

স্বাগতম



শিক্ষক পরিচিতি

প্রদীপ ঘোষ

ইন্সট্রাক্টর(কম্পিউটার)

খুলনা পলিটেকনিক ইন্সটিটিউট ।

পাঠদান কৃত বিষয়ের নামঃ
প্রোগ্রামিং এসেনশিয়ালস্
বিষয় কোড : ৬৬৬৩১

টি.	পি.	সি.	২	৩	৩
-----	-----	-----	---	---	---

Email:prodip87ghosh@gmail.com
Mobile Number:01754803422

আলোচ্য অধ্যায় সমূহ (সূচিপত্র)

- ❖ প্রোগ্রামিং-এর মৌলিক ধারণা ।
- ❖ পাইথন প্রোগ্রামিং-এর মৌলিক ধারণা ।
- ❖ পাইথন ভেরিয়েবল ও ডাটা টাইপ ।
- ❖ স্ট্রিং ।
- ❖ পাইথন অপারেটরস ।
- ❖ সিদ্ধান্ত গ্রহণ ।
- ❖ লুপিং ।
- ❖ লিস্ট ।
- ❖ ট্যুপল ।
- ❖ ফাংশন ।
- ❖ ফাইলের ইনপুট আউটপুট অপারেশন ।

১ম অধ্যায়

প্রোগ্রামিং-এর মৌলিক ধারণা

১.১ কম্পিউটার প্রোগ্রাম ও প্রোগ্রামিং

প্রোগ্রাম : কম্পিউটার দ্বারা সমস্যা সমাধানে ব্যবহৃত নির্দেশ সমষ্টি প্রোগ্রাম বা প্রোগ্রামিং সফটওয়্যার বলে। সাধারণ প্রোগ্রাম বলতে কম্পিউটার ইনস্ট্রাকশন, প্লানিং, ডিজাইন, উপাত্ত বিশ্লেষণ এবং পরীক্ষণকে বোঝানো হয়।

প্রোগ্রামিং:

নির্দিষ্ট প্রক্রিয়ায় কম্পিউটারের মাধ্যমে সমস্যা সমাধানের কৌশলকে প্রোগ্রামিং বলে। কম্পিউটারের পরিভাষায় প্রোগ্রামিং বলতে কোন সমস্যা সমাধানের উদ্দেশ্যে পর্যায়ক্রমে নির্দেশাবলি।

১.২ প্রোগ্রামিং -এর ব্যাখ্যা ও প্রকারভেদ

প্রোগ্রামিং ল্যাংগুয়েজ : কম্পিউটারের দ্বারা কোনা সমস্যা সমাধানের জন্য আমরা বিভিন্ন ধরনের প্রোগ্রাম বা সফটওয়্যার ব্যবহার করে যে ভাষা রচনা করা হয় সে সকল ভাষাকে প্রোগ্রামিং ল্যাংগুয়েজ বলে।

এ ধরনের কয়েকটি প্রোগ্রামিং ল্যাংগুয়েজ হচ্ছে FORTRAN, COBOL, BASIC, Pascal, Foxpro, Oracle, SQL, C, C++C#. Java, Prolog, Lisp ইত্যাদি।

(১) মেশিন ল্যাংগুয়েজ

কম্পিউটার মেশিনের নিজস্ব ভাষাকে মেশিন ভাষা বা নিম্নতরের ভাষা বলা হয়। মেশিন ল্যাংগুয়েজের প্রধান উপকরণ হচ্ছে বাইনারি ডিজিট ০ এবং ১। এটি অত্যন্ত কষ্টসাধ্য ও সময়সাপেক্ষ ল্যাংগুয়েজ। তাছাড়া অনেক কোম্পানির তৈরি কম্পিউটার এই ল্যাংগুয়েজ সাপোর্ট করে না।

ক. মেশিনের ভাষায় প্রোগ্রাম রচনার সুবিধা :

মেশিনের ভাষায় প্রোগ্রাম রচনা জটিল হলেও এ ভাষা ব্যবহারে বেশ কিছু সুবিধা রয়েছে। যেমন-

*এ ভাষা অনুশীলনের মাধ্যমে কম্পিউটারের অভ্যন্তরীণ সংগঠন সম্পর্কে ধারণা অর্জন করা সম্ভব।

*যেহেতু কম্পিউটার সরাসরি বাইনারি ভাষা বুঝতে পারে তাই এ ভাষায় তৈরি করা প্রোগ্রাম কে নির্বাহ করতে কম্পিউটারের সবচেয়ে কম সময়ের প্রয়োজন হয়।

খ. মেশিনের ভাষায় প্রোগ্রাম রচনার অসুবিধা : মেশিনের ভাষায় প্রোগ্রাম রচনা বেশ জটিল এবং সময়সাপেক্ষ বিধায় স ব্যবহারকারীরা এ ভাষায় প্রোগ্রাম লিখতে অভ্যস্ত নয়। এ ভাষা ব্যবহারে অসুবিধাসমূহ হলো:

*প্রোগ্রাম রচনা অত্যন্ত ক্লান্তিকর ও সময়সাপেক্ষ।

*এক ধরনের মেশিনের জন্য লিখিত প্রোগ্রাম অন্য ধরনের মেশিনে ব্যবহার করা যায় না।

*প্রোগ্রাম রচনার জন্য কম্পিউটারের সংগঠন সম্বন্ধে ধারণা থাকা অপরিহার্য।

(২) অ্যাসেম্বলি ল্যাংগুয়েজ : কম্পিউটার বা অনুরূপ যন্ত্রগুলো সরাসরি বুঝতে পারে এরূপ ভাষাকে নিম্নতরের ভাষা বলা হয় যাদেরকে অ্যাসেম্বল ল্যাংগুয়েজ ও বলা হয়।

১.৩ প্রোগ্রামিং ল্যাংগুয়েজের প্রজন্ম

প্রথম পর্যায় বা মেশিন ল্যাংগুয়েজ : এটি সরাসরি মেশিনের ভাষা। বাইনারি ০ এবং ১ দিয়ে মেশিনকে কমান্ড করতে হয়। মেশিন ল্যাংগুয়েজকে প্রথম পর্যায়ের ল্যাংগুয়েজ করে, বিভিন্ন ইনস্ট্রাকশনসমূহকে opcode এ পরিণত করে। প্রোগ্রামিং হলে তাকে প্রথম পর্যায়ের প্রোগ্রামিং ল্যাংগুয়েজ বলে। একে লেভেল ল্যাংগুয়েজ ও বলা হয়।

দ্বিতীয় পর্যায় : অ্যাসেম্বলি ভাষা। এটাও মেশিনের ভাষা তবে সরাসরি মেশিন ভাষা নয়। কিছুটা মানুষের বোধগম্য যেমন- Add : 12, 8. এটা দেখতে আমাদের সাধারণ গণিতের। এটা মেশিন কোডে রূপান্তর করে মেশিনকে বুঝাতে হয়। অ্যাসেম্বলি ল্যাংগুয়েজকে দ্বিতীয় পর্যায়ের ল্যাংগুয়েজ বলে। Binary opcode কে Simple symbol এর মাধ্যমে Represent করে Instruction হিসাবে ব্যবহার করে প্রোগ্রাম প্রোগ্রাম ডিজাইন করা হলে তাকে দ্বিতীয় প্রজন্মের প্রোগ্রামিং ল্যাংগুয়েজ বলে

তৃতীয় পর্যায় : উচ্চতর বা হাই লেভেল ভাষা। এটা মোটামুটি আমাদের বোধগম্য ভাষা। অর্থাৎ সাধারণ ইংরেজি ভাষার মতোই। এটা কম্পাইল করে মেশিন কোডে রূপান্তর করতে হয়। সি, জাভা, পাইথন এগুলো এই ভাষার উদাহরণ। মিদ লেভেল, ল্যাংগুয়েজ কে তৃতীয় পর্যায়ের ল্যাংগুয়েজ বলে।

চতুর্থ পর্যায় : অতি উচ্চতর বা ভেরি হাই লেভেল ভাষা। এটা কিছুটা তৃতীয় প্রজন্মের ভাষার। তবে তৃতীয় প্রজন্মের ভাষার তুলনার বেশি সহজ। আমরা সাধারণত যে ভাষা ব্যবহার করি প্রায় তার কাছাকাছি একটি ভাষা। এটা বিশেষ করে ডেটাবেজ প্রোগ্রামিং গুলোতে বেশি ব্যবহার হচ্ছে। হাই লেভেল ল্যাংগুয়েজ কে চতুর্থ প্রজন্মের ল্যাংগুয়েজ বলে।

১.৪) অনুবাদক প্রোগ্রাম

যে প্রোগ্রাম কম্পিউটারের উৎস প্রোগ্রামকে যন্ত্রভাষায় অনুবাদ করে বস্তু প্রোগ্রাম পরিণত করে সে প্রোগ্রামকে অনুবাদক প্রোগ্রাম বলে। এখানে উৎস প্রোগ্রাম, বলতে প্রথমে যে ভাষায় প্রোগ্রামটি লেখা হয় সেটিকে বুঝায়। যেমন- কিউবেসিকে একটি প্রোগ্রাম। লেখা হলা, কম্পিউটার এ প্রোগ্রামটি সরাসরি বুঝবে না, এ প্রোগ্রামটিকে অনুবাদ করে মেশিনের ভাষায় (বাইনারিতে) বুঝিয়ে দিতে হবে, এ অনুবাদের কাজে অনুবাদক প্রোগ্রাম ব্যবহৃত হয়। কম্পিউটার একমাত্র যন্ত্রভাষা বুঝতে পারে বলে অন্য ভাষায়।

ক. কম্পাইলার

খ. ইন্টারপ্রিটার

গ. অ্যাসেম্বলার

কম্পাইলার ও ইন্টারপ্রিটার এর মধ্যে পার্থক্যঃ

কম্পাইলার	ইন্টারপ্রিটার
কম্পাইলার হচ্ছে এমন এক ধরনের অনুবাদক, যা হাই লেভেল ল্যাংগুয়েজে লিখিত পুরো প্রোগ্রামকে একসাথে মেশিন ল্যাংগুয়েজে রূপান্তর করে।	যে অনুবাদক প্রোগ্রাম হাই লেভেল ল্যাংগুয়েজে লিখিত প্রোগ্রামের সোর্স কোডকে লাইন বাই লাইন বা এক লাইন করে মেশিন ল্যাংগুয়েজে রূপান্তর করে।
এটা দ্রুতিগতি সম্পন্ন।	এটা ধীরগতি সম্পন্ন।
এটা প্রোগ্রাম সমবায় অনুবাদক প্রোগ্রাম।	এটা পর্যায়ক্রমিক অনুবাদক প্রোগ্রাম।
একটি কম্পাইলার কেবলমাত্র একটি ভাষাকে অনুবাদ করতে পারে।	একটি ইন্টারপ্রিটার একাধিক ভাষাকে অনুবাদ করতে পারে।

১.৫ কম্পিউটার প্রোগ্রামের ব্যবহার

- ১) শিক্ষা সংক্রান্ত যাবতীয় কাজে
- ২) গবেষণা সংক্রান্ত কাজে
- ৩) চিকিৎসা সংক্রান্ত যাবতীয় কাজে
- ৪) শিল্পকারখানায়
- ৫) কৃষি সংক্রান্ত ও কৃষি গবেষণা সংক্রান্ত কাজে
- ৬) যানবাহন তৈরি ও নিয়ন্ত্রণ সংক্রান্ত কাজে
- ৭) আইন-শৃঙ্খলা রক্ষা ও আইন প্রয়োগ সংক্রান্ত কাজে
- ৮) বিচার ব্যবস্থাপনায়
- ৯) বড় বড় সমস্যাকে দ্রুততা ও দক্ষতার সাথে সম্পন্ন করার জন্য
- ১০) ডাটাবেজ ম্যানেজমেন্ট সংক্রান্ত কাজে
- ১১) গৃহস্থলী বিভিন্ন কাজে
- ১২) বিজ্ঞান সংক্রান্ত বা গাণিতিক সব ধরনের কাজে কম্পিউটার প্রোগ্রাম ব্যবহার করা হয় অতএব বলা যায়, কম্পিউটার প্রোগ্রামের ব্যবহার সর্ব ব্যাপি।

অ্যালগরিদমঃ

অ্যালগরিদম হলোঃ কোন সমস্যা সমাধানের উদ্দেশ্যে সুনির্দিষ্ট পর্যায়ক্রমিক ধাপসমূহের লিখিত রূপ।

অ্যালগরিদমের বৈশিষ্ট্যঃ

ধাপ ১: শুরু

ধাপ ২: ইনপুট

ধাপ ৩: সিদ্ধান্ত গ্রহণ

ধাপ ৪: প্রোসেসিং / প্রক্রিয়াকরণ

ধাপ ৫: ফলাফল প্রদর্শন / আউটপুট

ধাপ ৬: শেষ

ফ্লোচার্টঃ

ফ্লোচার্ট হলোঃ কোন সমস্যা সমাধানের উদ্দেশ্যে পর্যায়ক্রমিক ধাপ সমূহের চিত্র-ভিত্তিক রূপ।

প্রবাহচিত্র / ফ্লোচার্টঃ

১) শুরু-শেষ চিহ্নঃ

২) ইনপুট / আউটপুট চিহ্নঃ

৩) প্রসেসিং চিহ্নঃ

৪) সিদ্ধান্ত গ্রহন চিহ্নঃ

৫) নির্দেশনা চিহ্নঃ

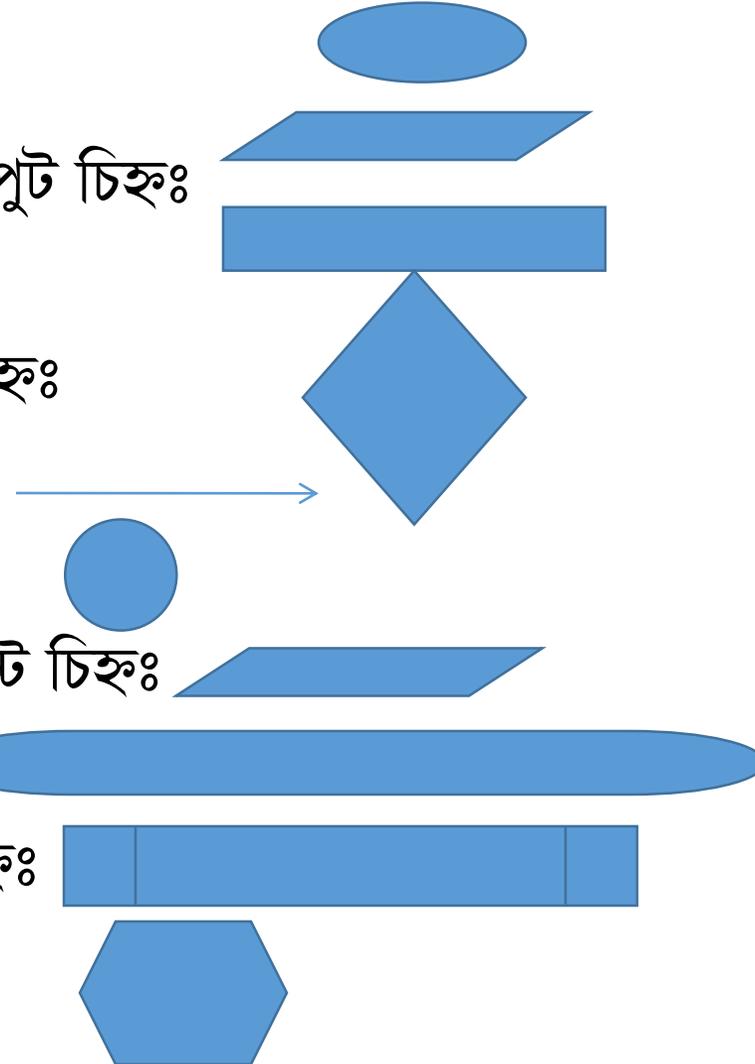
৬) সংযোগ চিহ্নঃ

৭) প্রিন্ট বা ডকুমেন্ট চিহ্নঃ

৮) প্রস্তুতি চিহ্নঃ

৯) সাবপ্রোগ্রাম চিহ্নঃ

১০) লুপ চিহ্নঃ



সাধারণ সমস্যাবলির জন্য অ্যালগরিদম ও ফ্লোচার্ট

সমস্যা-১ : দুটি সংখ্যার যোগফল নির্ণয় কর।

অ্যালগরিদম (Algorithm) :

ধাপ-১ : প্রোগ্রাম শুরু (Start)

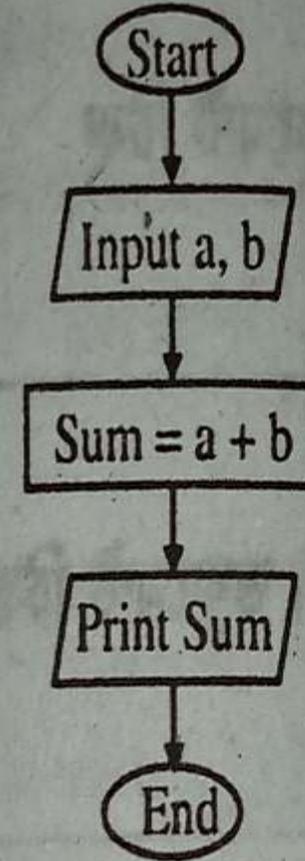
ধাপ-২ : সংখ্যা দুটি অর্থাৎ ভেরিয়ারবল a এবং b এর মান গ্রহণ (Input)

ধাপ-৩ : $Sum = a + b$ নির্ণয়

ধাপ-৪ : ফলাফল Sum প্রদর্শন (Output)

ধাপ-৫ : প্রোগ্রাম শেষ (End)।

ফ্লোচার্ট (Flowchart) :



সমস্যা-২ : আয়তক্ষেত্রের ক্ষেত্রফল নির্ণয় :

অ্যালগরিদম (Algorithm) :

ধাপ-১ : প্রোগ্রাম শুরু (Start)

