমেন্ট যখন একই কন্ডিশন বার বার chick করার প্রয়োজন হয় তখন elif স্টেটমেন্ট ব্যবহার করা হয়। প্রোগ্রামে একাধিক 'অন্যথায়' অর্থে if স্টেটমেন্টের সাথে elif স্টেটমেন্ট ব্যবহৃত হয়। একাধিক if স্টেটমেন্টের বিকল্প হিসেবে elif স্টেটমেন্ট ব্যবহৃত হয়। elif স্টেটমেন্ট if এবং else স্টেটমেন্টের মাঝখানে বসে। পাইথনে if স্টেটমেন্ট ব্যবহার করে নির্দিষ্ট একটি কন্ডিশনের উপর ভিত্তি করে কিছু স্টেটমেন্ট বা কোড ব্লককে রান করানো যায়। যদি কন্ডিশনটি বা এক্সপ্রেশনটি সত্য হয় তাহলে এর আওতাভুক্ত স্টেটমেন্ট বা স্টেটমেন্টসমূহ রান হয়, আর মিথ্যা হলে একাধিক শর্ত যাচাই করার জন্য elif স্টেটমেন্টসমূহ রান হয়। যদি if বা elif স্টেটমেন্টসমূহ রান না করে তাহলে সবশেষে else স্টেটমন্টে রান হয়।

সিনট্যাক্স (Syntax) ঃ

```
if Condition1:
        Statements
elif Condtion2:
        Statements
elif Condtion3:
        Statements
.........
.........
elif Condtion n:
        Statements
else:
        Statements
```

ফ্লোচার্ট (Flowchart) ঃ



চিত্রঃ elif স্টেটমেন্ট এর ফ্লোচার্ট

উদাহরণ-১ ঃ

```
>>> Number=int(input("Enter the Number:"))
Enter the Number:0
>>> if Number>0:
    print("This is a Positive Number")
elif Number==0:
    print("This is a Zero")
else:
    print("This is a Negative Number")

This is a Zero
>>>
```

উদাহরণ-২ ঃ

```
>>> a=int(input("Enter the value of a:"))
Enter the value of a:15
>>> b=int(input("Enter the value of b:"))
Enter the value of b:47
>>> c=int(input("Enter the value of c:"))
Enter the value of c:32
>>> if (a>b)and (a>c):
    print ("Largest Number is a")
elif (b>c)and (b>a):
    print ("Largest Number is b")
else:
    print ("Largest Number is c")

Largest Number is b
>>>
```

উদাহরণ-৩ ঃ

```
>>> Marks=int(input("Enter Your Marks:"))
Enter Your Marks:72
>>> if Marks>=80:
    print ("Your Grade is A+")
elif Marks>=70:
    print ("Your Grade is A")
elif Marks>=60:
    print ("Your Grade is A-")
elif Marks>=60:
    print ("Your Grade is B+")
elif Marks>=50:
    print ("Your Grade is B")
elif Marks>=40:
    print ("Your Grade is C")
else:
    print ("Your Grade is F")

Your Grade is A
>>>
```

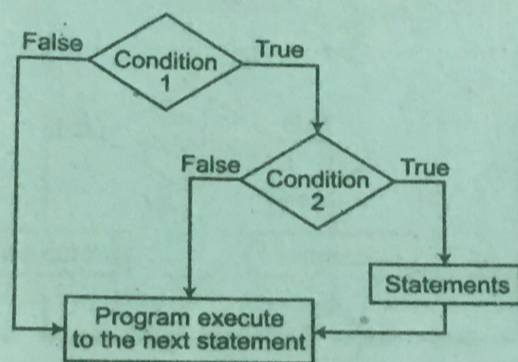
৬.৩ nested if, nested if else, nested elif স্টেটমেন্ট (nested if, nested if else, nested elif statements in python) ঃ

nested if স্টেটমেন্ট (nested if statement) ঃ একটি if স্টেটমেন্টের মধ্যে একাধিক if স্টেটমেন্ট ব্যবহার করলে তাকে nested if স্টেটমেন্ট বলে। এক্ষেত্রে প্রতিটি if স্টেটমেন্ট ব্যবহার করে নির্দিষ্ট একটি কন্ডিশনের উপর ভিত্তি করে কিছু স্টেটমেন্ট বা কোড ব্লককে রান করানো যায়। যদি কন্ডিশনটি বা এক্সপ্রেশনটি সত্য হয় তাহলে এর আওতাভুক্ত স্টেটমেন্ট বা স্টেটমেন্টসমূহ রান হয় আর মিথ্যা হলে পরবর্তী কোন অ্যাকশন সম্পন্ন হবেনা।

সিনট্যাক্স (Syntax) ঃ

```
if condition 1 :
    statement 1, # code to be executed when condition 1 is true.
    if condition 2 :
        statement 2 # code to be executed when condition 1 and
        .......          condition 2 both are true.
        .......
        if condition n :
                statement n
```

ফ্লোচার্ট (Flowchart) ঃ



চিত্র ঃ Flowchart of nested if statement.

