



প্রধান সূচিপত্র



অধ্যায়	অধ্যায়ের নাম	পৃষ্ঠা নং
অধ্যায় - ০১	প্রোগ্রামিং-এর মৌলিক ধারণা	
অধ্যায় - ০২	পাইথন প্রোগ্রামিং-এর মৌলিক ধারণা	
অধ্যায় - ০৩	ভেরিয়েবল ও ডাটা টাইপ	
অধ্যায় - ০৪	পাইথন অপারেটর	
অধ্যায় - ০৫	ব্রাঞ্চিং স্ট্রাকচার	
অধ্যায় - ০৬	লুপিং স্ট্রাকচার	
অধ্যায় - ০৭	লিস্ট স্ট্রাকচার	
অধ্যায় - ০৮	ট্যুপল স্ট্রাকচার	
অধ্যায় - ০৯	সেট স্ট্রাকচার	
অধ্যায় - ১০	ডিকশনারি স্ট্রাকচার	
অধ্যায় - ১১	ফাংশন অপারেশন	
অধ্যায় - ১২	ফাইল আই/ও অপারেশন	

অধ্যায়

০১

প্রোগ্রামিং-এর মৌলিক ধারণা

📖 এই অধ্যায়ের প্রধান সূচিপত্র (contents)

বিষয়	পৃষ্ঠা নং
❖ সূত্রাবলী ও শর্টকাট টেকনিক	
❖ অতি সংক্ষিপ্ত প্রশ্নাবলি	
❖ সংক্ষিপ্ত প্রশ্নাবলি	
❖ রচনামূলক প্রশ্নাবলি	

বোর্ড বোর্ড প্রশ্নাবলির বিশ্লেষণ (Board Questions Analysis) প্রশ্নাবলির বিশ্লেষণ (Board Questions Analysis)

এই অধ্যায় থেকে বিগত পাঁচ বছরের ডিপ্লোমা ইন ইঞ্জিনিয়ারিং, ডিপ্লোমা ইন টেক্সটাইল, ডিপ্লোমা ইন এগ্রিকালচার, ডিপ্লোমা ইন মেরিন পরীক্ষায় মোট প্রশ্ন। নিচে ‘Board Questions Analysis’ অংশে এই অধ্যায় থেকে কোন সালে কোন বোর্ডে কতটি প্রশ্ন হয়েছে তা দেখানো হলো।

সাল \ বোর্ড	অতি সংক্ষিপ্ত	সংক্ষিপ্ত প্রশ্ন	রচনামূলক প্রশ্ন
২০২৪	৩টি	—	২টি
২০২৩	—	—	১টি
২০২২	১টি	২টি	৩টি
২০২১	—	—	—



“যেকোন বোকাই এমন কোড লিখতে পারে যা কম্পিউটার বুঝে; ভালো প্রোগ্রামার সেই কোড লিখে যা মানুষও বুঝতে পারে।”

- Martin Fowler

অতি সংক্ষিপ্ত প্রশ্নোত্তর :

১. Flowchart কী?

[বাকশির্বো: '২১, '২৪]

অথবা, ফ্লোচার্ট বলতে কী বুঝায়?

[বাকশির্বো: '০৫, '০৭, '১২, '১৩, '১৪, '২০]

উত্তর: কোন নির্দিষ্ট সমস্যা সমাধানের যৌক্তিক ধাপসমূহের চিত্রভিত্তিক রূপকে ফ্লোচার্ট বলে।

২. কম্পাইলার কাকে বলে বা কম্পাইলারের কাজ কী?

[বাকশির্বো: '০৩, '১২, '১৩, '১৪, '১৭, '১৮, '২০R, '২২]

উত্তর: কম্পাইলার হলো এমন একটি Translator বা অনুবাদক প্রোগ্রাম যা হাই লেভেল

ল্যাংগুয়েজে লেখা সম্পূর্ণ প্রোগ্রামকে একসঙ্গে পড়ে মেশিন ল্যাংগুয়েজে রূপান্তর করে।

৩. মেশিন ল্যাংগুয়েজ প্রোগ্রাম কী?

[বাকশির্বো: '১৯]

উত্তর: কম্পিউটার মেশিনের ভাষা হচ্ছে ০ এবং ১। এই ০ এবং ১ কে ব্যবহার করে যখন

কম্পিউটারে ইন্সট্রাকশন প্রদান করে প্রোগ্রাম তৈরি করা হয়, তখন এ ধরনের প্রোগ্রামকে মেশিন ল্যাংগুয়েজ প্রোগ্রাম বলে।

৪. অনুবাদক প্রোগ্রাম বলতে কী বুঝায়?

[বাকশির্বো: '১৯]

অথবা, Translator প্রোগ্রাম কি?

উত্তর: যেসব প্রোগ্রাম ব্যবহার করে অ্যাসেম্বলি ল্যাংগুয়েজ ও হাই লেভেল ল্যাংগুয়েজ লিখিত সোর্স প্রোগ্রামকে মেশিন ল্যাংগুয়েজে রূপান্তর করা হয়, সেসব প্রোগ্রামকে অনুবাদক (Translator) বলে।

৫. অ্যালগরিদম কী?