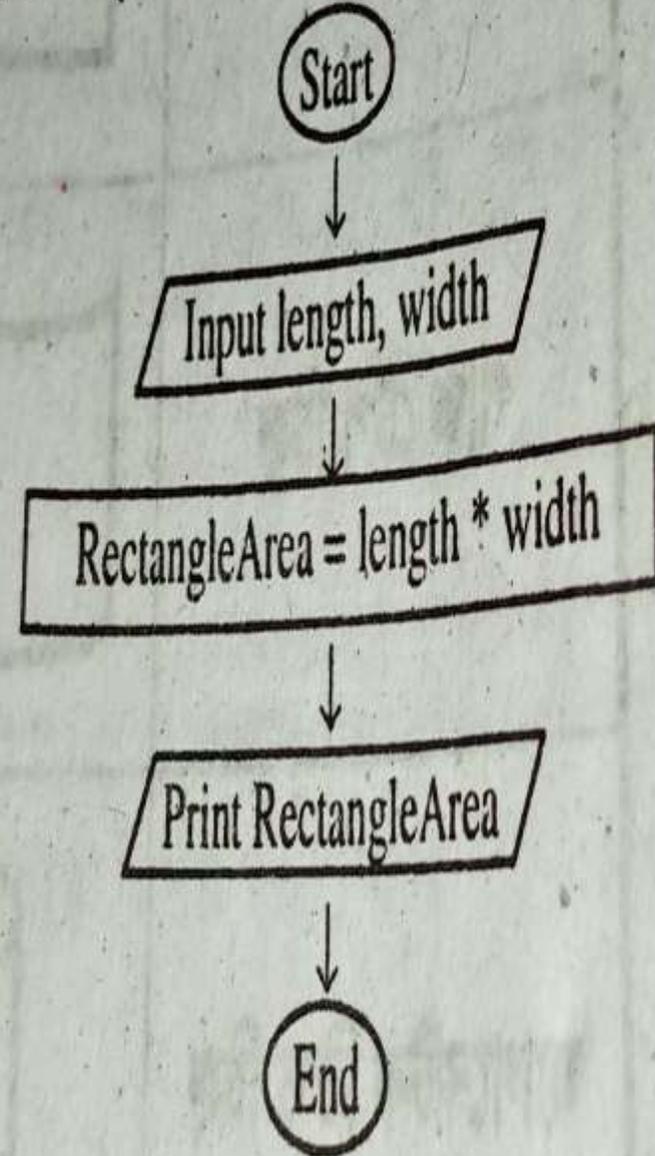
ধাপ-২ : ভেরিয়েবল Length এবং Width এর মান গ্রহণ (Input)

ধাপ-৩ : ক্ষেত্রফল নির্ণয় অর্থাৎ, $\text{RectangleArea} = \text{Length} * \text{Width}$

ধাপ-৪ : ফলাফল অর্থাৎ RectangleArea এর মান প্রদর্শন (Output)

ধাপ-৫ : প্রোগ্রাম শেষ (End)।

ফ্লোচার্ট (Flowchart) :



সমস্যা-৪ : বৃত্তের ক্ষেত্রফল নির্ণয় :

অথবা, বৃত্তের ক্ষেত্রফল নির্ণয়ের জন্য (Algorithm) লেখ।

অ্যালগরিদম (Algorithm) :

ধাপ-১ : প্রোগ্রাম শুরু (Start)

ধাপ-২ : ভেরিয়েবল radius (r) এর মান গ্রহণ (Input)

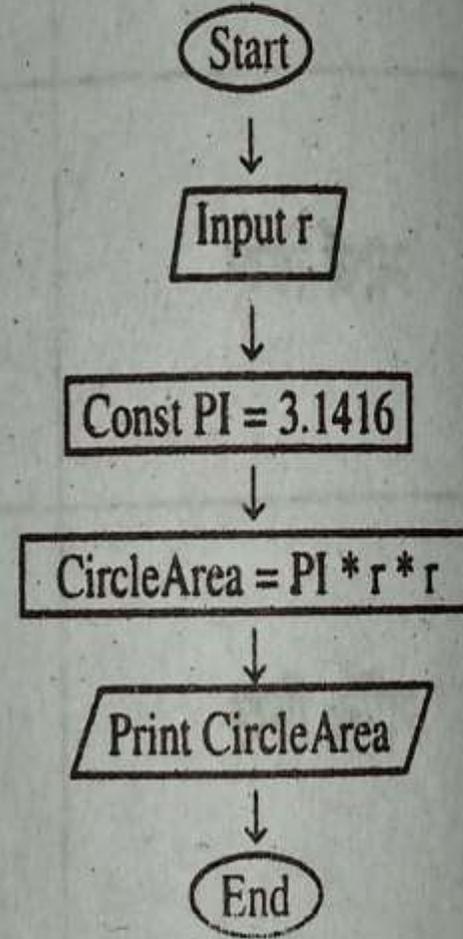
ধাপ-৩ : কনস্ট্যান্ট $PI = 3.1416$ ডিক্লারেশন

ধাপ-৪ : ক্ষেত্রফল নির্ণয় অর্থাৎ, $CircleArea = PI * r * r$

ধাপ-৫ : ফলাফল অর্থাৎ, CircleArea এর মান প্রদর্শন।

ধাপ-৬ : প্রোগ্রাম শেষ (End)।

ফ্লোচার্ট (Flowchart) :



সমস্যা-৬ : তিনটি সংখ্যার মধ্যে বৃহত্তম সংখ্যাটি নির্ণয় :

অথবা, তিনটি সংখ্যার মাঝে বৃহত্তম সংখ্যাটি বের করার Flow-chart তৈরি কর এবং C প্রোগ্রাম লেখ।

অ্যালগরিদম (Algorithm) :

ধাপ-১ : প্রোগ্রাম শুরু (Start)

ধাপ-২ : সংখ্যা তিনটি অর্থাৎ a, b, c, এর মান গ্রহণ

ধাপ-৩ : যদি $a > b$ এবং $a > c$ হয় তবে $largest = a$ হবে

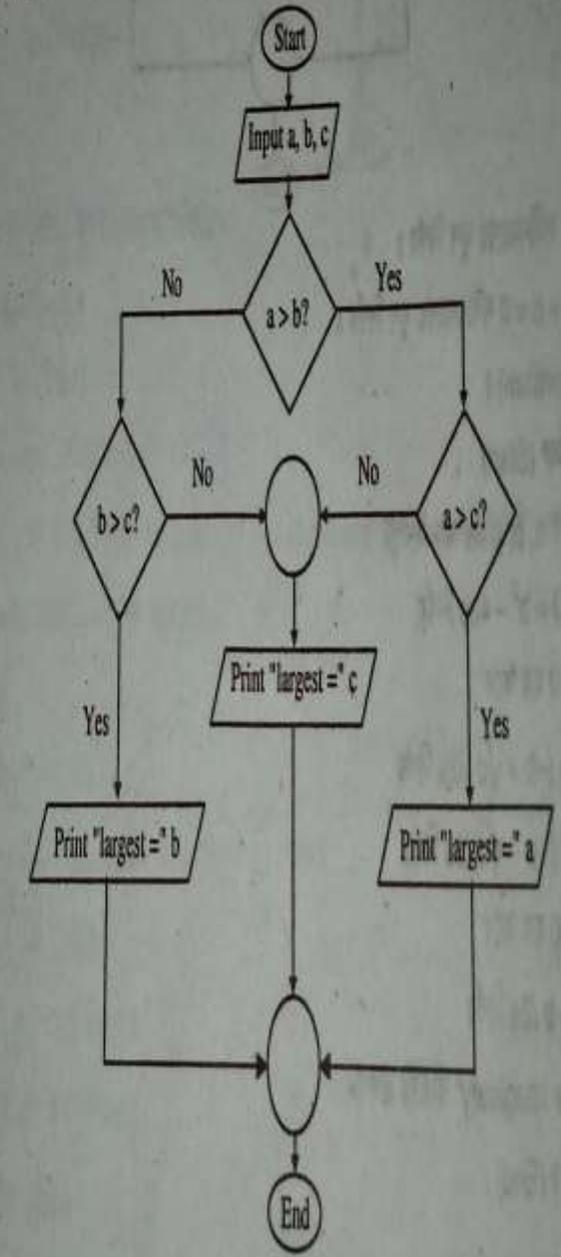
যদি $b > c$ এবং $b > a$ হয় তবে $largest = b$ হবে

না হলে $largest = c$ হবে।

ধাপ-৪ : Largest সংখ্যাটি আউটপুটে প্রদর্শন

ধাপ-৫ : প্রোগ্রাম শেষ (End)।

ফ্লোচার্ট (Flowchart) :



সমস্যা-৮ : দ্বিঘাত সমীকরণের মূল নির্ণয় :

অথবা, $ax^2 + bx + c = 0$ সমীকরণের মূল নির্ণয় :

অ্যালগরিদম (Algorithm) :

ধাপ-১ : প্রোগ্রাম শুরু (Start)

ধাপ-২ : ভেরিয়েবল a, b, c এর মান ইনপুট

ধাপ-৩ : নিশ্চয়ক $D = b^2 - 4ac$ নির্ণয়

ধাপ-৪ : যদি $D > 0$ হয় তবে

(a) $x_1 = (-b + \sqrt{d})/2a$ নির্ণয়

$x_2 = (-b - \sqrt{d})/2a$ নির্ণয়

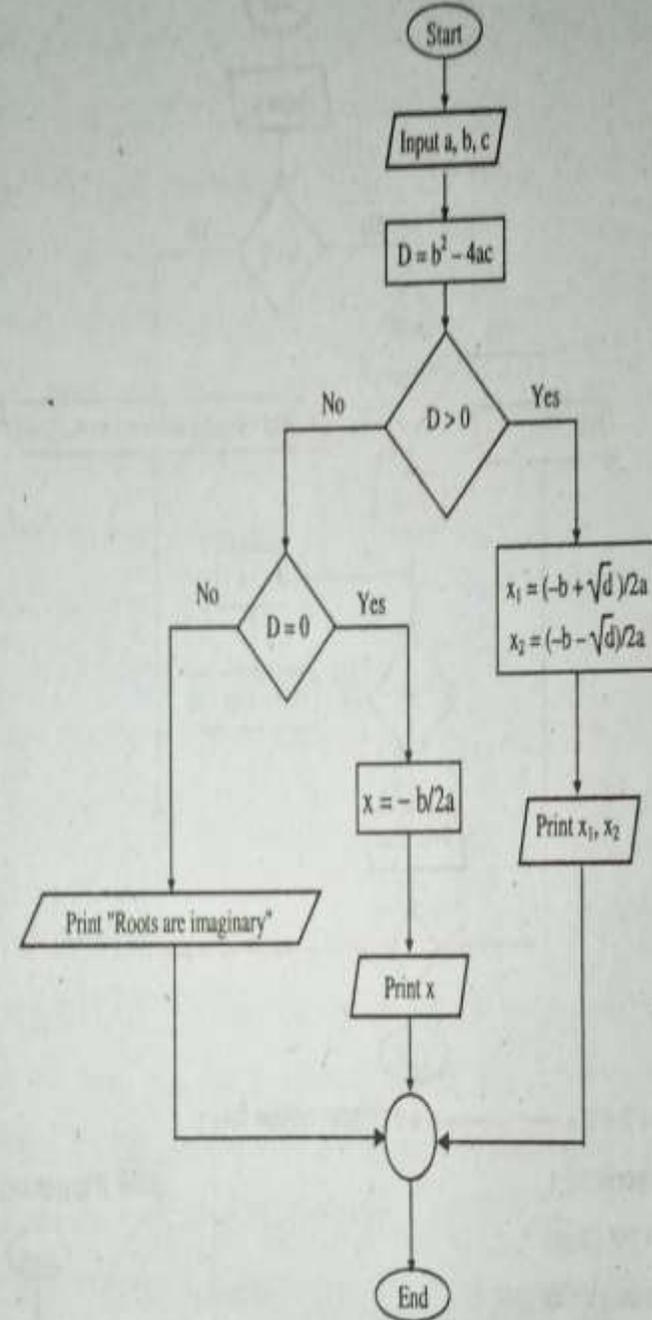
যদি $D = 0$ হয় তবে

(b) $x = -b/2a$ নির্ণয়

অন্যথায় "Roots are imaginary" বার্তাটি প্রদর্শন।

ধাপ-৫ : প্রোগ্রাম শেষ (End)।

ফ্লোচার্ট (Flowchart) :



প্রোগ্রামিং পরিকল্পনার ধাপসমূহ:

- ১) সমস্যা চিহ্নিত করা:
- ২) প্রোগ্রামের ইনপুট-আউটপুট নির্ণয়:
- ৩) অ্যালগরিদম উন্নয়ন:
- ৪) ফ্লোচার্ট উন্নয়ন:
- ৫) প্রোগ্রামিং ভাষা নির্ধারণ:
- ৬) প্রোগ্রাম রচনা:
- ৭) প্রোগ্রাম কম্পাইল করা:
- ৮) প্রোগ্রাম পরিক্ষা করণ:
- ৯) ডকুমেন্টেশন:
- ১০) ইনস্টলেশন:
- ১১) রক্ষনাবেক্ষণ:

২য় অধ্যায়

পাইথন প্রোগ্রামিং-এর মৌলিক ধারণা

২.১ পাইথনের ইতিহাস

পাইথন ১৯৯১ সালে পাইথনের কোড প্রকাশ করেন(ভার্সন ০.৯০)

পাইথন ডিজাইনের এই ভার্সনে মডুলা -৩ থেকে ধরা যায় বীজগণিতসমূহ যেমন :ক্লাস

ইনহেরিটেন্স,এক্রেপশন হ্যাডলিং ,ফাংশন ,ও প্রধান ডাটা টাইপ ইত্যাদি সংযুক্ত হয় ।

১৯৯৮ সালে পাইথনের প্রধান ফোরাম গঠিত হয় এবং পাইথনের ব্যবহারকারীদের জন্য তা মাইলফলক হিসেবে চিহ্নিত হয় ।

২.২ পাইথনের বৈশিষ্ট

*পাইথন সহজবোধ্য ।

*পাইথন সহজে পড়াডায়ায়।

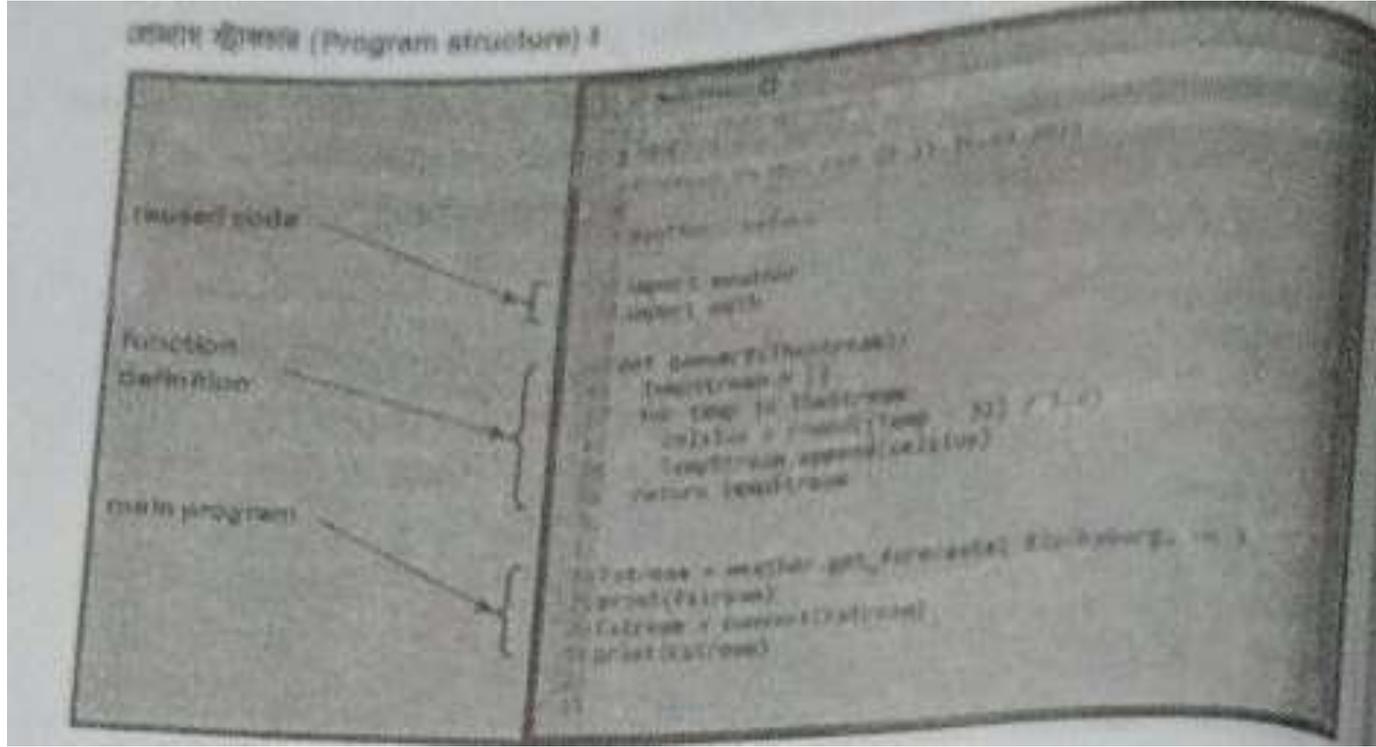
*পাইথন এর সোর্স কোড সহজে মেনটেনন করা যায়।

*পাইথনে প্রাই সব ধরণের বাণিজ্যিক ডাটাবেজ এ পাইথন এর ইন্টারফেস থাকে।

*বড় সাইজের প্রোগ্রামের স্ক্রিপ্টিং এর চেয়ে পাইথন এর ভালো কাজ হয়।

এছাড়া ও পাইথন এর কাজ আরো কিছু ফিচার আছে ।

২.৩ পাইথন প্রোগ্রামের স্ট্রাকচার



২.৪ পাইথনে ব্যবহৃত আইডেন্টিফায়ার ও কী-ওয়ার্ডসমূহ

আইডেন্টিফায়ার: পাইথনে ব্যবহৃত ভেরিয়েবল, ফাংশন বা অন্যান্য সকল অবজেক্ট আইডেন্টিফাই করার জন্য যে নাম ব্যবহার করা হয়, তাই আইডেন্টিফায়ার।

কী- ওয়ার্ড : কী ওয়ার্ড হলো কতগুলো সংরক্ষিত শব্দ, কম্পাইলারের নিকট বিশেষ অর্থ রয়েছে এবং যাদেরকে ভেরিয়েবল নাম হিসাবে ব্যবহার করা যায় না।

২.৫ পাইথনে ব্যবহৃত লাইন, ব্ল্যাংক লাইন, মার্জিন, মাল্টিলাইন স্টেটমেন্ট, ও একই লাইন স্টেটমেন্ট

- লাইন: পাইথন প্রম্পট (>>>) এর ডানে লিখিত স্টেটমেন্টকেই লাইন বলা হয়।
- ব্ল্যাংক লাইন: যখন কোন লাইনে শুধুমাত্র স্পেস, অথবা কোন কমেন্ট থাকে তাকে ব্ল্যাংক লাইন বলা হয়।
- মার্জিন: কোন একটি কোডের আগে স্পেস দেওয়া বা খালি জায়গা রাখাকে indentation বলে।
- মাল্টি লাইন স্টেটমেন্ট: যদি কোন স্টেটমেন্ট বুঝানোর জন্য একাধিক লাইনের কোড ব্লক তৈরি করার প্রয়োজন হয় তখন তাকে মাল্টি লাইন স্টেটমেন্ট বলে।
- একাধিক স্টেটমেন্ট গুহুপ: আলাদা আলাদা স্টেটমেন্ট-এর একটি মাএ গুহুপ যখন একসাথে একটি কোড ব্লক তৈরি করে তখন তাকে Suits বলে।
- একই লাইনে একাধিক স্টেটমেন্ট: পাইথনে অনেক সময় একই লাইনে একাধিক স্টেটমেন্টও ব্যবহার করা হয়।

২.৬ পাইথনের কোটেশন ও কমেন্টস

কোটেশন:

পাইথনে স্ট্রিং প্রকাশের জন্য একটি (') দুটি (") কিংবা তিনটি ('''' বা ''''') কোটেশন চিহ্ন ব্যবহার করা যায়। তবে সেই কোটেশন দিয়ে বাক্য শুরু হয়েছে, ঠিক সেটি দিয়েই শেষ করতে হবে।

কমেন্টস:

হ্যাশ চিহ্ন(#) ব্যবহার করে যদি কোন বাক্য লেখা হয়, তবে সেটি কমেন্ট হিসেবে গন্য হয়।

২.৭ কমান্ড লাইন আরগুমেন্ট

কমান্ড লাইন আরগুমেন্ট :

কোনো প্রোগ্রাম পাঠানোর প্রক্রিয়াকে কমান্ড লাইন আরগুমেন্ট বলে।

২য় অধ্যায়
পাইথন ভেরিয়েবল ও ডাটা টাইপ

৩.১ ভ্যারিয়েবল ও মান নির্ধারণ

একটি ভ্যারিয়েবলের মধ্যে কোন ভ্যাল জমা রাখার জন্য একটি সমান (=) চিহ্ন ব্যবহার করা হয়। এর আগ পর্যন্ত চ্যাপ্টারগুলোতে আমরা কোন ভ্যারিয়েবল ব্যবহার করে নি।

চলক :

সাধারণত ভেরিয়েবল বলতে চলকে বুঝায়। তবে প্রোগ্রামের ভাষায় ভেরিয়েবল হলো নির্দিষ্ট কিছু মেমোরি পরিসরের নাম। তবে প্রোগ্রামিং-এর সুবিধার্থে, প্রোগ্রামে ব্যবহৃত ডাটাসমূহ প্রোগ্রামে ব্যবহারের পূর্বে মেমোরিতে জমা করা হয়।

৩.২ ভেরিয়েবল নাম নির্ধারণের নিয়মাবলী:

ভেরিয়েবল এর নাম অবশ্যই একটি alphabetic letter (uppercase of lowercase) অথবা underscore হতে হবে। যেমন nafis,a,b,_variable লেখা যাবে ভ্যারিয়েবল হিসেবে কিন্তু 1name,@nafis,7a,%b এইভাবে লেখা যাবে না।

৩.৩ ভ্যারিয়েবল শ্রেণিবিভাগ

সি- ভ্যারিয়েবল চার প্রকার। যথা :

*Integer type variable(ইন্টিজার টাইপ ভ্যারিয়েবল)

*Character type variable(ক্যারেকটার টাইপ ভ্যারিয়েবল)

*Float type variable (ফ্লোট পয়েন্ট ভ্যারিয়েবল)

*Double type variable (ডবল টাইপ ভ্যারিয়েবল)

অবজেক্ট:

পাইথনে ডাটাকে অবজেক্ট ও বলে। পাইথনে ডকুমেন্টেশনে ডাটা বা ডাটা টাইপকে অবজেক্ট বা অবজেক্ট টাইপ ও বলা হয়। পাইথনে এই অবজেক্ট বিল্ট-ইন হিসেবে থাকে।

1. Numeric Types :

- (a) int
- (b) float
- (c) complex

1. Iterator Type :

- (a) generator

1. Sequence Types :

- (a) list
- (b) tuples
- (c) range

1. Text Sequence Types:

- (a) strings

1. Set Types :

- (a) set
- (b) frozen set

6. Binary Sequence

- (a) bytes
- (b) bytearray
- (c) memoryview

7. Mapping Types :

- (a) dictionary

8. Others :

- (a) modules
- (b) classes
- (c) functions
- (d) methods
- (e) code object
- (f) the type object
- (g) the ellipses objects
- (h) boolean values
- (i) the null object

ডাটা টাইপ নিশ্চিতকরণ:

প্রোগ্রামে কোন ধরনের ডাটা টাইপ ব্যবহার করা হয়েছে তা জানার জন্য পাইথনে `type ()` ফাংশন ব্যবহার করা হয়। নিম্নে `type ()` ফাংশন ব্যবহার উল্লেখ করা হলো:

উদাহরন:

```
>>> a = 1
>>> type(a)
<class 'int'>
>>> b = 3.1416
>>> type(b)
<class 'float'>
>>> c = "Hello world"
>>> type(c)
<class 'str'>
>>> d = True
>>> type(d)
<class 'bool'>
```

উপরোক্ত উদাহরনে দেখা যায় যে, `a` হলো ইন্টিজার টাইপের, `b` হলো ফ্লোটিং পয়েন্ট টাইপের, `c` হলো স্ট্রিং টাইপের, `d` হলো `bool` টাইপের, যা `type ()` ফাংশনের মাধ্যমে জানতে পারলাম।

৩.৪ : ডাটা টাইপ কনভার্সন:

এক টাইপ ভেরিয়েবলকে অন্য টাইপ ভেরিয়েবলে কনভার্ট পকরিয়াকে ডাটা টাইপ কনভার্সন বলে। একে টাইপ কাস্টিং ও বলে। পাইথনে এ ধরনের ডাটা টাইপকে অন্য ধরনের ডাটা টাইপে পরিবর্তন করার জন্য বেশ কিছু বিল্ট-ইন ফাংশন রয়েছে। নিম্নে পাইথনে ব্যবহৃত টাইপ কনভার্সন ফাংশনসূহ বরননা করা হল।

ফাংশন

বননা

`int(x[,base])`

স্ট্রিং x কে পূনসংখ্যায় পরিবর্তন করার জন্য এ ফাংশনটি ব্যবহৃত হয়।

`long(x[,base])`

স্ট্রিং x কে একটি দীঘ পূনসংখ্যায় পরিবর্তন করার জন্য এ ফাংশনটি ব্যবহৃত হয়।

`float(x)`

ভেরিয়েবল x কে দশমিক বিশিষ্ট সংখ্যায় (floating-point)রূপান্তর করার জন্য এ ফাংশনটি ব্যবহৃত হয়।

`chr(x)`

ইন্টিজার টাইপ ডাটাকে ক্যারেক্টার টাইপ ডাটার রূপান্তর করার জন্য এ ফাংশনটি ব্যবহৃত হয়।

`unichr(x)`

ইন্টিজার টাইপ ডাটাকে ইউনিকোড ক্যারেক্টার টাইপ ডাটায় রূপান্তর করার জন্য এ ফাংশনটি ব্যবহৃত হয়।

৩.৫: টাইপ কনভার্সন করে প্রোগ্রাম:

উদাহরন: ফারেনহাইট তাপমাত্রাকে সেন্টিগ্রেডে রূপান্তরের একটি প্রোগ্রাম লেখ।

```
fahrenheit = float(input("please enter the fahrenheit temperature:"))  
celsius = (fahrenheit - 32)/1.8  
print('%0.1f degree fahrenheit is=% 0.1f degree celsius' % (fahrenheit, celsius))
```

৪র্থ অধ্যায় স্ট্রিং

৪.১ স্ট্রিং এর মান প্রদর্শন ও স্ট্রিং আপডেটিং

স্ট্রিং এর মান এক্সেসিং :

পাইথন স্ট্রিং ফরম্যাট: একটা স্ট্রিং এর ভেতরে একটা ভ্যারি়াবল প্রিন্ট করার জন্য **String Formates** ব্যবহার করা হয়, যেমন আমাদের a ও b নামে দুইটি ভ্যারি়েবল আছে। a ও b মান আমরা স্ট্রিং-এর ভেতরে প্রিন্ট করব। এভাবে **Value of a=10. value of b=50**

এখন আমরা চাচ্ছি ওয়ার্ডগুলোকে আমার এক সাথে করতে। তার জন্য:

- 1.s="This is a string"
2. words= s.split ()
- 3.forw in words:
4. print (w)
5. prtnt (“ .join (words))

৪.২ এসকেপ ক্যারেঞ্জার

আউটপুটকে দৃষ্টিনন্দন , পরিচ্ছন্ন ও স্পষ্টভাবে জ্বিনে প্রদর্শনের জন্য ব্যাকস্লাশ সহযোগে কতগুলো বিশেষ ক্যারেঞ্জার ব্যবহার করা হ । এদেরকে ব্যাকস্লাশ ক্যারেঞ্জার কনস্ট্যান্ট বা এসকেপ ক্যারেঞ্জার বলে । নিউ লাইন ক্যারেঞ্জার (n) ব্যাকস্ল্যাস ক্যারেঞ্জার (/) ট্যাব, ইউনিকোড ক্যারেঞ্জার এদেরকেও এক্কেপ করেঞ্জার এদেরকেও এক্সপ কওে স্ট্রিং-এর মধ্যে ব্যবহার করতে হয় ।

পাইথনে নিউলাইন ক্যারেঞ্জারকে ম্যানুয়ালি লেকার দরকার পওে না যদি একাধিক লাইন সম্বলিত সেই স্টেং বা বাক্যকে তিনটি কওে কোটেশন এর মধ্যে ডিফাইন করা হয় ।

উপওে, দুই লাইন ওয়ালা একটি স্ট্রিংকে ইনপুট হিসেবে দিয়েছি এবং এর মধ্যে যেখানে নতুন লাইন দরকার সেখানে পাইথন স্বয়ংক্রিয়ভাবে (/n) ক্যারেঞ্জার বসিয়ে দিয়েছে ।

নিম্নে পাইথনে ব্যবহৃত এসকেপ ক্যারেঞ্জার সমূহ ও তাদের ব্যবহার উল্লেখ করা হল । যেমন-

এসকেপ ক্যারেঞ্জার

আউটপুটকে দৃষ্টি নন্দন, পরিচ্ছন্ন ও স্পষ্টভাবে স্ক্রিনে প্রদর্শনের জন্য ব্যাকস্লাশ সহযোগে কতকগুলো বিশেষ ক্যারেঞ্জার ব্যবহার করা হয়। এদেরকে ব্যাকস্লাশ ক্যারেঞ্জার বা এসকেপ ক্যারেঞ্জার বলে।

নিউ লাইন ক্যারেঞ্জার (`\n`), ব্যাকস্লাশ ক্যারেঞ্জার (`\`) ট্যাব, ইউনিকোড ক্যারেঞ্জার এদেরকেও এক্সেপ করে স্ট্রিং এর মধ্যে ব্যবহার করতে হয়।

পাইথনের নিউলাইন ক্যারেঞ্জারকে ম্যানুয়ালি লেখার দরকার পড়ে না যদি একাধিক লাইন সম্বলিত সেই স্ট্রিং বা বাক্যকে তিনটি করে কোটেশন এর মধ্যে ডিফাএন করা হয় । নিচের উদাহরণটি দেখি,

```
>>> " " " Me : Hi, There!
```

```
... She : Yes, please! " " "
```

```
`Me : Hi, There!/\nShe : Yes, please! `
```

উপরে, দুই লাইন ওয়ালা একটি স্ট্রিংকে ইনপুট হিসেবে দিয়েছি এবং আউটপুটে দেখা হচ্ছে সে স্ট্রিং এর মধ্যে যেখানে নতুন লাইন দরকার সেখানে পাইথন স্বয়ংক্রিয় ভাবে `\n` ক্যারেক্টারবিসিয়ে দিয়েছে।

নিম্নে পাইথনে ব্যবহৃত এসকেপ ক্যারেক্টার সমূহ ও তাদের ব্যবহার উল্লেখ করা হল । যেমন :

ক্রমিক	এসকেপ ক্যারেক্টার	ব্যবহার
১	\\	একটা ব্যাকস্ল্যাশ () প্রদর্শনের জন্য
২	\'	সিঙ্গেল কোট () প্রদর্শনের জন্য।
৩	\"	ডাবল কোট () প্রদর্শনের জন্য।
৪	\a	বেল বা এলার্ট (সতর্ক সংকেত) শোনানোর জন্য।
৫	\b	ব্যাকস্পেস দেয়ার জন্য।
৬	\s	স্পেস দেয়ার জন্য।
৭	\e	escape করার জন্য।
৮	\f	ফর্মফিড
৯	\n	আউটপুটকে লাইন নতুন লাইনে প্রদর্শনের জন্য বা লাইন ব্রেক দেওয়ার জন্য।
১০	\N{name}	ইউনিকোড ক্যারেক্টারের নাম
১১	\rASCII	ক্যারিজ রিটার্ন (ম্যাক ওস এক্স এ নিউ ক্যারেক্টার)
১২	\t	আউটপুট ডান দিকে ট্যাব দূরত্বে প্রদর্শনের জন্য।
১৩	\uxxxx	১৬ বিট হেক্সাডেসিম্যাল ভ্যালু সম্বলিত ইউনিকোড ক্যারেক্টার প্রদর্শনের জন্য।
১৪	\Uxxxxxxxx	৩২ বিট হেক্সাডেসিম্যাল ভ্যালু বিশিষ্ট ইউনিকোড ক্যারেক্টার প্রদর্শনের জন্য।
১৫	\v	আউটপুট ভার্টিক্যাল ট্যাব দূরত্বে প্রদর্শনের জন্য ।
১৬	\ooo	'000' অক্টাল ভ্যালু বিশিষ্ট ক্যারেক্টার প্রদর্শনের জন্য
১৭	\xhh	'hh'হেক্সাডেসিমেল ভ্যালুওয়ালা ক্যারেক্টার প্রদর্শনের জন্য।

৪.৩ স্ট্রিং-এর বিশেষ ও ফরমেটিং অপারেটরঃ

স্ট্রিং নিয়ে কাজ করার জন্য পাইথনে অনেকগুলো বিল্টইন মিথোড রয়েছে। যেগুলো আমরা কেউ ফাংশন বলি। তো পুরো লিস্ট পাওয়া যাবে এখানে।

<http://docs.python.org/library/srctypes.html>

এগুলো একবার দেখে নেওয়া দরকার। মুখস্ত করার কিছুই নেই। শুধু এশবার চোখ বুলিয়ে নিলে হবে। কী কী করা যায় তা জানতে হবে। এবং যখন আমরা কোড লিখবো তখন ডেঁটা কাজে লাগবে তার কোড দেখে আমরা লিখতে পারবো। সব প্রগ্রামে স্ট্রিং নিয়ে অনেক কাজ করতে হয়। গেম হোক, সফটওয়্যার হোক, মোবাইল গেম অ্যাপ। তাই স্ট্রিং নিয়ে ভালো ধারণা থাকা দরকার।

এখানে আমরা দুই একটা কোড লিখবো। যেমন একটা স্ট্রিং এর সবগুলো কারেকটারকে ছোট হাতের কনভার্ট করতে।

Step No: 1 `s="This is a string"`

Step No: 2 `print(s.lower())`

উপরে আমরা একটি স্ট্রিং নিয়েছি। এর দুইটা কয়েকটা বড় হাতের ছিল, এখন পরে তা lower কণ্ডে প্রিন্ট করাতে সবগুলোই ছোট হাতের অক্ষর হাতে পরিণত হয়েছে।

আবার Uper করলে সবগুলো আবার বড় হাতের হয়ে যাবে।

Step No: 1 `s="This is a string"`

Step No: 2 `print(s.lower())`

পাইথনে স্ট্রিং এর জন্য বহুল ব্যবহৃত অপারেটর সমূহ হলোঃ

- কনক্যাটেশন অপারেটর
- রিপিটিশন অপারেটর
- মেম্বারশীপ অপারেটর

কনক্যাটেশন অপারেটর

দুইটি স্ট্রিং নিয়ে সংযুক্ত করার পদ্ধতি হচ্ছে স্ট্রিং কনক্যাটেশন অপারেটর। এর জন্য (+) অপারেটর ব্যবহার করা হয়।

যেমনঃ

>>>“Bangladesh” + “To” “Bangladesh”

>>>‘Wellcome To Bangladesh’

>>>

দুইটি স্ট্রিং-এর মধ্যে স্পেস দেওয়ার নিয়মঃ

যেমনঃ

>>>“Wellcome” + Desh’

>>>‘Bangladesh’

>>> print (“Akhi” + “&” + “Anu”)

Akhi & Anu

>>>

রিপিটিশন অপারেটরঃ একই স্ট্রিং এর একাধিক কপির পুনরাবৃত্তির পদ্ধতি হচ্ছে স্ট্রিং রিপিটিশন। এর জন্য(*)

অপারেটর ব্যবহার করা হয়।

যেমনঃ

```
>>>'Desh'*3
```

```
>>>'DeshDeshDesh'
```

```
>>>4*'5'
```

```
'5555'
```

```
>>>
```

মেম্বারশীপ অপারেটরঃ সিকুয়েনশিয়াল ডাটা টাইপের ক্ষেত্রে (সদস্য) পরীক্ষা করার জন্য অথ্যাৎ কেউ কোন নির্দিষ্ট গ্রুপের

মেম্বার কিনা নির্ণয়ের জন্য মেম্বারশীপ অপারেটর ব্যবহার করা হয়। মেম্বারশীপ অপারেটর হিসেবে ইন এবং নট ইন দুটি অপারেটর ব্যবহার করা হয়।

In অপারেটরঃ যদি কোন একটি ক্যারেক্টার কোন নির্দিষ্ট স্ট্রিং-এর অন্তর্ভুক্ত হয় তবে রিটার্ন আসবে সত্য আর যদি অন্তর্ভুক্ত না হয় তবে মিথ্যা আসবে।

যেমনঃ

```
>>>
```

```
>>>'m' in 'akhi'
```

```
True
```

```
>>>'m' in 'akhi'
```

```
False
```

Not In অপারেটরঃ যদি কোন একটি ক্যারেক্টার কোন নির্দিষ্ট স্ট্রিং-এর অন্তর্ভুক্ত না হয় তবে রিটার্ন আসবে সত্য আর যদি অন্তর্ভুক্ত হয় তবে মিথ্যা আসবে ।

যেমনঃ

```
>>>
```

```
>>>'m' not in 'akhi'
```

```
False
```

```
>>>'d' not in 'akhi'
```

```
True
```

```
>>>'h' not in 'akhi'
```

```
False
```

```
>>>'p' not in 'akhi'
```

```
True
```

স্ট্রিং ফরম্যাট অপারেটরঃ পাইথনে স্ট্রিং ফরমেটিং অপারেটর হিসেবে %s অপারেটর ব্যবহার করা হয় ।

যেমনঃ

```
>>>
```

```
>>>x= "I have only %d %s "% (1000, 'Taka')
```

```
>>> print(x)
```

```
>>>
```

স্ট্রিং ফরম্যাট মেথডঃ একটি স্ট্রিং এর ভিতরে ভেরিয়েবল প্রিন্ট করার জন্য Format Function ব্যবহার করা হয়। মূলত নন স্ট্রিং ডাটার সাথে স্ট্রিং টাইপের ডাটাকে যুক্ত করে সুন্দর স্ট্রিং আউটপুট তৈরি করতে Format Method ব্যবহার করা হয়।

যেমনঃ

```
>>> a=10
>>> b=10
>>> print (Value of a={ }, value of b= { } format (a,b))
Value of a=10, Value of b=20
>>>
```