উদাহরণ-১ ঃ

```
num = int (input ("please enter the number:"))
if num > 0 :
    if num < 10 :
            print ("The number is single word")
```

উদাহরণ-২ ঃ

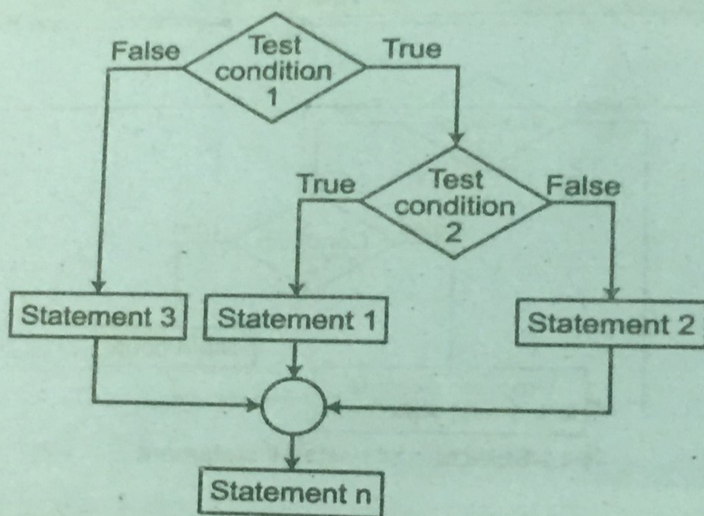
```
num = int (input ("please enter the number"))
if num < 500 :
    print ("First condition is true")
    if num > 100:
            print ("First and second conditions are true")
```

Nested if else স্টেটমেন্ট (nested if else statement) ঃ একটি if else স্টেটমেন্টের মধ্যে একাধিক if else স্টেটমেন্ট ব্যবহার করলে তাকে nested if else স্টেটমেন্ট বলে। এক্ষেত্রে if স্টেটমেন্ট ব্যবহার করে নির্দিষ্ট একটি কন্ডিশনের উপর ভিত্তি করে কিছু স্টেটমেন্ট বা কোড ব্লককে রান করানো যায়। যদি কন্ডিশনটি বা এক্সপ্রেশনটি সত্য হয় তাহলে এর আওতাভুক্ত স্টেটমেন্ট বা স্টেটমেন্টসমূহ রান হয়। মিথ্যা হলে if else স্টেটমেন্টের মধ্যে আরেকটি কন্ডিশন ব্যবহার করা হবে। যদি কন্ডিশনটি বা এক্সপ্রেশনটি সত্য হয় তাহলে এর আওতাভুক্ত স্টেটমেন্ট বা স্টেটমেন্টসমূহ রান হবে। আর মিথ্যা হলে if else স্টেটমেন্টের মধ্যে আরেকটি কন্ডিশন ব্যবহার করা হবে। এভাবে এ প্রক্রিয়াটি চলতে থাকবে। সর্বশেষ কন্ডিশনটি মিথ্যা হলে এর else স্টেটমেন্টের আওতাভুক্ত স্টেটমেন্ট বা স্টেটমেন্টসমূহ রান হবে।

সিনট্যাক্স (Syntax) ঃ

```
if (test condition 1) :
    # if test condition 1 is True then it will ckeck for test condition 2
    if (test condition 2) :
        # if test condition 2 is true then these statement will be executed.
        Test condition 2 True statement.
    else :
        # If test condition 2 is false then these statement will be executed.
        Test condition 2 false statements.
else :
    # if test condition 1 is false then these statement will be executed.
        Test condition 1 false statement.
```

ফ্লো-চার্ট (Flowchart) ঃ



চিত্র ঃ nested if else স্টেটমেন্ট এর ফ্লোচার্ট

উদাহরণ-১ ঃ

```
a=int(input("Enter the Value of a:"))
b=int(input("Enter the Value of b:"))
c=int(input("Enter the Value of c:"))
if(a>b):
    if(a>c):
        print("Largest Number is a")
    else:
        print("Largest Number is c")
else:
    if(b>c):
        print("Largest Number is b")
    else:
    print("Largest Number is c")
```