[বাকশির্বো: '০১,০২, '০৪, '০৫,০৬,০৮, '০৯, '১০, '১৩, '১৩R, '১৪R, '১৯, '২৪]

উত্তর: কোন নির্দিষ্ট সমাধানের যৌক্তিক ক্রমানুযায়ী ধাপসমূহের লিখিত রূপকে অ্যালগরিদম বলে।

৬. প্রোগ্রামিং ল্যাংগুয়েজ বলতে কী বুঝায়? [বাকশির্বো: '০১, '০৪, '০৫, '০৯, '১০, '১৪, '১৭]

উত্তর: প্রোগ্রামিং হলো সমস্যা সমাধানের পদ্ধতিগত প্রক্রিয়া। কম্পিউটারের ভাষায় প্রোগ্রামিং হলো কোনো সমস্যা সমাধানের উদ্দেশ্যে পর্যায়ক্রমে সাজানো নির্দেশাবলির সমষ্টি। অর্থাৎ কোনো একটি নির্দিষ্ট কাজের জন্য কম্পিউটারের কোনো একটি ভাষায় লিখিত নির্দেশাবলি বা ইনস্ট্রাকশনের সমষ্টি হচ্ছে প্রোগ্রাম এবং তা তৈরি করার পদ্ধতি হচ্ছে প্রোগ্রামিং। আর যে ভাষার সাহায্যে প্রোগ্রাম তৈরি করা হয় তাকে প্রোগ্রামিং ল্যাংগুয়েজ বলে।

৭. কম্পিউটার প্রোগ্রামিং বলতে কী বুঝায়?

[বাকশির্বো: '১৪]

অথবা, Programming বলতে কী বোঝায়?

[বাকশির্বো: ১০R, ০৫, ১০, ১১, ১৩]

উত্তর:

প্রোগ্রামিং (Programming): প্রোগ্রামিং হলো যেকোন সমস্যা সমাধানের পদ্ধতিগত কৌশল বা প্রক্রিয়া। কম্পিউটারের পরিভাষায় প্রোগ্রামিং বলতে কোন সমস্যা সমাধানের উদ্দেশ্যে পর্যায়ক্রমে নির্দেশাবলি সাজানোর কৌশলকে বুঝায়। প্রোগ্রামিং-এর মাধ্যমে নির্দিষ্ট কোন সমস্যাকে অল্প সময়ে সহজে সমাধান করা যায়।

৮. কয়েকটি High level language এর নাম লেখ।

[বাকশির্বো: '০৯, '১৩R]

অথবা, কয়েকটি OOP Language এর নাম লেখ।

উত্তর: BASIC Beginners All purpose Symbolic Instruction Code COBOL Common Business Oriented Language Prolog Programming in Logic

C, C++, visual BASIC

৯. ইন্টারপ্রেটার কাকে বলে?

[বাকশির্বো: '০৩, '১২, ১৩, ১৩, ১৪, ১৭, '১৮, '২০R]

উত্তর: কিছু অনুবাদক প্রোগ্রাম আছে যারা হাইলেভেল ল্যাংগুয়েজে লেখা সম্পূর্ণ প্রোগ্রামকে একসঙ্গে না পড়ে বরং একটির পর একটি লাইনকে পড়ে মেশিন ল্যাংগুয়েজে রূপান্তর করে। এ ধরনের অনুবাদক প্রোগ্রামকে ইন্টারপ্রেটার বলে।

১০. সোর্স প্রোগ্রাম কি?

[বাকশিবো: '২৪]

উত্তর: সোর্স প্রোগ্রাম হলো প্রোগ্রামারের লেখা মূল কোড।

সংক্ষিপ্ত প্রশ্নোত্তর :

১. Flowchart-এর বৈশিষ্ট্যগুলো লেখ।

[বাকশিবো: '২১]

উত্তর: ফ্লোচার্টের বৈশিষ্ট্য (Characteristics of Flowchart):

- ১। ফ্লোচার্ট যতটা সম্ভব সহজ সরল করা হয়।
- ২। ফ্লোচার্ট এ অ্যালগরিদমের গ্রাফিক্যাল উপস্থাপন হবে।
- ৩। অ্যালগরিদমের গুরুত্বপূর্ণ সবকিছু ধাপ ফ্লোচার্টে থাকবে।
- ৪। ফ্লোচার্টের শুরু এবং শেষ থাকবে।
- ৫। টেস্ট ডাটা দিয়ে ফ্লোচার্ট পরীক্ষা করলে সঠিক আউটপুট পাওয়া যাবে।
- ৬। ফ্লোচার্ট তৈরির সময় অপ্রয়োজনীয় Loop এড়িয়ে চলা উচিত।

২. অ্যালগরিদমের বৈশিষ্ট্য লিখ।

[বাকশিবো: '১৪, '১৯, '২০, '২৪]

অথবা, অ্যালগরিদমের নিয়মগুলো লিখ।

[বাকশিবো: '১১]

উত্তর: একটি স্ট্যান্ডার্ড অ্যালগরিদমের কিছু বৈশিষ্ট্য বিদ্যমান থাকে যা নিচে আলোচনা করা হলো:

- (১) নির্দিষ্টতা: একটি নির্দিষ্ট কাজের জন্য অ্যালগরিদম তৈরি হতে হবে।
- (২) কার্যকরী প্রতিটি ধাপের একটি নির্দিষ্ট কাজ থাকবে।
- (৩) এক বা একাধিক আউটপুট: প্রত্যেক অ্যালগরিদমের এক বা একাধিক আউটপুট থাকবে।
- (৪) ইনপুট: প্রত্যেক অ্যালগরিদমে শূন্য বা তার চেয়ে বেশি ইনপুট থাকবে।

(৫) সমাপন: নির্দিষ্ট সংখ্যক কার্য সম্পাদন করে অ্যালগরিদম শেষ হবে।

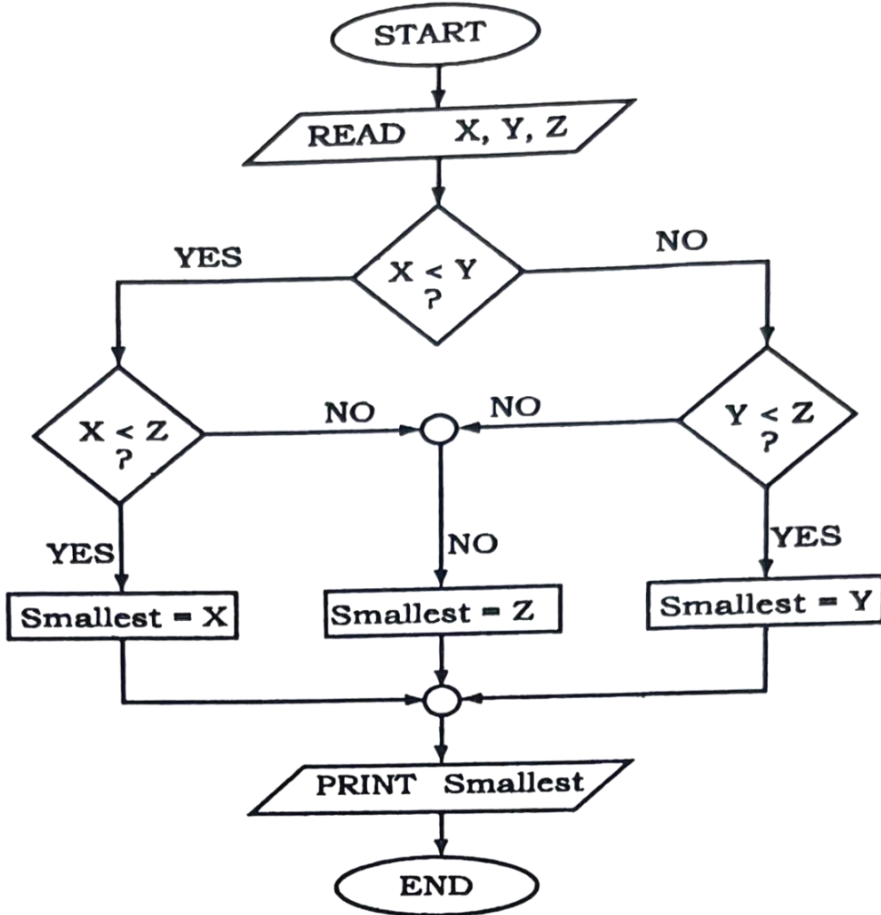
(৬) প্রয়োগ: একে ব্যাপকভাবে প্রয়োগকারী উপযোগী হতে হবে।

(৭) আউটপুট: প্রতিটি অ্যালগরিদমের সুনির্দিষ্ট আউটপুট থাকবে।

৩. তিনটি সংখ্যার মধ্যে সবচেয়ে ছোট সংখ্যাটি নির্ণয়ের ফ্লোচার্ট আঁক।

[বাকশিৰো: '০৪,০৫, ১৩, ১৪, '১৭, '১৮, '২২]

উত্তর: তিনটি সংখ্যার মধ্যে সবচেয়ে ছোট সংখ্যাটি নির্ণয়ের ফ্লোচার্টটি নিম্নে অঙ্কিত হলো।



চিত্র: তিনটি সংখ্যার মধ্যে সবচেয়ে ছোট সংখ্যাটি নির্ণয়ের ফ্লোচার্ট।

৪. কম্পাইলার (Compiler) এবং ইন্টারপ্রিটার (Interpreter) এর মাঝে পার্থক্য লেখ।

[বাকশিবি: '০৬, ০৭,০৯,১০, ১২,১৪,১৮]

উত্তর: কম্পাইলার ও ইন্টারপ্রিটারের পার্থক্য নিম্নে দেওয়া হলো:

কম্পাইলার (Compiler)	ইন্টারপ্রিটার (Interpreter)
কম্পাইলার হচ্ছে এমন এক ধরনের অনুবাদক, যা হাই লেভে ল্যাংগুয়েজে লিখিত পুরো প্রোগ্রামকে এক সাথে মেশিন ল্যাংগুয়েজে রূপান্তর করে, প্রোগ্রামের মধ্যে ভুলত্রুটি (Error/bug) থাকলে সংশোধন (Debug) করে এবং প্রয়োজনীয় সহায়ক ফাইল যুক্ত করে এক্সিকিউটেবল (.exe) ফাইলে রূপান্তরিত করে।	যে অনুবাদক প্রোগ্রাম হাই লেভেল ল্যাংগুয়েজে লিখিত প্রোগ্রামের সোর্স কোডকে লাইন বাই লাইন বা এক লাইন করে মেশিন ল্যাংগুয়েজে রূপান্তর করে, কোনো ভুল থাকলে তা সংশোধন করে পরবর্তী লাইনে কাজ করে তাকে ইন্টারপ্রিটার (Interpreter) বলে।
এটা দ্রুতগতিসম্পন্ন।	এটা ধীরগতিসম্পন্ন।
একসাথে সকল ইনস্ট্রাকশনকে মেশিন ল্যাংগুয়েজে রূপান্তর/অনুবাদ করে থাকে।	এক সাথে কেবলমাত্র একটি ইনস্ট্রাকশনকে মেশিন ল্যাংগুয়েজে অনুবাদ করে থাকে।
কম্পাইল করা কোনো প্রোগ্রামের পরিবর্তন করতে হলে সম্পূর্ণ পদ্ধতিটির পুনরাবৃত্তি করতে হবে।	ইন্টারপ্রিটারের ক্ষেত্রে কোনো নির্দেশের পরিবর্তন করতে হলে সম্পূর্ণ পদ্ধতিটির পুনরাবৃত্তি করতে হয় না।
একটি কম্পাইলার কেবলমাত্র একটি ভাষাকে অনুবাদ করতে পারে।	একটি ইন্টারপ্রিটার একাধিক ভাষাকে অনুবাদ করতে পারে।

৫. অ্যালগরিদম ও ফ্লোচার্টের মাঝে পার্থক্যগুলো লেখ।

[বাকশির্বো: '০৭, ১১, ১৭]

উত্তর: অ্যালগরিদম ও ফ্লো-চার্টের পার্থক্য নিম্নরূপ:

অ্যালগরিদম	ফ্লোচার্ট
১. লিখিত নির্দেশের পর্যায়ক্রমিক ধাপসমূহকে অ্যালগরিদম বলে।	১. অ্যালগরিদমের চিত্ররূপ হলো ফ্লোচার্ট।
২. অ্যালগরিদম বর্ণনামূলক।	২. ফ্লোচার্ট চিত্রমূলক।
৩. অ্যালগরিদমে কোন চিত্রাংশ ব্যবহার করা হয় না।	৩. ফ্লোচার্টে চিত্রাংশ ব্যবহার করতে হয়।
৪. অ্যালগরিদম হলো সমস্যার সিদ্ধান্ত গ্রহণ।	৪. ফ্লোচার্ট হলো চিত্রের মাধ্যমে প্রোগ্রামের গতি প্রবাহ।

রচনামূলক প্রশ্নোত্তর :

১. প্রোগ্রাম পরিকল্পনার ধাপসমূহ বর্ণনা কর।

[বাকাশিবো: '২১, '২২, '২৩, '২৪]

অথবা, প্রোগ্রাম প্ল্যানিং-এর ধাপগুলো বর্ণনা কর।

[বাকাশিবো: '১৭, '১৮, '২০]

উত্তর: সুষ্ঠু পরিকল্পনা ছাড়া কোনো কাজই সুসম্পন্ন হয় না। তাই কোনো প্রোগ্রাম লিখতে হলে সুনির্দিষ্ট কতকগুলো ধাপে ধাপে সুষ্ঠু পরিকল্পনার মাধ্যমে প্রোগ্রামটি সম্পন্ন হয়। প্রোগ্রাম পরিকল্পনার প্রসেস বা ধাপসমূহ নিম্নরূপ-

- (i) সমস্যা চিহ্নিত করা (Problem identification)
- (ii) সিস্টেম বিশ্লেষণ (System analysis)
- (iii) প্রোগ্রামের ইনপুট-আউটপুট নির্ণয় (Input-output)
- (iv) অ্যালগরিদম উন্নয়ন (Algorithm development)
- (v) ফ্লোচার্ট উন্নয়ন (Flow chart development)
- (vi) প্রোগ্রামিং ভাষা নির্ধারণ (Programming language)
- (vii) প্রোগ্রাম রচনা (Program coding)
- (viii) প্রোগ্রাম কম্পাইলেশন (Program compilation)
- (ix) প্রোগ্রাম পরীক্ষণ ও ত্রুটি সংশোধন (Program testing and debugging)
- (x) ডকুমেন্টেশন (Documentation)
- (xi) ইনস্টলেশন (Installation)
- (xii) সংরক্ষণ ও রক্ষণাবেক্ষণ (Maintenance) ।

সমস্যা চিহ্নিত করা (Problem identification): প্রোগ্রাম পরিকল্পনার প্রথম ধাপ হচ্ছে সমস্যা চিহ্নিত করা। অর্থাৎ ব্যবহারকারীর চাহিদা অনুযায়ী কোন সমস্যার জন্য প্রোগ্রাম লিখতে হবে তা চিহ্নিত করা।

সিস্টেম বিশ্লেষণ (System analysis): সমস্যা চিহ্নিত করার পর সমস্যা সম্পর্কিত বিভিন্ন তথ্য সংগ্রহ করে তা বিশ্লেষণ করতে হবে, যা প্রোগ্রামিং পরিকল্পনার পরবর্তী ধাপসমূহে ব্যবহার করা যাবে।