ইনডেক্সিংঃ পাইথনের মাধ্যমে স্ট্রিং এর প্রতিটি উপাদানের ইনডেক্সিং বের করা যায়। এর স্ট্রিং এর প্রতিটি ক্যারেক্টারের/ উপাদানের একটি নির্দিষ্ট পজিশন ভালু অনুসারে স্ট্রিং এর প্রতিটি উপাদানকে উপস্থাপনই ইনডেক্সিং এর কাজ। যদি নামের এই স্ট্রিং এর ইনডেক্সিং করতে চাই, তবে আগে নাম লিখতে হবে তারপর স্কয়ার ব্রাকেটের ভিতর নির্দিষ্ট নাম্বার দিতে হবে।

যেমনঃ

```
>>> Name= "Bangladesh"
>>>
>>> Name[0]
```

এখানে ভেরিয়েবল ডিক্লারেশনের পর ইনডেক্স হিসেবে ০ লেখায় আউটপুটে ০ পজিশনে বি কে প্রদর্শন করেছে। নেগেটিভ নাম্বারেও ইনডেক্সিং করা যায় অর্থাৎ উল্টো দিক থেকে ইনডেক্সিং হবে। যেমনঃ (-1) এ কোন ক্যারেক্টারের অবস্থান যদি দেখার জন্য কোডিং হবে নিম্নরূপ।

```
>>> Name= "Bangladesh"
>>>
>>> Name[-1]
```

Slicing: একটি স্ট্রিং এর ইনডেক্স নাম্বার অনুযায়ী একটি রেঞ্জ থাকে এবং সেই রেঞ্জকে বিভিন্ন ভাগে বাগ করে স্ট্রিং এর বিভিন্ন ক্যারেক্টারকে তাতেও নিজেদেও অবস্থান অনুযায়ী উপস্থাপন করাই স্লাইশিং।

যেমনঃ

```
>>> x= "Murad Hossain"
```

```
>>>x[0:1]
```

```
'M'
```

```
>>>x[0:2]
```

```
'Mu'
```

```
>>>x[0:3]
```

```
'Mur'
```

```
>>>x[0:4]
```

```
'Mura'
```

```
>>>x[0:5]
```

```
'Murad;
```

```
>>>x[:]
```

```
'Murad Hossain'
```

```
>>>x[0:]
```

```
'Murad Hossain'
```

```
>>>[3:]
```

```
'sain'
```

```
>>>[-3:]
```

```
'sain'
```

স্ট্রিং ফরম্যাটিং বিল্ট ইন মেথডঃ পাইথনে স্ট্রিং নিয়ে কাজ করার জন্য বেশ কিছু উল্লেখযোগ্য বিল্ট ইন ফ্যাংশন রয়েছে ।

যেমনঃ

- ✓ Len
- ✓ Join
- ✓ Replace
- ✓ Strip
- ✓ Split
- ✓ Captitalize
- ✓ Upper
- ✓ Lower
- ✓ Casfold
- ✓ Count
- ✓ find

লিন মেথডঃ সাধারনত স্ট্রিং এর মধ্যে কতকগুলো ক্রারেষ্ঠার বা সংখ্যা আছে তা নির্নয় করার জন্য লিন ফ্যাংশন () ব্যবহার করা হয় ।

লিন এর দেঘ্য এর সংক্ষিপ্তরূপ । এটি পাইথনের একটি Built- In- Function যা কিনা লাইব্রেরি থাকে ।

যেমনঃ

```
>>> x = 'Murad'
>>> len(x)
6
>>> y = 'Mahfuza Rahman'
>>> len(y)
>>>12
```

৪.৪ ট্রিপল কোডস ও ইউনিকোড স্ট্রিং

ট্রিপল কোডসঃ পাইথনে একাধিক লাইনে স্টেটমেন্ট লেখার জন্য ট্রিপল কোটস ব্যবহার করা হয়। ট্রিপল কোড বুঝাতে তিনটি সিঙ্গেল কোটেশন চিহ্ন (""") অথবা তিনটি ডাবল কোটেশন চিহ্ন ব্যবহার করা হয়।

উদাহরণঃ

```
>>> Jatiya Songit-'Amar sonar Bangla ami tomay valobashi, chiro
      din tomar aksh tomar batash tomar prane bajai          bashi""""
```

```
>>>print(Jatiya Songit-)
```

```
      Amar sonar Bangla ami tomay valobashi, chiro
      din tomar aksh tomar batash tomar prane bajai          bashi""""
```

ইউনিকোড স্ট্রিংঃ সাধারণত পাইথনে যে কোন স্ট্রিং ৪ বিট ফরম্যাটে স্টোর হয়। নরমাল স্ট্রিং ১৬ বিট আকারে স্টোর করার জন্য প্রিফিক্স হিসাবে প্রিন্ট কমান্ডের সঙ্গে ইউ ব্যবহৃত হয়।

উদাহরণঃ

```
>>> print (u "Wellcome to KPI')
```

```
Wellcome to KPI
```

```
>>>
```

৪.৫ স্ট্রিং ব্যবহার করে কতিপয় সাধারণ প্রোগ্রাম

সমস্যা-১ কোনো স্ট্রিং ইনপুট নিয়ে তার সকল স্পেস মুছে দেওয়ার প্রোগ্রাম।

প্রোগ্রামঃ

```
While True:
```

```
    print('Enter 'x' for exit)
    string = input('Enter any string:')
    if string == 'x':
        break
```

```
else:
```

```
    new_string = string.replace(" ", "")
    print("\New string after removing all spaces:")
    print(new_string, "\n")
```

আউটপুটঃ

```
Enter 'x' for exit.
```

```
Enter any string" Murad Hossain
```

```
New string after removing all spaces:
```

```
Muradhossain
```

সমস্যা-২ দুটো স্ট্রিং ইনপুট নিয়ে তাতেও মধ্যে তুলনা করার প্রোগ্রাম।

প্রোগ্রামঃ

While True:

```
print('Enter 'x' for exit)
string1 = input('Enter string1:')
string2 = input('Enter string2:')
if string1 == 'x':
    break
```

else:

```
if string1 == string2:
    print("Both strings are equal.\n")
else:
    print("string are not equal.\n")
```

আউটপুটঃ

Enter 'x' for exit.

Enter any string1" M.D Murad Hossain

Enter any string1" M D Murad Hossain

String are not Not Equal

৫ম অধ্যায়

পাইথন অপারেটরস

৫.১ পাইথন অপারেটর ও অপারেটরের শ্রেণিবিভাগ

অপারেটর : অপারেটর হলো এক্সপ্রেশনে ব্যবহৃত বিশেষ চিহ্ন বা শব্দ বা কম্পালারকে কোন বিশেষ গাণিতিক ,তুলনামূলক বা যৌক্তিক কার্য সম্পাদনে নির্দেশ প্রদান করে ।

অপারেটরের শ্রেণিবিভাগ: পাইথনে অপারেটরগুলোকে তাতেও প্রকৃতি ও কার্য আনুযায়ী আট ভাগে ভাগ করা হয়েছে,যথা-

- গাণিতিক অপারেটর
- তুলনামূলক অপারেটর
- যৌক্তিক অপারেটর
- অ্যাসাইনমেন্ট অপারেটর
- বুলিয়ান অপারেটর
- মেম্বরশিপ অপারেটর
- আইডেন্টিটি অপারেটর
- বিটওয়াইজ অপারেটর

৫.২ গাণিতিক, তুলনামূলক, যৌক্তিক অপারেটর

গাণিতিক অপারেটর: বিভিন্ন ধরনের গাণিতিক কার্যাদি সম্পন্ন করার জন্য যে-সব অপারেটর ব্যবহার করা হয়, তাদেরকে গাণিতিক অপারেটর বলে । যেমন : +,-,*,/,%,**,//ইত্যাদি

তুলনামূলক অপারেটর : দুই বা ততোধিক অবজেক্টের মধ্যে তুলনা করার জন্য তুলনামূলক অপারেটর ব্যবহার করা হয়। এদেরকে রিলেশনাল অপারেটরও বলা হয়। যেমন ==, !=, >, >=, <, <= ইত্যাদি।

যৌক্তিক অপারেটর : যে সব অপারেটর যৌক্তিক কার্যাদি সম্পন্ন করার জন্য ব্যবহার করা হয় তাদেরকে যৌক্তিক অপারেটর বলে। যেমন and, or, not.

৫.৩ এসাইনমেন্ট, বিটওয়াইজ, মেম্বরশিপ, আইডেন্টিটি ও বুলিয়ান অপারেটর

এসাইনমেন্ট অপারেটর: কোন এক্সপ্রেশন বা ভেরিয়েবলের মানকে অপর কোন ভেরিয়েবলের মান হিসাবে নির্ধারণ করার জন্য যে ধরনের অপারেটর (=) ব্যবহার করা হয়, তাকে এসাইনমেন্ট অপারেটর বলে।

বিটওয়াইজ অপারেটর: বাইনারি ডাটা অর্থাৎ বিট বা বাইট নিয়ে বিভিন্ন রকমের লজিক্যাল অপারেশন সম্পন্ন করার জন্য যে সব অপারেটর ব্যবহার করা হয় তাদেরকে বিটওয়াইজ অপারেটর বলে। যেমন &, |, ^, ~, <<, >> ইত্যাদি।

মেম্বরশিপ অপারেটর: কোন একটি ভেরিয়েবল একটি বিশেষ লিস্ট, টাপল, কিংবা স্ট্রিং এর মেম্বর কি না তা নির্ণয়ের জন্য যে সকল অপারেটর ব্যবহার করা হয়, তাদেরকে মেম্বরশিপ অপারেটর বলে। পাইথনে দুই ধরনের মেম্বরশিপ অপারেটর বিদ্যমান। যথা - in এবং not in.

আইডেন্টিটি অপারেটর: দুইটি অবজেক্টের মেমরি লোকেশনের মধ্যে তুলনা করার জন্য অর্থাৎ দুইটি ভেরিয়েবলের একই মেমরি লোকেশন কি না তা নির্ণয়ের জন্য যে সব অপারেটর ব্যবহার হয়, তাদেরকে আইডেন্টিটি অপারেটর বলে। পাইথনে দুই ধরনের আইডেন্টিটি অপারেটর বিদ্যমান। যথা - is এবং is not.

বুলিয়ান অপারেটর: বুলিয়ান অপারেটর হলো সে সব অপারেটর, যার সত্য অথবা মিথ্যা মান রিটার্ন করে। পাইথনে বুলিয়ান ভ্যালু হলো দুইটি

- True (T must be Capital Letter)
- False (F must be Capital Letter)

৫.৪ অপারেটর প্রেসিডেন্স ও অ্যাসোসিয়েটিভিটি

অপারেটর প্রেসিডেন্স: কোন এক্সপ্রেশনে একাধিক অপারেটর ব্যবহৃত হলে কম্পাইলার যে অনুক্রমের ভিত্তিতে অপারেটরের অগ্রগণ্যতা নির্ধারণ করে, তাকে অপারেটরের প্রেসিডেন্স বলে। যেমন প্রথমে ব্রাকেটের কাজ, তারপর পাওয়ার/এক্সপোনেন্ট, অঃপর গুণ ও ভাগ এবং শেষে যোগ ও বিয়োগ।

অপারেটর অ্যাসোসিয়েটিভিটি : কোন এক্সপ্রেশনে যদি একাধিক অপারেটর ব্যবহৃত হয় এবং এসকল অপারেটরের প্রেসিডেন্স যদি সমান হয় তবে এদের কাজ ডান দিক থেকে শুরু হবে না বাম দিকে শুরু হবে তা যে বৈশিষ্ট্যেও মাধ্যমে নির্ধারিত হয়, তাকে অপারেটর অ্যাসোসিয়েটিভিটি বলে।

৬ষ্ঠ অধ্যায় সিদ্ধান্ত গ্রহণ

৬.১ : কন্ডিশনাল ও আনকন্ডিশনাল ব্রাঞ্চিং ফ্লো বর্ণনা

প্রোগ্রামে যদি কোনো Statement শর্ত সাপেক্ষে বা অপর কোনো Statement এর ফলাফলের উপর নির্ভর করে সম্পন্ন করতে হয় বা Execution বন্ধ রাখতে হয় বা প্রোগ্রামের নিয়ন্ত্রণ অন্য কোনো Statement এ স্থানান্তর করতে হয় তখন প্রোগ্রামের এ ধরনের ফ্লোকে ব্রাঞ্চিং (Branching) ফ্লো বলে।

ক) কন্ডিশনাল ব্রাঞ্চিং ফ্লো (Conditional Branching Flow)

খ) আনকন্ডিশনাল ব্রাঞ্চিং ফ্লো (Unconditional Branching Flow)

কন্ডিশনাল ব্রাঞ্চিং ফ্লো (Conditional Branching Flow) :

যখন কোনো শর্ত সাপেক্ষে প্রোগ্রামের এক বা একাধিক স্টেটমেন্টকে প্রোগ্রামের নির্বাহ (Execution) হতে বিরত রাখা হবে বা এক অংশকে বাদ নিয়ে অন্য অংশ নির্বাহ করা হবে তখন প্রোগ্রামের এ ধরনের প্রবাহ বা ফ্লোকে কন্ডিশনাল ব্রাঞ্চিং ফ্লো বলে। নিম্নে ফ্লোচার্টেও মাধ্যমে তা বিস্তারিত বুঝানো হলো:

আনকন্ডিশনাল প্রোগ্রাম ফ্লো (Unconditional Branching Flow) : কোনো প্রোগ্রাম যদি Unconditional Statement এর প্রভাবে নির্বাহ হয় তবে তাকে Unconditional program flow বলে। Unconditional Statement এর প্রভাবে প্রোগ্রামের কোন অংশ বারবার আবর্তিত হতে থাকে কিংবা নির্দিষ্ট কোন Statement এ jump করে।

৬.২ if স্টেটমেন্ট, if else স্টেটমেন্ট ও else স্টেটমেন্ট

কন্ডিশনাল ব্রাঞ্চিং স্টেটমেন্ট (Conditional Branching statement) :

ডসদ্বান্তমূলক কাজ যেমন শর্তেও ভিত্তিতে ছাত্রের জিপিএ নির্ধারণ ইত্যাদি নির্বাহের জন্য কন্ডিশনাল ব্রাঞ্চিং স্টেটমেন্টগুলো খুবই কার্যকর। কাজের জটিলতা (Complexity) অনুযায়ী সফলভাবে প্রোগ্রাম নির্বাহের জন্য নিম্নলিখিত ব্রাঞ্চিং স্টেটমেন্টগুলো সচরাচর ব্যবহার হয়ে থাকে।

- i) If statement
- ii) if-else statement
- iii) Else if statement
- iv) switch statement

আনকন্ডিশনাল ব্রাঞ্চিং স্টেটমেন্ট :

কোন সিদ্ধান্ত কিম্বা শর্ত আরোপ ব্যতিরেকে অনির্দিষ্টভাবে লক্ষহীন কোন কার্য সম্পাদনের জন্য আনকন্ডিশনাল ব্রাঞ্চিং স্টেটমেন্ট প্রয়োজন হয়। আনকন্ডিশনাল ব্রাঞ্চিং এর জন্য সচরাচর goto-label স্টেটমেন্ট ব্যবহৃত হয়ে থাকে।

নিচে এসকল ব্রাঞ্চিং স্টেটমেন্টগুলো ব্লক চিত্রের মাধ্যমে দেখানো হলো।

if Statement :

if হলো পাইথন প্রোগ্রামের একটি অত্যন্ত গুরুত্বপূর্ণ **Decision control statement** প্রোগ্রামের শর্ত সাপেক্ষে কোনো স্টেটমেন্ট সম্পাদনের জন্য **if** স্টেটমেন্ট ব্যবহৃত হয়।

সি-প্রোগ্রামে **if Statement** চারভাবে ব্যবহার করা যায়। যথা:

- a) **Simple if statement** (স্পিগল **if Statement** স্টেটমেন্ট)
- b) **if else statement** (**if else** স্টেটমেন্ট)
- c) **Nested if else statement** (নেস্টেড **if else** স্টেটমেন্ট)
- d) **Else if ladder** (**if-else** ল্যাডার)

if-else স্টেটমেন্ট :

এটি simple if statement এরই একটি বর্ধিত রূপ। এই statement টি test expression-এর ফলাফল true হতে true-block এর স্টেটমেন্টগুলো নির্বাহ করবে আর False হলে false-block এর স্টেটমেন্টগুলোর নির্বাহ করবে। নিম্নে সিনটেক্স দেওয়া হলো:

If (test-expression)

True-block statement

Else

False-block-statement

Else স্টেটমেন্ট

আগের চ্যাপ্টাওে আমরা দেখেছি কিভাবে একটি If কন্ডিশন সত্য হলে তার আওতাভুক্ত কোড ব্লকটি রান হয়। Else বস্তুত If এর সাথেই সম্পর্কিত। অর্থাৎ যদি উল্লেখিত If কন্ডিশনটি সত্য না হয় তাহলে Else এর আওতাভুক্ত কোডব্লক রান বা এক্সিকিউট হয়।

X=4

If x ==5;

Print (“Its 5”)

Else:

Print (“its not 5”)

আউটপুট,

Its not 5

If Else স্টেটমেন্ট

একটি If স্কোপের মধ্যে যেহেতু যেকোনো কোডই থাকতে পাও সেহেতু এর মধ্যে আরও এক বা একাধিক If বা Else লজিক অবস্থান করতেই পারে। যেমন নিচের উদাহরণে প্রথমেই একটি If দিয়ে চেক করা হচ্ছে যে num এর ভ্যালু 5 কিনা।

Else এর আরও ব্যবহার

শুধু If এর সাথে ব্যবহার বাদেও Else কে ব্যবহার করা যায় for এবং while লুপের সাথেও। উদাহরণস্বরূপ, যখন কোন ফর লুপের কাজ স্বাভাবিক ভাবে শেষ হয় তখন এর সাথে যুক্ত Else ব্লকের কোড এক্সিকিউট হয়।

Simple if statement

এই **statement**-টির দ্বারা শর্ত সাপেক্ষে প্রোগ্রামের কোনো একটি লাইন বা একাধিক লাইন (**Block**) কে নির্বাহ করা বা নির্বাহ থেকে বিরত রাখার জন্য ব্যবহার করা হয়। **if (condition)** স্টেটমেন্টের পর কোনো সেমিকোলন থাকবে না। যদি **if Statement** এর শর্তেও অদীনে মাত্র একটি **Statement** থাকে তবে তার সিনটেক্স বা ফরম্যাট নিম্নরূপ ফ্লোচার্ট হিসেবে চিত্র-৬.১ কে ব্যবহার করা যেতে পারে:

If (test-expression or Condition)

Statement:

উদাহরণ-৩ :

a=40 b=20 if a>b:

Print (“a is greater than b”)

A is greater than b

৬.৩ nested if else স্টেটমেন্ট :

একটি if else স্টেটমেন্টের মধ্যে একাধিক if else স্টেটমেন্ট ব্যবহার করলে তাকে nested if else স্টেটমেন্ট বলে

সিনট্যাক্স

If Condition1:

Statements

else: if Condition2:

Statements

.....