উদাহরণ-২ ঃ

```
a = int (input ("Please enter the value of a:"))
b = int (input ("Please enter the value of b:"))
c = int (input ("Please enter the value of c:"))
if (a<b):
    if(a<c):
        print ("Smallest value is a")
    else:
        print ("Smallest value is c")
else:
    if (b<c):
        print ("Smallest value is b")
    else:
        print ("Smallest value is c")
```

উদাহরণ-৩ ঃ

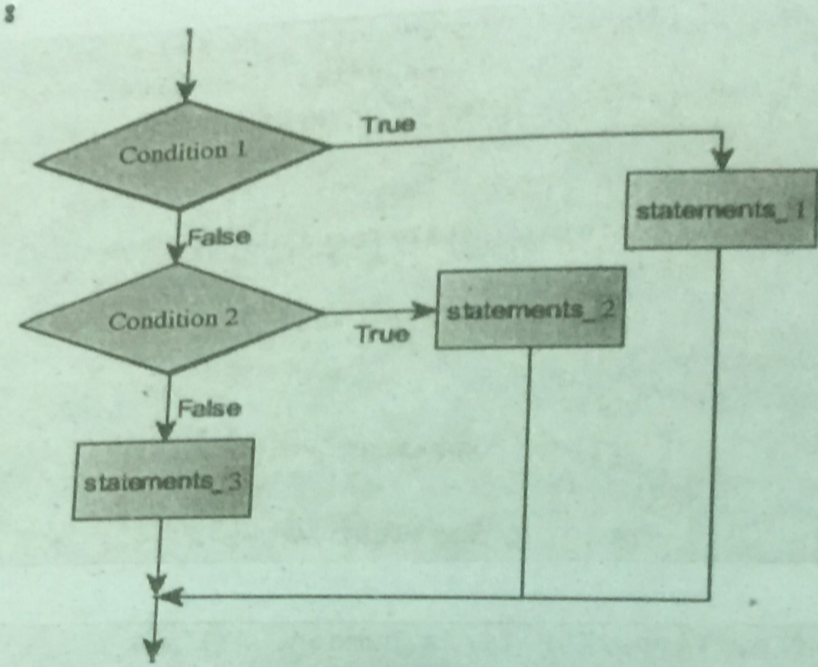
```
num = float(input("Enter a number: "))
if num>= 0:
    if num == 0:
        print("Zero")
    else:
        print("Positive number")
else:
    print("Negative number")
```

nested elif স্টেটমেন্ট (nested elif Statement)ঃ একটি **elif** স্টেটমেন্টের মধ্যে একাধিক **elif** স্টেটমেন্ট ব্যবহার করলে তাকে **nested elif** স্টেটমেন্ট বলে। এক্ষেত্রে **if** স্টেটমেন্ট ব্যবহার করে নির্দিষ্ট একটি কন্ডিশনের উপর ভিত্তি করে কিছু স্টেটমেন্ট বা কোড বস্নককে রান করানো যায়। যদি কন্ডিশনটি বা এক্সপ্রেশনটি সত্য হয় তাহলে এর আওতাভুক্ত স্টেটমেন্ট বা স্টেটমেন্টসমূহ রান হয়। মিথ্যা হলে **elif** স্টেটমেন্টের মধ্যে আরেকটি কন্ডিশন ব্যবহার করা হবে। যদি কন্ডিশনটি বা এক্সপ্রেশনটি সত্য হয় তাহলে এর আওতাভুক্ত স্টেটমেন্ট বা স্টেটমেন্টসমূহ রান হবে। আর মিথ্যা হলে **elif** স্টেটমেন্টের মধ্যে আরেকটি কন্ডিশন ব্যবহার করা হবে। এভাবে এ প্রক্রিয়াটি চলতে থাকবে। সর্বশেষ কন্ডিশনটি মিথ্যা হলে এর **else** স্টেটমেন্টের আওতাভুক্ত স্টেটমেন্ট বা স্টেটমেন্টসমূহ রান হবে।

সিনট্যাক্স (Syntax) ঃ

```
if  Condition1:
     Statements
elif  Condition2:
     Statements
.........
.........
elif  Condition n:
     Statements
else:
     Statements
```