প্রোগ্রামের ইনপুট/আউটপুট নির্ণয় (Input-output): এ পর্যায়ে সমস্যা সমাধানের জন্য প্রয়োজনীয় ইনপুট ও আউটপুটসমূহ নির্ণয় করা হয়। প্রোগ্রামে কোন কোন ডাটা বা তথ্য ইনপুট হিসেবে ব্যবহার করা হবে এবং উল্লিখিত ডাটার জন্য কী কী আউটপুট আসবে, এর জন্য কোন কোন ফর্মুলা বা সূত্রের ব্যবহার প্রয়োগ করতে হবে তা নির্ধারণ করে দেয়া হয়।

প্রোগ্রামের অ্যালগরিদম উন্নয়ন করা (Algorithm development): কোনো সমস্যা সম্পর্কে পূর্ণ ধারণা এবং তার সমাধানের সূত্র পাওয়া গেলে পরবর্তী কাজ হলো সমস্যা সমাধানের ক্রমানুযায়ী ধারাবাহিক ধাপসমূহকে লিখিতরূপে প্রকাশ করা, অর্থাৎ প্রোগ্রামটির জন্য অ্যালগরিদম উন্নয়ন করা।

ফ্লোচার্ট তৈরি করা (Flow chart development): একটি প্রোগ্রাম তৈরি করার পূর্বে প্রোগ্রামটি পর্যায়ক্রমে একের পর এক কীভাবে কাজ করবে সেই ধাপগুলো চিত্রের বা ছবির মাধ্যমে অথবা ড্রয়িং-এর মাধ্যমে প্রকাশ করাকে ফ্লোচার্ট বলে। প্রোগ্রাম লেখার আগে প্রোগ্রামটি কীভাবে কাজ করবে সেই সম্পর্কে স্বচ্ছ ধারণা থাকতে হবে। অনেক সময় প্রোগ্রামের আকার বেশি বড় হলে ফ্লোচার্ট তৈরি করে নিলে 'কাজ করতে সুবিধা হয়। গ্রাফ দেখে যেমন কয়েক বছরের ডাটা বা তথ্য সম্পর্কে পরিষ্কার ধারণা পাওয়া যায়, তেমনি ফ্লোচার্ট দেখে প্রোগ্রামের মূল কার্যকারিতা অতি সহজে বুঝা যায়। এতে প্রোগ্রাম অন্য কেউ বিশ্লেষণ করতে পারে, প্রোগ্রামিং করতে সুবিধা হয় এবং সময় বাঁচে। প্রোগ্রামিং করার সময় সমস্যার সৃষ্টি হলে ফ্লোচার্ট দেখে সমস্যার সমাধান করা যায়। দক্ষ প্রোগ্রামাররা সবসময় ফ্লোচার্ট তৈরি করে প্রোগ্রাম তৈরি করেন।

প্রোগ্রামিং ভাষা নির্ধারণ (Programming language): কখনো কখনো একটি বিশেষ প্রোগ্রামিং ভাষার পক্ষে হয়ত-বা বিশেষ কোনো কাজ করা সম্ভব নাও হতে পারে। সেক্ষেত্রে অন্য কোনো ভাষার সহায়তা নেয়া যেতে পারে। উল্লেখ্য, কোনো কোনো সফটওয়্যার কোম্পানি নির্দিষ্ট কোনো কাজের জন্য বিশেষ কোনো প্রোগ্রামিং ভাষাকে প্রাধান্য দিয়ে থাকে। অনেক সময় প্রোগ্রামিং ভাষাটি কঠিন পদ্ধতিতে যেমন

মেশিন কোডে নির্বাচন না করে অ্যাসেম্বলারের সাহায্য নেয়ার পরামর্শ দেয়া হয়ে থাকে, অথবা কোনো হাই লেভেল ভাষা নির্বাচন করতে বলা হয়ে থাকে।

প্রোগ্রাম রচনা করা (Program coding): এ ধাপে প্রোগ্রামিং লেখার জন্য যে নির্দিষ্ট ভাষাটি নির্বাচন করা হয়েছে সেই প্রোগ্রামিং ল্যাংগুয়েজ ব্যবহার করে প্রোগ্রাম রচনা করা হয়।

কম্পাইল করা (Program compilation): হাই লেভেল ল্যাংগুয়েজে রচিত প্রোগ্রামটি কম্পিউটার বুঝতে পারে না। তাই হাই লেভেল ল্যাংগুয়েজে রচিত প্রোগ্রামটিকে মেশিন ল্যাংগুয়েজে রূপান্তর করে দিতে হয়, অর্থাৎ কম্পাইল করতে হয়। প্রোগ্রাম কম্পাইল করার সময় প্রচুর কম্পাইলিং এরোর (ভুল) খুঁজে পাওয়া যায়। কম্পাইলারের নির্দেশ মোতাবেক কাজ করলে প্রোগ্রাম কম্পাইলিং-এর কাজটি নিখুঁতভাবে করা যায়। এ সমস্ত টেস্টগুলো প্রোগ্রামের যথার্থতা বা নির্ভুলতা বিচারে যথেষ্ট সহায়ক হয়ে থাকে।