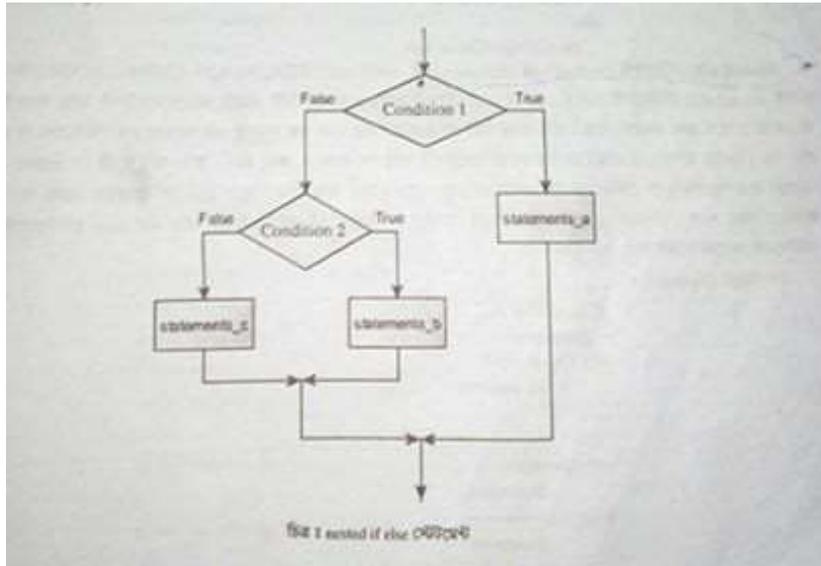
.....

.else: if Condition n:

Statements

else:

Statements

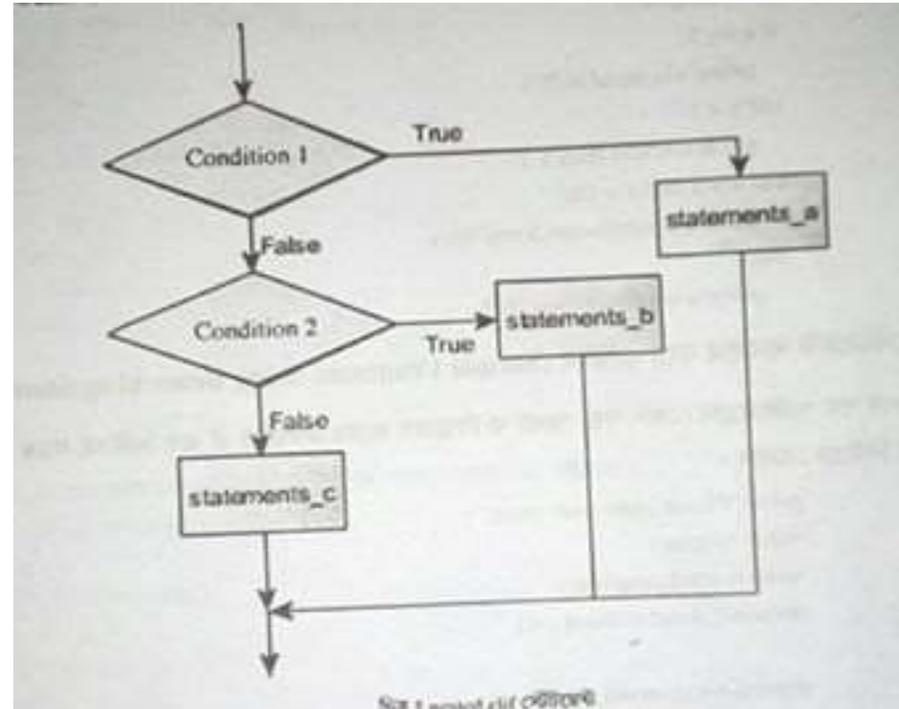


nested elif স্টেটমেন্ট :

একটি elif স্টেটমেন্টের মধ্যে একাধিক elif স্টেটমেন্ট ব্যবহার করলে তাকে nested elif স্টেটমেন্ট বলে

সিনট্যাক্স :

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



Nested if স্টেটমেন্ট:

Nested if Statement একটি if স্টেটমেন্টের মধ্যে একাধিক if স্টেটমেন্ট ব্যবহার করলে তাকে nested if বলে

সিনট্যাক্স:

if Conditiona1:

Statements

If Condition2:

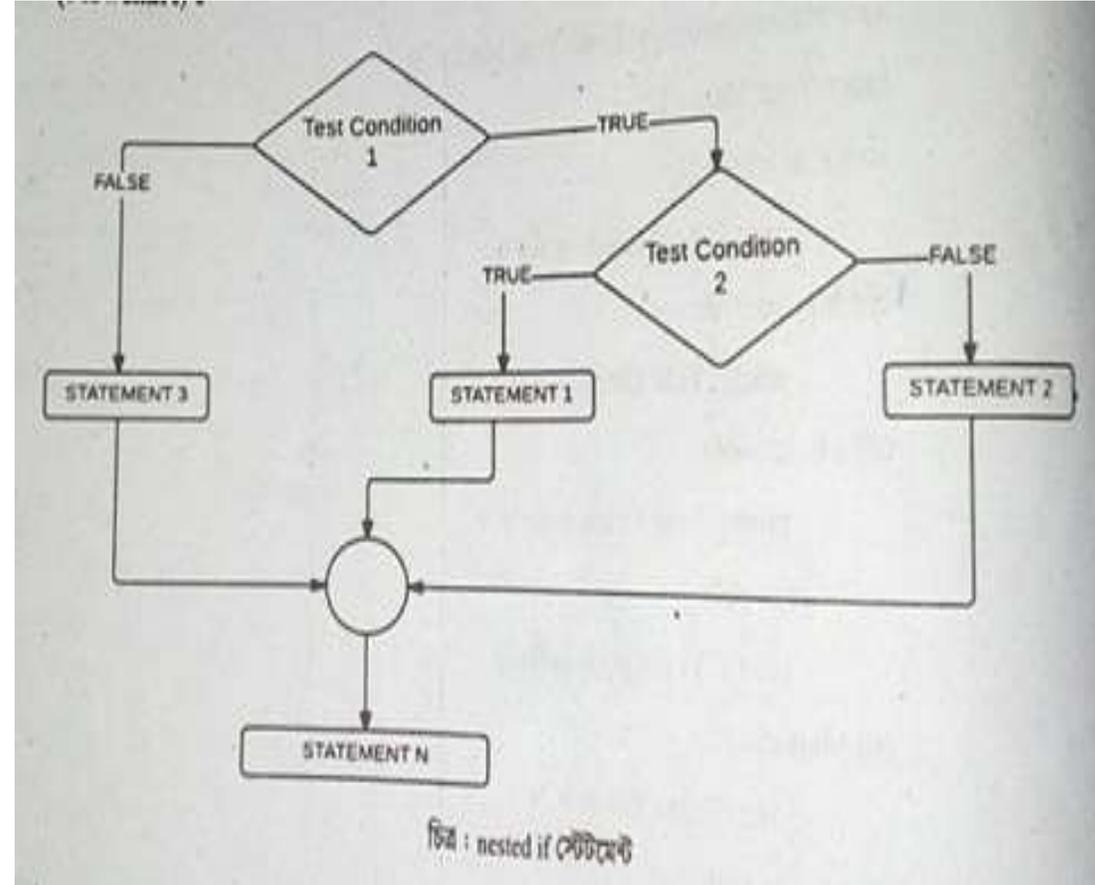
Statement

.....

.....

If Condition n:

Statement



৭.১ কন্ডিশনাল ও আনকন্ডিশনাল লুপিং ফ্লো

একটি প্রোগ্রামের এক বা একাধিক লাইন বা একটি নির্দিষ্ট অংশকে একাধিকবার শর্তে নির্বাহ করা হয় অর্থাৎ প্রোগ্রাম একটি নির্দিষ্ট অংশে কাজ শর্তে একাধিকবার নির্বাহ করা হয় তখন প্রোগ্রামের এ ধরনের কাজকে কন্ডিশনাল লুপিং বলে

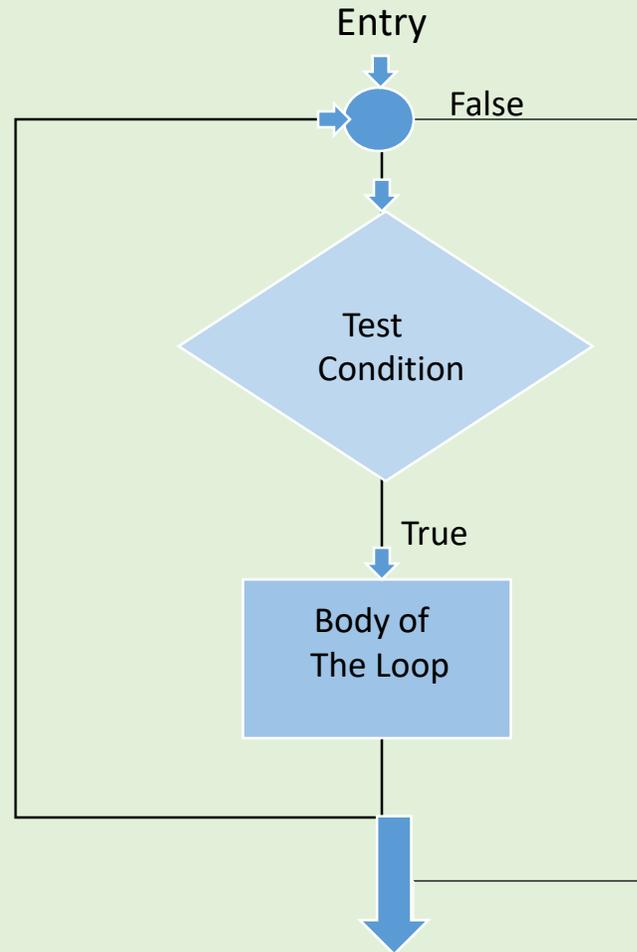
শর্তবিহীনভাবে প্রোগ্রামের কোনো একটি নির্দিষ্ট অংশে বারবার নির্বাহ হতে থাকে তখন এ ধরনের লুপিংকে আনকন্ডিশনাল লুপিং ফ্লো বলে /

কন্ডিশনাল লুপিং এর বৈশিষ্ট্য হচ্ছে লুপিং সংঘটিত হওয়ার জন্য একটি Test Condition বা শর্ত দেওয়া থাকবে / যে সময় পর্যন্ত Test Condition টি True অথবা False হবে ততসময় লুপিং নির্বাহ হতে থাকবে / প্রোগ্রামের যেটুকু অংশ বারবার নির্বাহ হয় সেটুকু অংশকে Body of the loop বা লুপের বডি বলে /

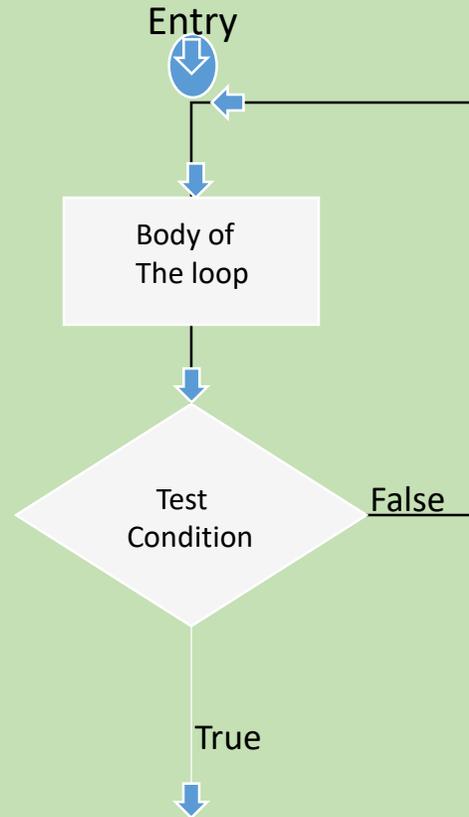
লুপের বডির নির্বাহ বন্ধ হওয়ার পদ্ধতি দুটি

1. Entry Control system .
২. Exit control system

Entry Control System হচ্ছে সেখানে Test Condition টি true হলে লুপের বডি পুনরায় নির্বাহ হবে আর false হলে প্রোগ্রামের পরবর্তী অংশে যাবে



Exit Control Systemসেখানে লুপের বডি একবার নির্বাহ হওয়ার পর test condition টি true হলে কন্ট্রোল প্রোগ্রামের পরবর্ত অংশে চলে যাবে আর false হলে লুপের বডি পুনরায় নির্বাহ হবে



৭.২ Syntax এর ব্যখ্যা

এখানে value হচ্ছে ভেরিয়েবল যা প্রতিবার ইটারেশন এর সময় সিকুয়েন্স থেকে ভেলু গ্রহন করে সিকুয়েন্সের শেষ উপাদানে না পৌঁছানো পর্যন্ত লুপ চলতেই থাকে / for এর body কে আলাদা করার জন্য ইন্ডেন্টেশন ব্যবহার করা হয় /

উদাহরন : পাইথন ..

```
# এই প্রোগ্রামটি লিস্ট এ জমা রাখা সকল আইটেম
# এর যোগফল নির্ণয় করবে।

# সংখ্যার লিস্ট
numbers = [1,9,7,8,2,5,8]
# যোগফল জমা রাখার ভ্যারিয়েবল
sum = 0
# লিস্টের সবগুলো আইটেম পুনরাবৃত্তি (iterate) করবে।
for val in numbers :
# যোগফল প্রিন্ট করবে।
print ("The sum is", sum)
# Output The sum is 40
Python
Copy
```

Range:

প্যারামিটার এর ব্যাখ্যা

* Start এবং stop যথাক্রমে ব্যবধির (range) প্রথম এবং শেষ উপাদান।

* Step size হলো উপাদানের ক্রমিক ধাপ। কোনো কিছু না বলে দিলে step size এর ডিফল্ট মান 1 থাকে।

range () ফাংশনের সকল ড্যাটা মেমোরিতে সংরক্ষিত হয় না। সুতরাং এটি শুধুমাত্র Start, stop এবং step size মনে রাখে এবং সেই অনুযায়ী নাম্বার উৎপাদন করে।

এই ফাংশন এর মাধ্যমে সকল উপাদান বা আইটেম এর আউটপুট নিতে আমরা list() ফাংশন ব্যবহার করতে পারি।

নিচের উদাহরণের মাধ্যমে এটি আপনার কাছে আরও পরিষ্কার হয়ে যাবে।

```
# রেঞ্জ প্রিন্ট হবে
print (range(10))
# Output range (0, 10)

# 0 থেকে 10 এর মধ্যে রেঞ্জ এর লিস্ট প্রিন্ট হবে
print(list(range(10)))
# Output : [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

# 3 থেকে 6 এর মধ্যে রেঞ্জ এর লিস্ট প্রিন্ট হবে
print(list(range(3,6)))
# Output : [3, 4, 5]

# 4 ব্যবধানে 3 থেকে 20 এর মধ্যে রেঞ্জ এর লিস্ট প্রিন্ট হবে
print(list(range(3, 20, 4)))
# Output [3, 7, 11, 15, 19]
Python
Cypy
```

for লুপ এর মধ্যে range() ফাংশন ব্যবহার করে আমরা নাম্বারের সিকুয়েন্সকে ইটারেট (iterate) করতে পারি। ইন্ডেক্সিং ব্যবহার করে সিকুয়েন্সকে ইটারেট করতে এই ফাংশন এর প্যারামিটার হিসাবে আমরা len() ফাংশন ব্যবহার করতে পারি।

উদাহরণস্বরূপ :

```
# index ব্যবহার করে সমগ্র লিস্টকে ইটারেট করবে।
# color এর লিস্ট
color=['red','green','blue']
# index ব্যবহার করে সম্পূর্ণ লিস্টের পুনরাবৃত্তি ঘটাবে
for i in range(len(color)):
    print("I like", color[i])
# Output:
# My favourite color is red
# My favourite color is green
# My favourite color is blue
Python
```

else for loop :

For লুপের সাথে অতিরিক্ত হিসেবে else ব্লকও থাকতে পারে / for লুপের কন্ডিসন false হলে else এর কোড ব্লক সম্পাদিত হবে/

For লুপ বন্ধ করার জন্য break স্টেটমেন্ট ব্যবহার করা হয়/

লুপের else অংশ কেবল তখনই সম্পন্ন হয় যখন কোনো break স্টেটমেন্ট থাকে না এবং কন্ডিসন false হয়/

উদাহরন:

```
# for লুপ এর সাথে else ব্লক এর ব্যবহার
# ডিজিট এর একটি লিস্ট
list_of_digits = [0,1,2,3,4,5,6,7,8,9]

# বিভিন্ন রেজাল্ট পাওয়ার ইনপুট ডিজিট পরিবর্তন করুন
input_digit = 4

# ইউজার থেকে ইনপুট নিতে নিচের কোড এর কমেন্ট তুলে দিন
# input_digit = int (input("Enter a digit: "))

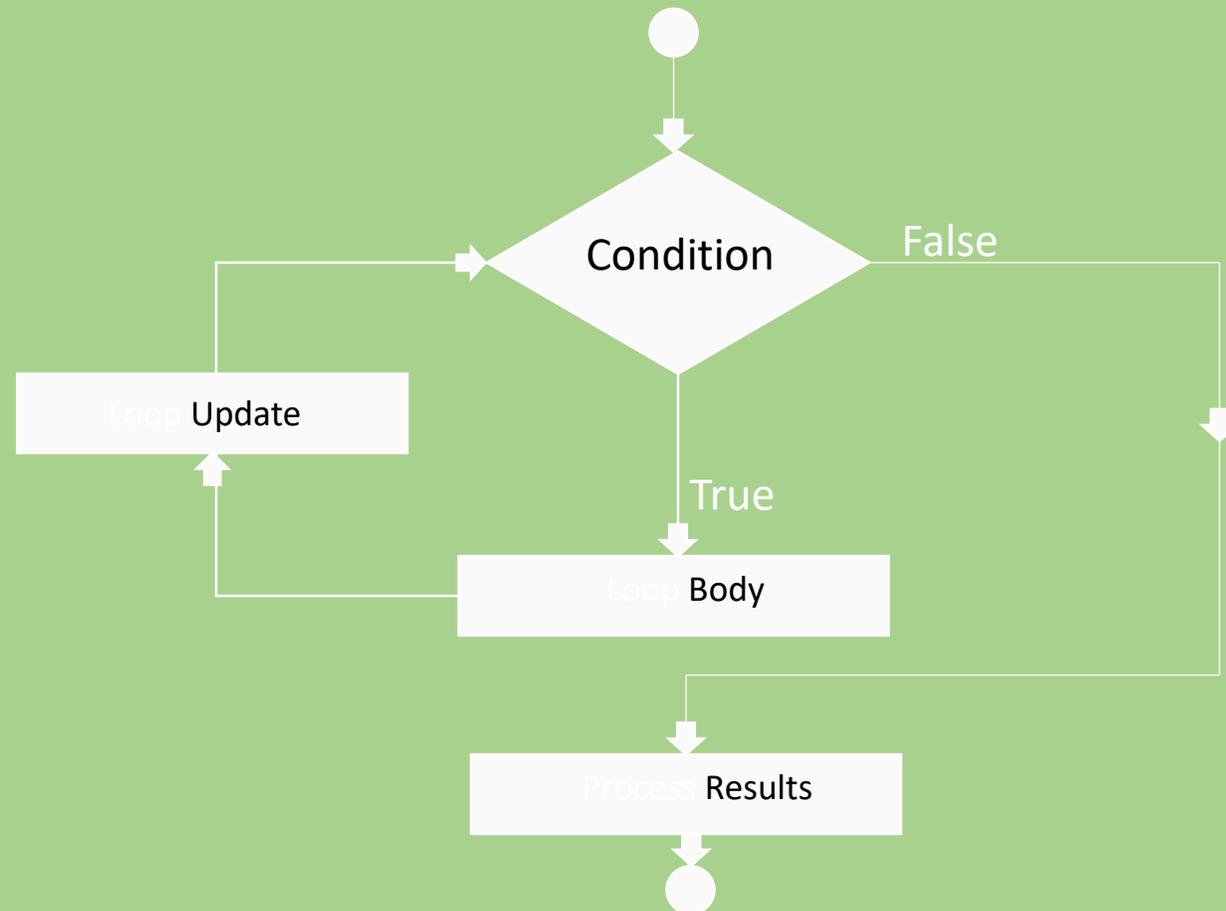
# আমাদের লিস্ট থেকে ইনপুট ডিজিট খুঁজে বের করবে।
for i in list_of_digits :
    if input_digit == i:
        print("Digit is in the list")
        break

    else :
        print ("Digit not found in list")

# Output : আপনার ইনপুট এর উপর ভিত্তি করে ফলাফল পাবেন।
Python
Copy
```