ফ্লোচার্ট (Flowchart) ঃ



চিত্র ঃ if.....elif.......else স্টেটমেন্ট এর ফ্লোচার্ট

উদাহরণ-১ ঃ

```
choice=input("Enter your choice:")
if choice == 'a':
    print("You chose 'a'.")
elif choice == 'b':
    print("You chose 'b'.")
elif choice == 'c':
    print("You chose 'c'.")
else:
    print("Invalid choice.")
```

উদাহরণ-২ ঃ

```
num = int(input("Enter Your Number:"))
if num == 9:
    print("Number is 9")
elif num == 13:
    print("Number is 13")
elif num == 17:
    print("Number is 17")
else:
    print("Number isn't 9, 13 or 17")
```

উদাহরণ-৩ ঃ

```
a = int(input())
if a == 13:
    print('a is equal to 13.')
elif a < 13:
    print('a is less than 13.')
elif a > 13 and a < 20:
    print('a is inbetween 13 and 20.')
else:
    print('a is greater than 20.')
```

৬.৪ ব্রাঞ্চিং স্টেটমেন্ট ব্যবহার করে প্রোগ্রাম (Simple programs using branching statements) ঃ

প্রোগ্রাম-১ ঃ কোনো শব্দ প্যালিনড্রোম (কোনো শব্দ, সংখ্যা বা সিকুয়েন্স যাদের উল্টালেও ঐ এক জিনিসই থাকে যেমন– 707 বা Madam) কি না তা নির্ণয়ের প্রোগ্রাম।

```
print("Please input your word:")
word = input()
word = word.casefold()
reversed_word = word[::-1]

if word == reversed_word:
    print('It is a pallindrome.')
else:
    print('It is not a pallindrome.')
```

প্রোগ্রাম-২ ঃ nested if else ব্যবহার করে মার্কস ইনপুট নিয়ে গ্রেড নির্ণয়ের প্রোগ্রাম ঃ

```
x = int(input("Enter your Marks:"))
def grade(x):
    if x >= 90:
        return "A+"
    else:
        if x >= 80:
            return "A"
        else:
            if x >= 70:
                return "B"
            else:
                if x >= 60:
                    return "C"
                else:
                    return "F"

print( "Grade:", grade(x))
```

প্রোগ্রাম-৩ ঃ nested elif ব্যবহার করে মার্কস ইনপুট নিয়ে গ্রেড নির্ণয়ের প্রোগ্রাম ঃ

```
x = int(input(" Enter your Marks:"))
def grade(x):
    if x >= 90:
        return "A"
    elif x>= 80:
        return "A"
    elif x>= 70:
        return "B"
    elif x>= 60:
        return "C"
    else:
        return "F"
print("Grade:", grade(x))
```

প্রোগ্রাম-৪ ঃ nested elif ব্যবহার করে সময় মোতাবেক সম্ভাষণ জানানোর প্রোগ্রাম ঃ

```
x = int(input("What is the time:"))
if x < 10:
    print ("Good morning")
elif x<12:
    print ("Soon time for lunch")
elif x<18:
    print ("Good day")
elif x<22:
    print ("Good evening")
else:
    print ("Good night")
```

প্রোগ্রাম-৫ ঃ কোনো সংখ্যা জোড় না বেজোড় তা নির্ণয়ের প্রোগ্রাম ঃ

```
x = int(input('Enter the number: '))
if x%2 == 0:
    print('Even')
else:
    print('Odd')
```

প্রোগ্রাম-৬ ঃ ইনপুটকৃত সংখ্যাটি ২ অথবা ৩ দ্বারা বিভাজ্য কি না তা নির্ণয়ের প্রোগ্রাম ঃ

```
x = int(input('Enter the Number: '))
if x%2 == 0:
    if x%3 == 0:
        print('Divisible by 2 and 3')
    else:
        print('Divisible by 2 but not by 3')
elif x%3 == 0:
    print('Divisible by 3 but not by 2')
else:
    print('Not divisible by either 2 or 3')
```

প্রোগ্রাম-৭ ঃ তিন বাহুর মান দিয়ে ত্রিভুজের সম্ভাব্যতা যাচাইপূর্বক ক্ষেত্রফল নির্ণয়ের একটি প্রোগ্রাম তৈরি কর। [বাকাশিবো-২০১৭]

```
import math
a=int(input("Enter the Value of a:"))
b=int(input("Enter the Value of b:"))
c=int(input("Enter the Value of c:"))
if (a+b)>c and (b+c)>a and (c+a)>b:
        S=(a+b+c)/2
        Area=math.sqrt(S*(S-a)*(S-b)*(S-c))
        print (Area)
else:
        print("Triangle is not Possible")
```