ডকুমেন্টেশন (Documentation): ডকুমেন্টেশন হচ্ছে প্রোগ্রাম পরিকল্পনার বিভিন্ন ধাপসমূহ যেমন- অ্যালগরিদম, ফ্লোচার্ট, প্রোগ্রাম কোডিং, বাগ, ডিবাগ, ফলাফল ও ব্যবহারকারীর জন্য প্রয়োজনীয় নির্দেশ ইত্যাদির লিখিত বিবরণ ডকুমেন্ট আকারে লিপিবদ্ধ করা, যাতে পরবর্তীতে কোনো সমস্যা সৃষ্টি হলে তা সমাধান করা সহজ হয়।



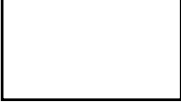
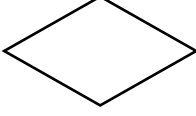
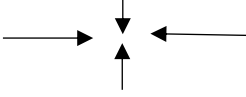

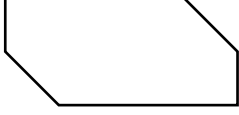
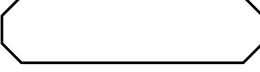
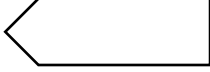
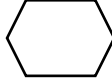

ইনস্টলেশন (Installation): পরীক্ষার সন্তোষজনক ফল লাভের পর প্রোগ্রামসমূহকে প্রোডাকশন লাইব্রেরিতে লোড করা হয়। এ লাইব্রেরিকে ইনস্টলেশন বলা হয়। অনেক ক্ষেত্রে এ ইনস্টলেশন অনেক জটিলও হতে পারে, তবে দলীয়ভাবে করলে এ পর্যায়ে ভালো ফল পাওয়া সম্ভব।

রক্ষণাবেক্ষণ (Maintenance): রক্ষণাবেক্ষণ বলতে প্রয়োজনীয় নানা কারণে প্রোগ্রামের ছোটোখাটো পরিবর্তন ও পরিবর্ধনকে বুঝায়। যে-কোনো সফটওয়্যার যখন বাজারে আসে তার একটি ভার্সন বাজারে আসে। প্রতিনিয়ত সফটওয়্যারগুলোর ভার্সন সংখ্যা উত্তরোত্তর বৃদ্ধি পেতে থাকে। সফটওয়্যারকে নানাবিধ কারণে পরিবর্তন, পরিবর্ধন করতে হয় এবং উন্নত সংস্করণে রূপ দিতে হয়।

২. ফ্লোচার্টের প্রতীকগুলো ঐকে এদের কাজ লেখ।

[বাকশিভো: '০৩, '০৪,০৫,০৬,০৯, '১৪R,'২১]

উত্তর: প্রোগ্রাম ফ্লোচার্টের বিভিন্ন প্রতীকের নাম, প্রতীক ও কাজ নিম্নরূপ-

প্রতীক/সিম্বলের নাম	প্রতীক	কাজ
আরম্ভ/সমাপ্তি চিহ্ন		ফ্লোচার্ট শুরু ও শেষ করার জন্য এই সিম্বল ব্যবহৃত হয়।
ইনপুট/আউটপুট চিহ্ন		কোনো মান ইনপুটে নেয়া কিংবা কোনো মান আউটপুটে প্রদর্শনের জন্য এই চিহ্ন ব্যবহৃত হয়।
প্রসেস/প্রক্রিয়াকরণ চিহ্ন		বিভিন্ন ধরনের গাণিতিক ও যৌক্তিক অপারেশন সম্পন্ন করার কাজে এই সিম্বল ব্যবহৃত হয়।
সিদ্ধান্ত গ্রহণ চিহ্ন		শর্ত বা কন্ডিশন সম্পর্কিত কার্যাদি সম্পাদনের ক্ষেত্রে এই সিম্বল ব্যবহৃত হয়।
নির্দেশনা চিহ্ন		প্রোগ্রাম ফ্লো কোন দিকে যাবে তা বুঝানোর জন্য এই সিম্বল ব্যবহৃত হয়।
সংযোগ চিহ্ন		ফ্লোচার্টের একাধিক অংশকে সংযুক্ত করার জন্য এই সিম্বল ব্যবহৃত হয়।
প্রিন্ট/ডকুমেন্ট চিহ্ন		প্রোগ্রামের কোনো অংশকে প্রিন্ট করার প্রয়োজন বুঝাতে এই সিম্বল ব্যবহৃত হয়।
প্রস্তুতি চিহ্ন		ফ্লোচার্টের ধারাবাহিকতা বজায় রেখে সংযোজন ও বিয়োজন কার্য সম্পাদনের জন্য এই সিম্বল ব্যবহৃত হয়।
ডিসপ্লে/প্রদর্শন চিহ্ন		ফ্রিনের ফলাফল প্রদর্শনের কাজে এই সিম্বল ব্যবহৃত হয়।
লুপ চিহ্ন		লুপিং স্টেটমেন্টসমূহকে লেখার জন্য এই সিম্বল ব্যবহৃত হয়।
সাবরুটিন/সাবপ্রোগ্রাম চিহ্ন		প্রোগ্রামে সাবরুটিন/সাবপ্রোগ্রাম থাকলে সেগুলো এই চিহ্ন ব্যবহার করে লেখা হয়

৩. দ্বিঘাত সমীকরণের মূল বের করার Algorithm ও Flowchart লেখ। [বাকশিৰো: '২০R, '২২]