৭.৩ While Loop

কোনো নির্দিষ্ট কোড ব্লকের রিপিট করার জন্য প্রোগ্রামিং এর লুপ ব্যবহার করা হয়/ যত সময় পর্যন্ত Text Expression বা কন্ডিশন True থাকে তত সময় পর্যন্ত কোড ব্লককে ইটারেট করার জন্য পাইথনে While Loop ব্যবহৃত হয়/কোনো কোড ব্লককে কত সংখকবার ইটারেট করতে হবে পূর্বে তা জানা না থাকলে আমরা সাধারনত লুপ ব্যবহার করি/



উদাহরন:

স্বাভাবিক সংখ্যা যোগফল নির্ণয়ের প্রোগ্রাম

```
# sum = 1+2+3+... + n
```

```
# ইউজার থেকে ইনপুট গ্রহন এর জন্য
```

```
# কमेंট তুলে দিন
```

```
# n = int(input("Enter n: "))
```

```
n = 10
```

```
# count এবং sum কে ইনিশিয়লাইজ করা
```

```
count = 1
```

```
sum = 0
```

```
while count <= n:
```

```
sum = sum + count
```

```
count = count + 1 # update counter
```

```
# যোগফল প্রিন্ট হবে
```

```
print ("The sum is", sum)
```

```
Python
```

```
Copy
```

else while loop

For লুপের ন্যায় while লুপেরও অতিরিক্ত else ব্লক থাকতে পারে /
এখানে while loop কন্ডিশন false হলে else অংশ সম্পাদিত হবে/ while বন্ধ করার জন্য break স্টেটমেন্ট
ব্যবহার করা হয়/ এটি else অংশকে এরিখে যায়/
সুতরাং loop এর else অংশ কেবল তখন সম্পন্ন হয় যখন কোনো break স্টেটমেন্ট থাকে না এবং
কন্ডিশন false হয়/

```
# for লুপ এর সাথে else স্টেটমেন্ট এর ব্যবহার
```

```
counter = 0
```

```
while counter < 5:
```

```
    print("Inside while loop")
```

```
    counter = counter + 1
```

```
else :
```

```
    print ("Inside else")
```

```
Python
```

```
Copy
```

৭.৪ Infinite ও nested লুপ স্টেটমেন্ট

Infinite লুপ স্টেটমেন্ট: প্রোগ্রামের এক বা একাধিক স্টেটমেন্টকে শর্তহীনভাবে বা অনির্দিষ্ট সংখ্যক বার আবর্তন করানোর প্রক্রিয়াকে ইনফিনিট লুপিং বলে। পাইথনে দুই ধরনের ইনফিনিট লুপ তৈরি করা যায়। যথা-

➤ Infinite for loop: for স্টেটমেন্ট ব্যবহার করে যে লুপ তৈরি করা হয় তাকে ইনফিনিট for লুপ বলে। যেমন-

```
l=[1]
```

```
for x in l:
```

```
    l.append(x+1)
```

```
    print(x)
```

➤ Infinite while loop: while স্টেটমেন্ট ব্যবহার করে Infinite while লুপ তৈরি করা হয়।

```
উদাহরণ-১ :
while 1 == 1:
    print("Welcome! Don't afraid. Please Press ctrl+c")

উদাহরণ-২ :
flag = 1
while (flag):
    print ('Given flag is really true!')
    print ("Good bye!")
```

Nested লুপ স্টেটমেন্ট: একটি লুপিং স্টেটমেন্টের মধ্যে অন্য কোনো এক বা একাধিক লুপ স্টেটমেন্ট ব্যবহার করলে তাকে নেস্টেড লুপিং বলে। পাইথনে দুই ধরনের নেস্টেড লুপ থাকে। যথা-

1. Nested for লুপ স্টেটমেন্ট: যখন একটি for লুপ স্টেটমেন্টের মধ্যে অন্য কোনো এক বা একাধিক for লুপ স্টেটমেন্ট ব্যবহার করা হয় তখন তাকে নেস্টেড for লুপ বলে।

সিনটাক্স:

```
for iterating_variable in sequence:
    for iterating_variable in sequence:
        statements(s)
    statements(s)
```

উদাহরণ-১ :

```
num_list = [1, 2, 3]
alpha_list = ['a', 'b', 'c']

for number in num_list:
    print(number)
    for letter in alpha_list:
        print(letter)
```

আউটপুট :

```
1
a
b
c
2
a
b
c
3
a
b
c
```

2.Nested while লুপ স্টেটমেন্ট: যখন একটি while লুপের মধ্যে এক বা একাধিক while লুপ ঘোষণা করা হয়, তখন তাকে Nested while লুপ বলে।

সিনটাক্স:

```
while Condition:  
    while Condition:  
        statement(s)  
    statement(s)
```

```
b = 0  
while b <= 2:  
    x = 0  
    print (b)  
    while x <= 2:  
        print (x)  
        x += 1  
    b += 1
```

আউটপুট :

```
0  
0  
1  
2  
1  
0  
1  
2  
2  
0  
1  
2
```

৭.৫ Break, continue ও pass স্টেটমেন্ট

পাইথনে তিন ধরনের লুপ কন্ট্রোল স্টেটমেন্ট রয়েছে। যথা-

- **Break statement:** Break স্টেটমেন্টের কাজ হলো Break স্টেটমেন্ট পাওয়া মাত্রই লুপের এক্সিকিউশন তাৎক্ষণিকভাবে থেমে লুপ থেকে বের হয়ে যাবে এবং লুপের বাইরে কোনো স্টেটমেন্ট থাকলে সেগুলোর এক্সিকিউশন শুরু করা।

উদাহরণ-১ :

```
for letter in "string":  
    if letter == "n" :  
        break  
    print(letter )  
print ("The end" )
```

আউটপুট :

```
M  
a  
h  
i  
The end
```

➤ **Continue statement:** Continue স্টেটমেন্টের কাজ হলো Continue স্টেটমেন্ট পাওয়া মাত্রই লুপের এক্সিকিউশন তাৎক্ষণিকভাবে থেমে যাবে কিন্তু লুপ থেকে বের হয়ে যাবে না। শুধুমাত্র কন্ডিশনের অংশটুকু বাদ যাবে এবং র বাকি অংশে কাজ সম্পন্ন করবে।

উদাহরণ-২ :

```
for number in range(1, 10):  
    if number == 6:  
        continue  
    print(number)
```

আউটপুট :

1
2
3
4
5
7
8
9

➤ **Pass statement:** Empty লুপ তৈরির জন্য Pass স্টেটমেন্ট ব্যবহৃত হয়। Pass স্টেটমেন্ট ব্যবহার করলে কোনো স্ট্রিং -এর সর্বশেষ লিস্ট এর সর্বশেষ সংখ্যা আউটপুট হিসেবে দেখাবে।

```
8
for letter in 'Mahi is a good boy':
    pass
print ('Last Letter :', letter)
```

আউটপুট :

```
Last Letter : y
```

৭.৬লুপিং স্টেটমেন্ট ব্যবহার করে প্রোগ্রাম

প্রোগ্রাম -১: কোনো একটি পূর্ণসংখ্যা ইনপুট নিয়ে ঐ পূর্ণসংখ্যার নামতা প্রিন্ট করার প্রোগ্রাম

```
print('Please, input the number:')  
number = int(input())  
count = 1  
while count <= 10:  
    print(number, 'x', count, '=', number*count)  
    count += 1
```

৮ম অধ্যায়

লিস্ট

৮.১ লিস্ট ও লিস্টের উপাদান

লিস্ট: লিস্ট হচ্ছে পাইথনের সবচেয়ে বৈচিত্রপূর্ণ ডাটা টাইপ, যা স্ফার ব্র্যাকেট [] এর ভেতর কন্মার সাহায্যে উপাদানসহ প্রকাশ করা হয়। তবে লিস্ট এর উপাদানগুলো অ্যারের মতো একই রকম ডাটা টাইপ হবার প্রয়োজন নেই।

৮.২ লিস্টের উপাদান অ্যাক্সেসিং

লিস্টের উপাদান অ্যাক্সেসিং করার জন্য পাইথনে বিভিন্ন পদ্ধতি প্রচলিত আছে। যথা:

- ইনডেক্সিং পদ্ধতি
- নেগেটিভ ইনডেক্সিং পদ্ধতি
- স্লাইসিং পদ্ধতি

ইনডেক্সিং পদ্ধতি : যে পদ্ধতিতে লিস্টের কোন আইটেমকে অ্যাক্সেস করার জন্য ইনডেক্স নম্বর ব্যবহার করা হয় তাকে ইনডেক্সিং পদ্ধতি বলে। লিস্টের ইনডেক্স নম্বর শুরু হয় ০ থেকে।

নেগেটিভ ইনডেক্সিং পদ্ধতি : পাইথনে ইনডেক্স -১ দ্বারা সর্বশেষ আইটেম এবং -২ দ্বারা দ্বিতীয় সর্বশেষ আইটেমকে বুঝায়। এভাবেই নেগেটিভ ইনডেক্সিং -এর ধারা চলেতে থাকে।

স্লাইসিং পদ্ধতি : স্লাইসিং অপারেটর (:) কোলন ব্যবহার করে লিস্টের আইটেমকে অ্যাক্সেস পদ্ধতিকে স্লাইসিং বলে।

৮.৩ লিস্টের উপাদান আপডেটিং ও ডিলেটিং

লিস্টের উপাদান আপডেটিং: লিস্টের উপাদান আপডেটিং বলতে লিস্টের ইনডেক্সে যে আইটেমগুলো আছে সেগুলোর মধ্যে কোন পরিবর্তন, পরিবর্ধন, সংযোজন, বিয়োজন ইত্যাদি অপারেশনকে বুঝায়।

লিস্টের নতুন কোন উপাদান সংযোজন : লিস্টে নতুন কোন আইটেম সংযোজন করতে চাইলে পাইথনে তিন ধরনের ফাংশন ব্যবহার করা হয়। যথা append() ফাংশন, insert() ফাংশন, extend() ফাংশন।

লিস্টের নতুন কোন উপাদান বিয়োজন : লিস্ট থেকে এর কোন আইটেম রিমুভ করার জন্য পাইথনে দুই ধরনের ফাংশন ব্যবহার করা হয়। যথা del() ফাংশন ও remove() ফাংশন।

৮.৪ লিস্টের বিভিন্ন অপারেশন

পাইথনে মোট তিন ধরনের অপারেশন আছে। যথাঃ

- কনক্যাটেশন অপারেটর
- রিপিটিশন অপারেটর
- মেম্বারশীপ অপারেটর

কনক্যাটেশন অপারেটরঃ

দুটি লিস্টের উপাদান সমূহ কে সংযুক্ত করে নতুন কোন লিস্ট তৈরী করার জন্য কনক্যাটেশন অপারেটর (+) ব্যবহার করা হয়। যেমনঃ

```
List1= ["Mahbub", 960021, "Mahi", 23022015]
```

```
List2= [19072017, "Mahdee", "Tahsin"]
```

```
List3= List1+List2
```

```
Print(List3)
```

Output: ["Mahbub", 960021, "Mahi", 23022015, 960021, "Mahi", 23022015]

রিপিটিশন অপারেটর:

কোন লিস্টে বিদ্যমান উপাদান সমূহ কে একাধিকবার রিপিট করার জন্য রিপিটিশন অপারেটর (*) ব্যবহার করা হয়। যেমন:

```
List1=["Mahbub", 960021, "Mahi", 23022015]
print(List1*2)
```

Output: ["Mahbub", 960021, "Mahi", 23022015,"Mahbub", 960021, "Mahi", 23022015]

মেম্বারশীপ অপারেটর:

কোন লিস্টের মধ্যে নিদৃষ্ট কোন এলিমেন্টের উপস্থিতি বা অনুপস্থিতি চেক করার জন্য মেম্বারশীপ অপারেটর (ইন/ নট ইন) ব্যবহার করা হয়। যেমন:

```
List1=["Mahbub", 960021, "Mahi", 23022015]
List2= [19072017, "Mahdee", "Tahsin"]
print("Samsul" in List1)
print("Mahdee" in List1)
print("Mahdee" in List2)
print("Mahbub" in List2)
```

Output:

```
False
False
True
False
```

৮.৫ লিস্ট ব্যবহৃত বিল্ট ইন ফাংশন ও মেথডসমূহ

লিস্ট ব্যবহৃত বিল্ট ইন ফাংশনঃ

পাইথন প্রোগ্রামে লিস্ট নিয়ে দ্রুততার সঙ্গে কাজ সম্পাদনের জন্য এতে বেশ কিছু বিল্ট ইন ফাংশন রয়েছে। যেমনঃ

বিল্ট ইন ফাংশন

বর্ণনা

- ❖ Min - লিস্টের সকল উপাদানের মধ্যে ছোট আইটেম জানার জন্য
- ❖ Any - একটি মান সত্য হলে True রিটার্ন করে। লিস্টটি খালি হলে False রিটার্ন করে
- ❖ Enumerate - লিস্টের সকল উপাদানের ইনডেক্স নম্বর ও তাদের মান ধারণ করে
- ❖ All - লিস্টের সকল উপাদানের মান সত্য হলে True রিটার্ন করে।
- ❖ Cmp(list1, list2) - একাধিক লিস্টের উপাদান সমূহের মধ্যে তুলনা করে
- ❖ Len - লিস্টের সকল উপাদানের সংখ্যা জানার জন্য ব্যবহার হয়
- ❖ List - ইটারেবল ডাটা সমূহ কে লিস্ট রূপান্তরের জন্য ব্যবহার হয়
- ❖ Max - লিস্টের সকল উপাদানের মধ্যে বড়টি জানার জন্য ব্যবহার হয়
- ❖ Sorted - সর্টেড লিস্ট রিটার্ন এর জন্য ব্যবহার হয়
- ❖ Sum - লিস্টের সকল উপাদানের যোগফল জানার জন্য ব্যবহার হয়

মেথড

বর্ণনা

- ❖ Extend - এক লিস্টের উপাদান অন্য লিস্টে সংযোজন করতে ব্যবহার হয়
- ❖ Copy - কপি করার জন্য ব্যবহার হয়
- ❖ Append - লিস্টের শেষে একটি এলিমেন্ট যুক্ত করার জন্য ব্যবহার হয়
- ❖ Removed - একটি আইটেম কে রিমুভ করার জন্য ব্যবহার হয়
- ❖ Insert - কোন আইটেম সংযোজনের জন্য ব্যবহার হয়
- ❖ Index - এলিমেন্টের অবস্থান চেক করার জন্য ব্যবহার হয়
- ❖ Pop - একদম শেষের আইটেম কে রিমুভ করার জন্য ব্যবহার হয়
- ❖ Clear - এক লিস্টের সকল উপাদান কে রিমুভ করার জন্য ব্যবহার হয়
- ❖ Count - এক লিস্টের সকল উপাদান সংখ্যা গণনা করার জন্য ব্যবহার হয়
- ❖ Sort - এক লিস্টের সকল উপাদান এ্যাসেনডিং অর্ডারে সাজানোর জন্য ব্যবহার হয়
- ❖ Reverse - এক লিস্টের সকল উপাদান রিভার্স অর্ডারে সাজানোর জন্য ব্যবহার হয়

৮.৬ লিস্ট ব্যবহার করে কতিপয় প্রোগ্রাম

একটি লিস্টে কতগুলো সংখ্যা ইনপুট নিয়ে তাদের যোগফল নির্ণয়ের প্রোগ্রাম

```
Numbers = [1,4,9,16]
sum = 0
for num in Numbers:
    sum += num
    print("summation is=", sum)
z = [10, 20, 30]
print(all([x>10 for x in z]))
print(all(["", "0", [0], "None" ]))
print(any ([ x>10 for x in z]))
print(any([ x>50 for x in z]))
print(any([], ( ),{ },None 0 ))
print(any([ 0 ], ( ),{ },None 0 ))
print(all(([ ]))
print(all(([ ]))
```

৯ম অধ্যায়

টাপল

৯.১ টাপল ও টাপলে অ্যাসাইনমেন্ট

টাপল : টাপল হচ্ছে পাইথনের সবচেয়ে বৈচিত্রপূর্ণ ডাটা টাইপ, যা প্যারেনথেসিস ব্র্যাকেট () এর ভেতর কমার সাহায্যে উপাদানসহ প্রকাশ করা হয়। তবে

টাপল এর উপাদানগুলো অ্যারের মতো একই রকম ডাটা টাইপ হবার প্রয়োজন নেই।

টাপলে ভ্যালু অ্যাসাইনমেন্ট

১। টাপলের মধ্যকার সকল ভ্যালুকে এক লাইনে আলাদা আলাদা নতুন ভ্যারিয়েবল হিসেবে অ্যাসাইন করা যায়।

২। টাপলের মধ্যে কোন ভেরিয়েবলের জন্য একাধিক ভ্যালু অ্যাসাইন করতে হলে ঐ ভেরিয়েবলের পূর্বে * চিহ্ন যুক্ত করে যে কোন সংখ্যক ভ্যালুকে এর মধ্যে অ্যাসাইন করা যায়।

টাপল উপাদান অ্যাক্সেসিং

টাপলের উপাদান অ্যাক্সেসিং করার জন্য পাইথনে বিভিন্ন পদ্ধতি প্রচলিত আছে। যথা:

- ইনডেক্সিং পদ্ধতি
- নেগেটিভ ইনডেক্সিং পদ্ধতি
- স্লাইসিং পদ্ধতি

ইনডেক্সিং পদ্ধতি : যে পদ্ধতিতে টাপলের কোন আইটেমকে অ্যাক্সেস করার জন্য ইনডেক্স নম্বর ব্যবহার করা হয় তাকে ইনডেক্সিং পদ্ধতি বলে। টাপল ইনডেক্স নম্বর শুরু হয় ০ থেকে।