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

উত্তর: 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

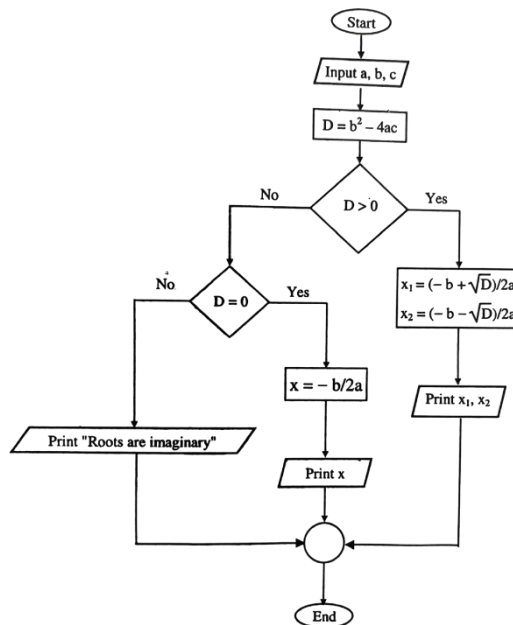
$x_2 = (-b - \sqrt{D})/2a$ নির্ণয় যদি $D = 0$ হয় তবে

(b) $x = -b/2a$ নির্ণয়

অন্যথায় "Roots are imaginary" বার্তাটি প্রদর্শন।

ধাপ-৫: প্রোগ্রাম শেষ (End)।

ফ্লোচার্ট (Flowchart) :



৪. সেন্টিগ্রেড থেকে ফারেনহাইট তাপমাত্রা রূপান্তরের জন্য ফ্লো-চার্টসহ প্রোগ্রাম লেখ।

[বাকশিৰো: '২৪]

উত্তর:

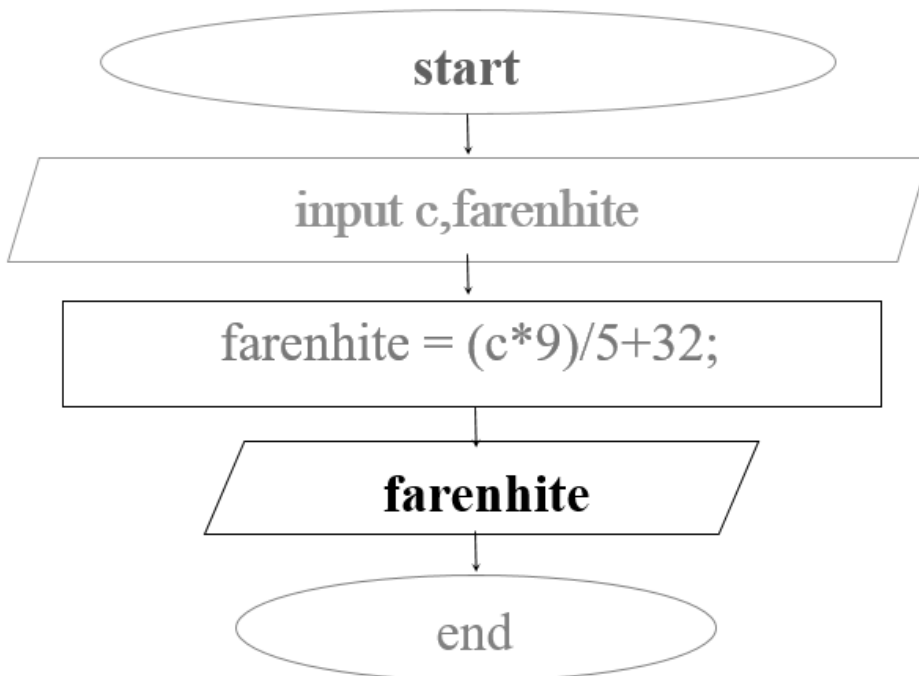
প্রোগ্রাম;

```
celsius = float(input("সেন্টিগ্রেড তাপমাত্রা লিখুন: "))
```

```
fahrenheit = (celsius * 9/5) + 32
```

```
print("ফারেনহাইট তাপমাত্রা:", fahrenheit, "°F")
```

ফ্লো-চার্ট;



৫. $১ + ৩ + ৫ + ৭ + \dots + ৯৯$ সিরিজটির সমষ্টি নির্ণয়ের পাইথন প্রোগ্রাম লেখ।

[বাকশির্বো: '২২, '২৩]

উত্তর:

$১ + ৩ + ৫ + ৭ + \dots + ৯৯$ সিরিজের সমষ্টি বের করার প্রোগ্রাম

sum = 0

for i in range(1, 100, 2): # ১ থেকে ৯৯ পর্যন্ত বিজোড় সংখ্যা

sum = sum + i

print("সিরিজটির সমষ্টি হলো:", sum)

অধ্যায়

০২

পাইথন প্রোগ্রামিং-এর মৌলিক ধারণা

২.১ অধ্যায়ের প্রধান সূচিপত্র (contents)

বিষয়	পৃষ্ঠা নং
❖ সূত্রাবলী ও শর্টকাট টেকনিক	
❖ অতি সংক্ষিপ্ত প্রশ্নাবলি	
❖ সংক্ষিপ্ত প্রশ্নাবলি	
❖ রচনামূলক প্রশ্নাবলি	