নেগেটিভ ইনডেক্সিং পদ্ধতি : পাইথনে ইনডেক্স -১ দ্বারা সর্বশেষ আইটেম এবং -২ দ্বারা দ্বিতীয় সর্বশেষ আইটেমকে বুঝায়। এভাবেই নেগেটিভ ইনডেক্সিং -এর ধারা চলেতে থাকে।

স্লাইসিং পদ্ধতি : স্লাইসিং অপারেটর (:) কোলন ব্যবহার করে টাপলের আইটেমকে অ্যাক্সেস পদ্ধতিকে স্লাইসিং বলে।

৯.২ টাপলের উপাদান আপডেটিং ও ডিলেটিং

টাপলের উপাদান আপডেটিং: টাপলের উপাদান আপডেটিং বলতে লিস্টের ইনডেক্সে যে আইটেমগুলো আছে সেগুলোর মধ্যে কোন পরিবর্তন, পরিবর্ধন ,সংযোজন,বিয়োজন ইত্যাদি অপারেশনকে বুঝায়। পাইথনে লিস্টের মতো টাপলে ইচ্ছেমতো পরিবর্তন, পরিবর্ধন করা যায় না

টাপলের নতুন কোন উপাদান বিয়োজন : টাপলের থেকে এর কোন আইটেম রিমুভ করা যায় না। তবে সম্পূর্ণ টাপলকে ডিলিট করা যায়। এজন্য del() ফাংশন ব্যবহার করা হয়।

৯.৩ টাপলের বিভিন্ন অপারেশন

পাইথনে মোট তিন ধরনের অপারেশন আছে। যথাঃ

- কনক্যাটেশন অপারেটর
- রিপিটিশন অপারেটর
- মেম্বারশীপ অপারেটর

কনক্যাটেশন অপারেটরঃ

দুটি টাপল উপাদান সমূহ কে সংযুক্ত করে নতুন কোন টাপল তৈরীকরার জন্য কনক্যাটেশন অপারেটর (+) ব্যবহার করা হয়। যেমনঃ

```
List1= ["Mahbub", 960021, "Mahi", 23022015]
```

```
List2= [19072017, "Mahdee", "Tahsin"]
```

```
List3= List1+List2
```

```
Print(List3)
```

Output: ["Mahbub", 960021, "Mahi", 23022015, 960021, "Mahi", 23022015]

রিপিটিশন অপারেটর:

কোন টাপলে বিদ্যমান উপাদান সমূহ কে একাধিকবার রিপিট করার জন্য রিপিটিশন অপারেটর (*) ব্যবহার করা হয় । যেমনঃ

```
List1=["Mahbub", 960021, "Mahi", 23022015]  
print(List1*2)
```

Output: ["Mahbub", 960021, "Mahi", 23022015,"Mahbub", 960021, "Mahi", 23022015]

মেম্বারশীপ অপারেটর:

কোন টাপলের মধ্যে নিদৃষ্ট কোন এলিমেন্টের উপস্থিতি বা অনুপস্থিতি চেক করার জন্য মেম্বারশীপ অপারেটর (ইন/ নট ইন) ব্যবহার করা হয় । যেমনঃ

```
List1=["Mahbub", 960021, "Mahi", 23022015]  
List2= [19072017, "Mahdee", "Tahsin"]  
print("Samsul" in List1)  
print("Mahdee" in List1)  
print("Mahdee" in List2)  
print("Mahbub" in List2)
```

Output:

```
False  
False  
True  
False
```

৯.৫ টাপলে ব্যবহৃত বিল্ট ইন ফাংশন ও মেথডসমূহ

টাপলে ব্যবহৃত বিল্ট ইন ফাংশনঃ

পাইথন প্রোগ্রামে টাপলে নিয়ে দ্রুততার সঙ্গে কাজ সম্পাদনের জন্য এতে বেশ কিছু বিল্ট ইন ফাংশন রয়েছে। যেমনঃ

বিল্ট ইন ফাংশন

বর্ণনা

- ❖ Min - টাপলে সকল উপাদানের মধ্যে ছোট আইটেম জানার জন্য
- ❖ Any - একটি মান সত্য হলে True রিটার্ন করে। টাপলে খালি হলে False রিটার্ন করে
- ❖ Enumerate - টাপলে সকল উপাদানের ইনডেক্স নম্বর ও তাদের মান ধারণ করে
- ❖ All - টাপলে সকল উপাদানের মান সত্য হলে True রিটার্ন করে।
- ❖ Cmp(list1, list2) - একাধিক টাপলে উপাদান সমূহের মধ্যে তুলনা করে
- ❖ Len - টাপলে সকল উপাদানের সংখ্যা জানার জন্য ব্যবহার হয়
- ❖ List - ইটারেবল ডাটা সমূহ কে টাপলে রূপান্তরের জন্য ব্যবহার হয়
- ❖ Max - টাপলে সকল উপাদানের মধ্যে বড়টি জানার জন্য ব্যবহার হয়
- ❖ Sorted - সর্টেড টাপলে রিটার্ন এর জন্য ব্যবহার হয়
- ❖ Sum - টাপলে সকল উপাদানের যোগফল জানার জন্য ব্যবহার হয়

মেথড

বর্ণনা

- ❖ Extend - এক লিস্টের উপাদান অন্য লিস্ট সংযোজন করতে ব্যবহার হয়
- ❖ Copy - কপি করার জন্য ব্যবহার হয়
- ❖ Append - লিস্টের শেষে একটি এলিমেন্ট যুক্ত করার জন্য ব্যবহার হয়
- ❖ Removed - একটি আইটেম কে রিমুভ করার জন্য ব্যবহার হয়
- ❖ Insert - কোন আইটেম সংযোজনের জন্য ব্যবহার হয়
- ❖ Index - এলিমেন্টের অবস্থান চেক করার জন্য ব্যবহার হয়
- ❖ Pop - একদম শেষের আইটেম কে রিমুভ করার জন্য ব্যবহার হয়
- ❖ Clear - এক লিস্টের সকল উপাদান কে রিমুভ করার জন্য ব্যবহার হয়
- ❖ Count - এক লিস্টের সকল উপাদান সংখ্যা গণনা করার জন্য ব্যবহার হয়
- ❖ Sort - এক লিস্টের সকল উপাদান এ্যাসেনডিং অর্ডারে সাজানোর জন্য ব্যবহার হয়
- ❖ Reverse - এক লিস্টের সকল উপাদান রিভার্স অর্ডারে সাজানোর জন্য ব্যবহার হয়

৮.৬ লিস্ট ব্যবহার করে কতিপয় প্রোগ্রাম

একটি লিস্টে কতগুলো সংখ্যা ইনপুট নিয়ে তাদের যোগফল নির্ণয়ের প্রোগ্রাম

টাপলে ব্যবহৃত বিল্ট ইন মেথডসমূহঃ

মেথড

বর্ণনা

- ❖ Index () - এলিমেন্টের অবস্থান চেক করার জন্য ব্যবহার হয়
- ❖ Count () - এক টাপলের সকল উপাদান সংখ্যা গণনা করার জন্য ব্যবহার হয়

পাইথনের সেট

পাইথনের সেট: পাইথনের সেট হলো লিস্টের মতোই এক ধরনের ডাটা টাইপ।

সেটের উপাদান আপডেটিং: পাইথন সেটে কোন একটি এলিমেন্ট সংযোজনের জন্য add() মেথড এবং এবং একাধিক এলিমেন্ট সংযোজনের জন্য update() মেথড ব্যবহার করা হয়।

সেটের উপাদান বিয়োজন: পাইথনে সেটের সুনির্দিষ্ট কোন উপাদানকে রিমুভ করার জন্য চারটি মেথড ব্যবহার করা হয়। যথা- discard () মেথড, pop() মেথড, remove() মেথড, clear() মেথড।

পাইথন সেটের বিভিন্ন অপারেশন:

পাইথনে মোট পাঁচ ধরনের সেট অপারেশন রয়েছে, যথা-

- ইউনিয়ন অপারেশন
- ইন্টারসেকশন অপারেশন
- ডিফারেন্স অপারেশন
- মেম্বারশিপ অপারেশন ও
- সিমেন্ট্রিক ডিফারেন্স অপারেশন

পাইথনের ডিকশনারি

ডিকশনারি এমন এক ধরনের ডাটা স্ট্রাকচার যাতে লিস্টের মতই বিভিন্ন ধরনের এলিমেন্ট বা অবজেক্ট স্টোর করা যায়। তবে ডিকশনারিতে এলিমেন্ট গুলোকে ম্যানুয়ালি ইনডেক্সিং করতে হয়। প্রোগ্রামারকেই প্রতিটি এলিমেন্টের জন্য আলাদা আলাদা ইনডেক্স নির্ধারণ করে দিতে হয়।

৯.৬ টাপল ব্যবহার করে একটি প্রোগ্রাম

```
my-tuple = (1,2,3)
```

```
Print (my-tuple )
```

```
output: = (1,2,3)
```

```
my-tuple = (1,"Hello",3.4)
```

```
Print (my-tuple )
```

```
Output = (1,"Hello",3.4)
```

১০ম অধ্যায়

ফাংশন

১০.১ ফাংশন

ফাংশন : ফাংশন হচ্ছে একটি বড় প্রোগ্রামের একটি স্বাধীন ক্ষুদ্র অংশ, যার একটি নির্দিষ্ট নাম থাকে, যা এক বা একাধিক স্টেটমেন্টের সমন্বয়ে গঠিত এবং নির্দিষ্ট কোন সমস্যা সমাধানের জন্য ব্যবহৃত হয়।

ফাংশন ব্যবহারের সুবিধা:

- ফাংশন ব্যবহার করার ফলে প্রোগ্রামের কোনো অংশ মূল প্রোগ্রামে বারবার লেখার প্রয়োজন হয় না।
- প্রোগ্রামের সাইজ ছোট হয়।
- প্রোগ্রাম লিখতে সময় অপেক্ষাকৃত কম লাগে।
- প্রোগ্রামের ভুল নির্ণয় ও সংশোধন করা সহজ হয়।
- অত্যন্ত জরুরি প্রোগ্রামকে ফাংশন হিসাবে লাইব্রেরিতে জমা করে রাখলে অন্য ব্যবহারকারী প্রয়োজনে তা ব্যবহার করতে পারে।

ফাংশনের প্রকারভেদ:

- বিল্ট ইন ফাংশন বা লাইব্রেরি ফাংশন
- ইউজার ডিফাইন ফাংশন

বিল্ট ইন ফাংশন বা লাইব্রেরি ফাংশন : যে সকল ফাংশন ডেভলপার কর্তৃক পূর্ব থেকেই নির্ধারিত কাজের জন্য ডিজাইনকৃত, ব্যবহারকারী চাইলেও যাদেও নাম পরিবর্তন করতে পারে না সে সকল ফাংশনকে লাইব্রেরি ফাংশন বলে।

ইউজার ডিফাইন ফাংশন : ইউজার বা ব্যবহারকারী তার নিজস্ব প্রয়োজন এবং প্রজ্ঞা অনুযায়ী যে-সব ফাংশন ডিফাইন করে প্রোগ্রামে ব্যবহার করেন সে সকল ফাংশনকে ইউজার ডিফাইন ফাংশন বলে।

১০.২ ফাংশন তৈরি ও কল করা

ফাংশন তৈরি: পাইথনে ফাংশন তৈরির সাধারণ গঠন নিম্নরূপ।

```
def FunctionName (Parameters):
```

```
    Statements
```

```
    return [expression]
```

ফাংশন কলিং: call শব্দের অর্থ ডাকা বা অহ্বান করা। আমরা যখন কোন ফাংশন ডিফাইন করি তখন তার একটি নাম দেই এবং প্রত্যেকটি ফাংশন কোন নির্দিষ্ট কাজ করার জন্য ডিফাইন করা হয়।

১০.৩ পাস বাই রেফারেন্স ও পাস বাই ভ্যালু

পাস বাই রেফারেন্স :রেফারেন্স ভেরিয়েবলের আরগুমেন্টের মাধ্যমে এক ফাংশন থেকে অন্য ফাংশনে ডাটা পাঠানোর পদ্ধতিকে পাস বাই রেফারেন্স বলে।

পাস বাই ভ্যালু : ফাংশন কল করার সময় আরগুমেন্ট ভেরিয়েবলের মাধ্যমে এক ফাংশন থেকে অন্য ফাংশনে ডাটা পাঠানোর পদ্ধতিকে পাস বাই ভ্যালু বলে।

১০.৪ ফাংশনে আরগুমেন্টের ব্যবহার

প্রোগ্রামিং একটি ফাংশনের কিছু আর্গুমেন্ট থাকতে পারে যেগুলো পক্ষান্তরে ফাংশন বডি'র মধ্যে ব্যবহৃত হয়ে চাহিদা মোতাবেক প্রসেস হবে। এই আর্গুমেন্টগুলো পাঠানোর দায়িত্ব হচ্ছে তার,যে এই ফাংশনকে কল করবে বা ব্যবহার করতে চাইবে। উদাহরণ :

```
def show_double(x):
```

```
Print(x*2)
```

```
Show_double(2)
```

```
Show_double(100)
```

```
আউটপুট,
```

```
4
```

```
200
```

```
ইনপুট নিয়ে..
```

```
Def make_sum(x,y):
```

```
Z=x+y
```

```
print(z)
```

```
make_sum(5,10)
```

```
Make_sum(500,500)
```

```
আউটপুট,
```

```
15
```

```
1000
```

* যখন একটি ফাংশনকে ডিফাইন করা হয় তখন এর ভেরিয়েবলকে প্যারামিটার বলা হয় আর যখন একটি ফাংশনকে কল করা হয় তখন সেই ফাংশনের প্যারামিটার হিসাবে যে ভ্যালু পাঠানো হয় তাকে আর্গুমেন্ট বলা হয়।

১০.৫ তারিখ ও সময় ফাংশনের ব্যবহার

*তারিখ ও সময়কে () নিয়ে পাইথন প্রোগ্রামে বিভিন্ন ধরনের অপারেশন সম্পাদনের জন্য মডিউল রয়েছে। যথা:

(i)Date

(ii)Time

(iii)Datetime

(iv)Timedelta এবং

(v)Tzinfo

আগামি দিনের তারিখ নির্ণয়:

উদাহরণ:

```
from datetime import date
from datetime import timedelta
def tomorrow():
    return date.today()+timedelta(days=1)
print(date.today())1
print(tomorrow())
```

আউটপুট:

2018-01-20

2018-01-21

১০.৬ ফাংশন ব্যবহার করে প্রোগ্রাম

*ফাংশন ব্যবহার করে আয়তক্ষেত্রের ক্ষেত্রফল নির্ণয়ের প্রোগ্রাম:

```
def Area():
    length =intinput("Enter the value of Length:")
    Width =int (input("Enter the value of Width:"))
    RectangleArea=Lengh*Width
    Print("Area of Rectangle is =",RectangleArea)
```

১০.৭ ভেরিয়েবলের স্কোপ

ভেরিয়েবলের স্কোপ এর উপর ভিত্তি করে ভেরিয়েবল প্রধানত দুইপ্রকার । যথা-

- লোকাল ভেরিয়েবল
- গ্লোবাল ভেরিয়েবল

লোকাল ভেরিয়েবল: যে- সব ভেরিয়েবলের কার্যকারিতা শুধুমাত্র কোন নির্দিষ্ট অংশ বা ফাংশনের মধ্যে সীমাবদ্ধ থাকে, তাকে লোকাল ভেরিয়েবল বলে ।

গ্লোবাল ভেরিয়েবল: যখন কোন ভেরিয়েবলের কার্যকারিতা শুধুমাত্র কোন নির্দিষ্ট অংশ বা ফাংশনের মধ্যে সীমাবদ্ধ না থেকে এর মান সকল ফাংশনেই ব্যবহার করা যায়,তাকে গ্লোবাল ভেরিয়েবল বলে ।

১০.৮ রিকার্সিভ ফাংশন

রিকার্সিভ ফাংশন : কোন ফাংশন নিজেই নিজেকে যখন কল কওে তখন সেই ফাংশনকে রিকার্সিভ ফাংশন বলে ।

১০.৯ অ্যানোনিমাস ফাংশন ঃ ল্যাম্বডা

ল্যাম্বডা : পাইথনে ল্যাম্বডা অপারেটর ব্যবহার করে এক লাইনের যে ফাংশন তৈরি করা হয়,তাকে ল্যাম্বডা বলে ।একে অ্যানোনিমাস ফাংশনও বলে ।

১১ম অধ্যায়

ফাইলের ইনপুট আউটপুট অপারেশন

১১.১ স্ক্রিনে আউটপুট প্রদর্শন

প্রদর্শন স্ক্রিনের মধ্যে ডাটা আউটপুটের জন্য পাইথন Print() ফাংশন ব্যবহার করা হয়। Print() ফাংশন কেবলমাত্র তাই আউটপুট দেয় যা এর আর্গুমেন্ট হিসেবে দেওয়া হয়।

```
>>> Print("Hello, How are you?")
```

আউটপুট:

```
Hello, How are you
```

Print() ফাংশন এর সিনট্যাক্স: Print() ফাংশন এর সিনট্যাক্স বা গঠন বা ফরম্যাট হল নিম্নরূপ:

```
Print(*Objects,sep=",",end='\n',file=sys.stdout,flush=False)
```

যেখানে-

#objects হলো সেই যাকে প্রিন্ট করতে হবে।

#sep হল একাধিক ভ্যালু মধ্যকার সেপারেটর।

১১.২ কীবোর্ড থেকে ইনপুট গ্রহন

পাইথন প্রোগ্রামিং এর ক্ষেত্রে প্রোগ্রাম দুইভাবে মান ইনপুট নেওয়া যায়। যথাঃ

১. সরাসরি

২. ইউজার হতে

সরাসরি: অন্যান্য প্রোগ্রামিং ল্যাংগুয়েজের মত পাইথনেও সরাসরি মান ইনপুট নেওয়া যায়। এজন্য ভেরিয়েবল ডিক্লেয়ারেশনের সময় অ্যাসাইনমেন্ট অপারেটর ব্যবহার করে মান ইনপুট নেওয়া যায়।

ইউজার হতে: পাইথনে ইউজারের মাধ্যমে কী-বোর্ড থেকে ইনপুই নেওয়া যায়।

১০.৩ ফাইল ও ফাইল অপারেশন

ফাইল : ফাইল বা ডেটা হচ্ছে স্টোরেজ ডিভাইসের এমন একটি স্পেস যেখানে স্থায়ী ভাবে তথ্য সংরক্ষন করা যায় এবং প্রয়োজনে তথ্য রিড করা যায়।

ফাইল অপারেশন:

- # ফাইলের নামকরণ করা
- # ফাইল খোলা
- # ফাইল থেকে ডাটা পড়া
- # ফাইল ডাটা লেখা
- # ফাইল বন্ধ করা

১০.৪ ফাইল খোলা ও ফাইল বন্ধ

ফাইল খোলা: সংরক্ষিত ডিক্লের ফাইল কিছু লিখতে বা ফাইল থেকে কিছু পড়তে হলে তার পূর্বে ফাইল ওপেন করতে হয়। ফাইল খোলার জন্য পাইথনে ওপেন() ফাংশন ব্যবহার করা হয়।

Open () ফাংশন: পাইথনে ফাইল অপেন করার জন্য () ফাংশন ব্যবহার করা হয়। সিনট্যাক্স:

```
file object=open (file_name [,access_mode][,buffering])
```

Open () ফাংশনে তিনটি প্যারামিটার আছে। যথা-

- #ফাইলের নাম
- #আক্সেস মোড
- # বাফারিং

ফাইল বন্ধ করা: ফাইল ওপেন করে রিড/রাইড করার পর ফাইল অবশ্যই বন্ধ করতে হবে । নাহলে এর কারনেই পাইথনের কাছে ফাইলটি ওপেন অবস্থায় থাকবে যা বন্ধত মেমোরী দখল করে রাখবে এবং প্রোগ্রামের পারফরমেন্সে খারাপ ভূমিকা রাখবে । ফাইল close করার জন্য পাইথনের শেষ() ফাংশন ব্যবহৃত হয় । close() ফাংশনের ফরম্যাট হলোঃ

FileName.close()

১১.৫ এরর হ্যান্ডলিং

ফাইল অপারেশনের সময় পাইথন প্রোগ্রামিং ল্যাংগুয়েজে লিখিত প্রোগ্রামে নানা কারণে নানান ধরনের error message বা error দেখায় ।

- ১.ফাইলের 'end of file' সঠিকভাবে delect না করতে পারা ।
- ২.ওপেন করা হয় নেই এমন কোন ফাইল ব্যবহারের চ্যাষ্টা করা ।
৩. অন্য কোনো operation এর জন্য open কৃত File এ operation চালানোর চেষ্টা করা ।
- 4.Writre protected file-এ কোনকিছু Write করার চেষ্টা করা ।
৫. Invalid file name যুক্ত file open করার চেষ্টা করা ।
৬. ফাইলে ডাটা সংরক্ষণের জন্য পর্যাপ্ত জায়গা বরাদ্দ না থাকা ।
৭. ফাইলের এক্সটেনশন নেইম ব্যবহারে ভুল করা ।
- ৮.রিড মোডের ফাইল রাইট মোডে রিড করার চেষ্টা করা ।
১০. ফাইল কোনো অবৈধ অপারেশনের চেস্টা করা ।

** ব্যবহারিক **

***Experiment Name:[Simple programs using basic *structure* of a programming language(python)]**

***Do not adopt unfair means in the exam.**

- Print(“welcome to python programming”)

***”Welcome to pythone programming” to print this massase:**

Print(“welcome to python programming”)

- দুইটি ইন্টিজার সংখ্যার যোগফল:

```
num1 = int(input('Enter first number '))
```

```
num2 = int(input('Enter second number '))
```

```
result = num1+num2
```

```
Print (“Result is “, result)
```

***দুইটি দশমিক বিশিষ্ট (ফ্লোটিং পয়েন্ট) সংখ্যা যোগফল নির্ণয়ের প্রোগ্রাম।**

```
num1 = float(input('Enter first number '))
```

```
num2 = float (input('Enter second number '))
```

```
result = num1+num2
```

```
Print ("Result is ", result)
```

***কোনো স্ট্রিং ইনপুট নিয়ে সেটি প্রদর্শনের প্রোগ্রাম।**

```
String = input ('Enter your String:')
```

```
Print ('You have entered:'String)
```

***দুটি সংখ্যাকে স্ট্রিং হিসেবে নিয়ে তাদের কনক্যাটেনেট প্রদর্শন প্রোগ্রাম।**

```
Number1 = input('Enter first number:')
```

```
Number2 = input('Enter second number:')
```

```
Result = Number1+Number2
```

```
Print('The result is=',result)
```

***দুটি স্ট্রিং নিয়ে তাদের কনক্যাটেনেট প্রদর্শন প্রোগ্রাম।**

```
String1 = input('Enter first number:')
```

```
String2 = input('Enter second number:')
```

```
Result = String1+String2
```

```
Print('The result is=',result)
```

***দুটি সংখ্যার ভাগশেষ নির্ণয়ের প্রোগ্রাম**

```
Number1 = input('Enter first number:')
```

```
Number2 = input('Enter second number:')
```

```
remainder = int(Number1 )% int(Number2)
```

```
Print('The result is=', remainder)
```

***দুটি সংখ্যার ভাগফল নির্ণয়ের প্রোগ্রাম**

```
Number1 = input('Enter first number:')
```

```
Number2 = input('Enter second number:')
```

```
division = float(Number1 )/ floatNumber2)
```

```
Print('The result is=', division)
```

***n সংখ্যক সংখ্যার যোগফল নির্ণয়ের প্রোগ্রাম।**

```
n=int(input("Enter a number:"))
```

```
If n<0:
```

```
    Print(Enter a positive number")
```

```
Else:
```

```
    sum=0
```

```
While(n>0):
```

```
    Sum=0
```

```
    sum+=n
```

```
    n-=1
```

```
    Print("The sum is=",sum")
```

***কতগুলো সংখ্যা ইনপুট নিয়ে তাদের গড় নির্ণয়ের প্রোগ্রাম।**

```
n=int(input("how many number you have ?"))
```

```
a=[]
```

```
for i in range (0,n):
```

```
    item=int(input(Enter intger number:"))
```

```
    a.append(item)
```

```
Avg=sum(a)/print("average of entered numbers is:",avg)
```

সেলসিয়াস তাপমাত্রাকে ফারেনহাইটে রূপান্তরের প্রোগ্রাম।

```
f=int(input("Enter temperature number in farnhite"))
```

```
C=(f-32)/1.8
```

```
Print("The valu is c")
```

ফারেনহাইট তাপমাত্রাকে সেলসিয়াসে রূপান্তরে প্রোগ্রাম।

```
F=float(input("Please Enter the farhanite temparature"))
```

```
c=(f-32)/1.8
```

```
Print('%0.3if deggre farhanite is =%0=35.0 deggre celcius'
```

কিলোমিটারকে মাইলে রূপান্তরের প্রোগ্রাম।

```
Kilometers=float(input("Enter value In kilometers;"))
```

```
Conv_fac=0.621371
```

```
Miles=kilomiters*Conv_fac
```

```
Print=Print('%0.3if Kilometers is =%0=35.0 Miles'%( Kilometers, Miles))
```

*আয়তক্ষেত্রের ক্ষেত্রফল নির্ণয়ের প্রোগ্রাম।

```
l=int(input("Enter the length in rectangular"))
```

```
w=int(input("Enter the width in rectangular"))
```

```
A=l*w
```

```
Print=("The area of rectangular is=a")
```

*বৃত্তের ক্ষেত্রফল নির্ণয়ের প্রোগ্রাম।

```
import math
```

```
r=float(input("Enter the radius of the circle;"))
```

```
a=math.p*r*r
```

```
Print=("area of this circle is =%.2f"area)
```

*বর্গক্ষেত্রের ক্ষেত্রফল নির্ণয়ের প্রোগ্রাম।

```
a=int(input("Enter the length in square"))
```

```
Area=a**2
```

```
Print=("The area of square is=Area")
```

*দিনকে মাস ও দিনে পরিবর্তনের প্রোগ্রাম।

```
python Sohan.py - C:\Users\ASUS\Documents\python Sohan.py (3.8.0)
File Edit Format Run Options Window Help
# A program to convert days to month and days
# writer: Moni Mukta Sohan

days=int(input("Enter The Days"))
month=days//30
day=days%30

print("Month is:",month, "Days:",day)
```

*সমকোণী ত্রিভুজক্ষেত্রের ক্ষেত্রফল নির্ণয়ের প্রোগ্রাম

```
*triangel.py - C:\Users\ASUS\Documents\triangel.py (3.8.0)*
File Edit Format Run Options Window Help
input math
a=int(input("Enter the arm of a triangel A"))
b=int(input("Enter the arm of a triangel B"))
c=int(input("Enter the arm of a triangel C"))

s=(a+b+c)/2
area=(s(s-a)(s-b)(s-c))

print("The area of the triangel is",area)
```

দুটি সংখ্যার মধ্যে বৃহত্তম সংখ্যা নির্ণয়ের প্রোগ্রাম

পাইথন প্রোগ্রামে - PowerPoint

INSERT DESIGN TRANSITIONS ANIMATIONS SLIDE SHOW REVIEW VIEW

```
triangel.py - C:\Users\ASUS\Documents\triangel.py (3.8.0)
File Edit Format Run Options Window Help
a=int(input("A="))
b=int(input("B="))

if a>b:
    print("Largest Number is",a)
if a<b:
    print("Largest Number is",b)
```

*দুটি সংখ্যার মধ্যে ছোট সংখ্যা নির্ণয়ের প্রোগ্রাম

```
triangel.py - C:\Users\ASUS\Documents\triangel.py (3.8.0)
File Edit Format Run Options Window Help
a=int(input("A="))
b=int(input("B="))

if a<b:
    print("Smaller Number is",a)
if a>b:
    print("Smaller Number is",b)|
```

*কোন সংখ্যা জোড় না বিজোড় নির্ণয়ের প্রোগ্রাম

```
even or odd.py - C:/Users/ASUS/Documents/even or odd.py (3.8.0)
File Edit Format Run Options Window Help
Number=int(input("Number="))
if (Number%2==0) :
    print("The Number is Even")
if (Number%2==1) :
    print("The Number is odd")
```

*একটি সংখ্যা ইনপুট নিয়ে সেটা পজেটিভ নেগেটিভ না শূন্যতা নির্ণয়ের প্রোগ্রাম

```
num=float(input("Enter a number"))
if num>=0:
    if num ==0:
        print("it is zero")
    else:
        print("it is a Positive number")
else:
    print("it is a Neagitive number")
```

*তিনটি সংখ্যার মধ্যে বৃহত্তম সংখ্যাটি নির্ণয়ের প্রোগ্রাম

```
num1 = int(input("Input number 1:"))
```

```
num2 = int(input("Input number 2:"))
```

```
num3 = int(input("Input number 3:"))
```

```
if num1 > num2 and num1 > num3:
```

```
    print ("Largest number is:" num1)
```

```
if num2 > num1 and num2 > num3:
```

```
    print ("Largest number is:" num2)
```

```
else:
```

```
    print ("Largest number is:" num3)
```

*কোন ইংরেজি সালের মান ইনপুট নিয়ে সেটা লিপ ইয়ার নাকি তা নির্ণয়ের প্রোগ্রাম

```
year= int(input("plwase Enter the valur of year."))
```

```
if((Year%4==0 and Year%100!=0)or (Year%400==0)):
```

```
    print ("This is not a leap year")
```

```
else:
```

```
    print ("This is not a leap year")
```

*শর্তসাপেক্ষে বিষমবাহু ত্রিভুজ ক্ষেত্রের ক্ষেত্রফল নির্ণয়ের প্রোগ্রাম

```
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)> and(c+a)> b):
    S=(a+b+c)/2
    Area= mathg.sqrt(s*(s-a)*(s-b)*(s-c))
else:
    print("The Triangle is not possible*")
```

*দ্বিঘাত সমীকরণের মূলসমূহ নির্ণয়ের প্রোগ্রাম।

```
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:"))
d=(b*b)-(4*a*c)
if (d==0):
    x=-b/(2*a)
    print ("Roots are real & equal & are:,"x,x")
elif (d>0):
    x1= (-b +math.sqrt(d))/(2*a)
    x2= (-b -math.sqrt(d))/(2*a)
    print ("Roots are real & equal & are:,"x1,x2")
else:
    print(" Roots are imaginary")
```

*কোন একটি সংখ্যা মৌলিক কিনা তা নির্ণয়ের প্রোগ্রাম

```
num = int (input("Enter a number:*))
```

```
if num > 1:
```

```
    for i in range (2,num):
```

```
        if (num %i)==0:
```

```
            print (num, "is not prime number")
```

```
            break
```

```
        else:
```

```
            print(num, "is a prime number")
```

```
else:
```

```
    print(num, "is a prime number")
```

*কোন সংখ্যার ফ্যাক্টোরিয়াল নির্ণয় প্রোগ্রাম।

```
num = int (input("Enter a number:"))
```

```
factorial=1
```

```
if num <0:
```

```
    print("sorry, factorial dose not exist for negative numbers")
```

```
elif num ==0:
```

```
    print("the factorial of 0 is 1")
```

```
else:
```

```
    for i in rang (1, num+1):
```

```
        factorial= factorial*i
```

```
    print("The factorial of", num , "is", factorial)
```

*ফিরোনাসিসি সিকোয়েন্স এর মান নির্ণয় এর প্রোগ্রাম

```
Terms= int(input("How many terms"))
n1=0
n2=1
count=0
if Terms <=0:
    Print ("please enter a positive number")
elif Terms==1:
    Print("Fibonacci Seqence upto",Terms,",")
    Print (n1)
else:
    Print("Fibonacci Seqence upto",Terms",)
    while count <Terms:
        Print (n1,end=',')
        nth=n1+n2
        n1=n2
        n2=nth
        count+=1
```

*গ্রেড নির্ণয়ের প্রোগ্রাম

While True:

```
print("Enter 'x' for exit")
print("Enter marks obtained in 5 subjects :")
mark1
mark2
mark3
mark4
mark5
if mark1 == 'x':
    break
else:
    sum = mark1 + mark2 + mark3 + mark4 + mark5
    average = sum/5
    if(average>91 and average<=100):
        print(" Your Grade is A +")
```

```
elif (average>=81 and average<=90):
```

```
    print("Your Grade is A")
```

```
elif( average>=71 and average<=80):
```

```
print(Your Grade is B+)
```

```
elif( average>=61 and average<=70):
```

```
    print(Your Grade is B)
```

```
elif( average>=51 and average<=60):
```

```
    print(Your Grade is c+)
```

```
elif( average>=41 and average<=50):
```

```
    print(Your Grade is c)
```

```
elif( average>=0 and average<=40):
```

```
    print(Your Grade is F)
```

*১ থেকে ১০০ পর্যন্ত সংখ্যাগুলোর মধ্যে জোড় সংখ্যাগুলো প্রদর্শনের প্রোগ্রাম

```
for i in range (2, 100, 2) :
```

```
Print (i)
```

*১ থেকে ১০ পর্যন্ত সংখ্যাগুলোর মধ্যকার মৌলিক সংখ্যাগুলো প্রদর্শনের প্রোগ্রাম

```
Lower = int(input("Enter lower range:"))
```

```
Upper = int(input("Enter upper range:"))
```

```
Print("prime numbers between",lower,"and",upper,"are,")
```

```
for num in range(lower , upper + 1):
```

```
    # Prime numbers are greater than 1
```

```
    If num>1:
```

```
        for in range(2,num):
```

```
            If(num % i)==0:
```

```
                break
```

```
        else:
```

```
            print(num)
```

*নিম্নোক্ত প্যাটার্ন প্রদর্শনের প্রোগ্রাম

```
for i in range(0,5):
```

```
    for j in range(0,i+0:
```

```
        Print("*",end= "")
```

```
    Print()
```

*নিম্নোক্ত প্যাটার্ন প্রদর্শনের প্রোগ্রাম

```
K = 1
```

```
for j in range(0,5):
```

```
    for j in range(0,k):
```

```
        Print("*", end = "")
```

```
K = K + 2
```

```
Print()
```

```
*
```

```
* *
```

```
* * *
```

```
* * * *
```

```
* * * * *
```

```
*
```

```
* * *
```

```
* * * * *
```

```
* * * * * * *
```

```
* * * * * * * *
```

***এক সেট ডাটাকে অ্যাসেনডিং অর্ডারে সাজানোর প্রোগ্রাম।**

```
listl.sort()
print("sort ascending:",listl)
listl.reverse()
print("sort descending:",listl)
```

***দুইটি ম্যাট্রিক্স এর গুনফল নির্ণয়ের প্রোগ্রাম।**

```
Matrix1={{10,11,12},
          {12,14,15},
          {16,17,18}}
Martix2={{1,2,3},
          {4,5,6},
          {7,8,9}}
rmatrix={{0,0,0},
          {0,0,0},
          {0,0,0}}
For l in range (len(matrix1)):
    For j in range (len(matrix1{0}))
        Rmatrix{i}{j}=matrix1 {i} {j}-matrix2[i][j]
For r in matrix:
Print(r)
```

*ফাংশন ব্যবহার করে ত্রিভুজের ক্ষেত্রফল নির্ণয়ের প্রোগ্রাম।

Def calculateArea():

```
import math
```

```
a=int(input("Enter the arm of a triangle"))
```

```
b=int(input("Enter the arm of a triangle"))
```

```
c=int(input("Enter the arm of a triangle"))
```

```
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("the area of the triangle isnot possible")
```

```
calculateArea()
```

*ফাংশন ব্যবহার করে কোন সংখ্যা মৌলিক কিনা তা নির্ণয়ের প্রোগ্রাম।

```
def test_prime(n):  
    if(n==1):  
        return false  
    elif(n==2):  
        return true;  
    else:  
        for x in range(2,n):  
            if(n%x==0):  
                return false  
        return true  
Print(test_prime(19))
```

*রিকার্সিভ ফাংশন ব্যবহার করে কোন সংখ্যার ফ্যাক্টোরিয়াল মান নির্ণয় প্রোগ্রাম

```
def recur_factorial(n):
```

```
    if n==1:
```

```
        else:
```

```
            return n*recurfacurial(n-1)
```

```
            num =int("input("Enter a number:"))
```

```
    if num<0:
```

```
        Print("Sorry,factorial does not exist for negative numbers")
```

```
    elif num==0:
```

```
        if num<0:
```

```
            print("The factorial of 0 is 1")
```

```
    else:
```

```
        Print("The,factorial of",num,"is",recur_factorial(num))
```

*ফাংশন ব্যবহার করে ৩টি সংখ্যার মধ্যে ক্ষুদ্রতম সংখ্যাটি নির্ণয়ের প্রোগ্রাম

```
def smallest( ):
```

```
    num1 float(input("Enter first number:"))
```

```
    num2 float(input("Enter second number:"))
```

```
    num3 float(input("Enter third number:"))
```

```
    If(num1<=num2)and(num1<=num3):
```

```
        Smallest num2
```

```
    else:
```

```
        smallest=num3
```

```
    Print("The smallest number among",num1,num2,"and",num3,"is" smallest)
```

```
smallest( )
```

*কোন ফাইলের যাবতীয় কন্টেন্ট পড়ার প্রোগ্রাম

```
Def file_read(fname):
```

```
    txt = open (fname)
```

```
    print(txt.read())
```

```
file_read('test.txt')
```

*কোন ফাইলের যাবতীয় কন্টেন্টকে লাইন বাই লাইনপড়ে এদেরকে কোন ভেরিয়েবলে সেটার করার প্রোগ্রাম

```
Def file_read(fname):
```

```
    with open (fname,"r") as myfile:
```

```
        data= myfile.readlines()
```

```
        print(data)
```

```
file_read('test.txt')
```

*কোন ফাইল বন্ধ আছে কি না তা জানার প্রোগ্রাম

```
F= open('abc.txt','r')
```

```
Print(f.closed)
```

```
f.close()print
```

```
(f.closed)
```

সকলকে ধন্যবাদ