বোর্ড বোর্ড প্রশ্নাবলির বিশ্লেষণ (Board Questions Analysis) প্রশ্নাবলির বিশ্লেষণ (Board Questions Analysis)

এই অধ্যায় থেকে বিগত পাঁচ বছরের ডিপ্লোমা ইন ইঞ্জিনিয়ারিং, ডিপ্লোমা ইন টেক্সটাইল, ডিপ্লোমা ইন এগ্রিকালচার, ডিপ্লোমা ইন মেরিন পরীক্ষায় মোট প্রশ্ন। নিচে ‘Board Questions Analysis’ অংশে এই অধ্যায় থেকে কোন সালে কোন বোর্ডে কতটি প্রশ্ন হয়েছে তা দেখানো হলো।

সাল \ বোর্ড	অতি সংক্ষিপ্ত	সংক্ষিপ্ত প্রশ্ন	রচনামূলক প্রশ্ন
২০২৪	—	২টি	—
২০২৩	১টি	—	—
২০২২	১টি	১টি	—
২০২১	—	—	—



“ কম্পিউটার এমন একটি যন্ত্র যা তোমার চিন্তাকে দ্রুত করে, কিন্তু সবসময় ভালো করে না। ”

- বিল গেটস (Bill Gates)

অতি সংক্ষিপ্ত প্রশ্নোত্তর :

১. Keyword কী?

[বাকশির্বো: '২০R, '২২]

অথবা, কীওয়ার্ড বলতে কী বোঝায়?

উত্তর: কীওয়ার্ড হলো কতগুলো সংরক্ষিত শব্দ, কম্পাইলারের নিকট যাদের বিশেষ অর্থ রয়েছে এবং যাদেরকে কোনো ভেরিয়েবল, কনস্ট্যান্ট, ফাংশন, মডিউল কিংবা অবজেক্টের নাম হিসেবে ব্যবহার করা যায় না।

২. পাইথনে আইডেন্টিফায়ার বলতে কী বোঝায়?

[বাকশির্বো: '১৮, '২০]

অথবা, Identifiers বলতে কী বোঝায়?

[বাকশির্বো: '২০]

উত্তর: পাইথনে ব্যবহৃত ভেরিয়েবল, কনস্ট্যান্ট, ক্লাস, মডিউল, ফাংশন বা অন্যান্য সকল অবজেক্ট আইডেন্টিফাই করার জন্য যে নাম বা শব্দ ব্যবহার করা হয়, তাই পাইথন আইডেন্টিফায়ার।

৩. পাইথনে কোটেশন চিহ্ন ব্যবহার করা হয় কেন?

[বাকশির্বো: '১৯]

উত্তর: পাইথনে স্ট্রিং প্রকাশের জন্য একটি ('), দুটি (") কিংবা তিনটি ("" বা """) কোটেশন চিহ্ন ব্যবহার করা হয়।

৪. ইনডেন্টেশন বলতে কী বোঝায়?

[বাকশির্বো: '১৭]

উত্তর: কোন একটি কোডের আগে পর পর চারটি অথবা আটটি ক্যারেক্টার জায়গা ফাঁকা রাখাকে Indentation বলে।

৫. পাইথনের জনক কে?

উত্তর: পাইথনের জনক হলেন নেদারল্যান্ডের সিডব্লিউআই'র গবেষক Guido Van Rossum

৬. পাইথন কবে প্রথম রিলিজ করা হয়?

উত্তর: ১৯৮৯ সালের ডিসেম্বর মাসে বড়দিনের ছুটিতে পাইথন তৈরি করা শুরু হয় এবং ১৯৯১ সালে এটি প্রথম পাবলিশ করা হয়।

৭. পাইথনের মোট ভার্সন কয়টি ও কী কী?

উত্তর: পাইথনের দুটি ভার্সন রয়েছে, যথা পাইথন ২.x এবং পাইথন ৩.x।

৮. পাইথনে ব্যবহৃত কীওয়ার্ড সংখ্যা কত?

উত্তর: পাইথনে (ভার্সন-৩) মোট ৩৩টি কীওয়ার্ড ব্যবহৃত হয়।

৯. কখন ইন্ডেন্টেশন প্রয়োজন হয়?

উত্তর: সাধারণত: While loop/for loop বা if statement সহ বিভিন্ন স্টেটমেন্ট Indentation এর প্রয়োজন হয়।

১০. পাইথনে ব্যবহৃত যে-কোনো ছয়টি কীওয়ার্ডের নাম লেখ।

উত্তর: পাইথনে ব্যবহৃত ছয়টি কীওয়ার্ড হলো-and, or, not, if, else, for, while ইত্যাদি।

১১. পাইথন কি?

[বাকশির্বো: '২৩]

উত্তর: পাইথন হলো একটি উচ্চ স্তরের (High-Level), ইন্টারপ্রেটেড (Interpreted) এবং সাধারণ উদ্দেশ্যপ্রণোদিত (General-Purpose) প্রোগ্রামিং ভাষা।

সংক্ষিপ্ত প্রশ্নোত্তর :

১. পাইথনের বৈশিষ্ট্যগুলো লেখ।

[বাকশির্বো: '২১, '২৪]

অথবা, ল্যাংগুয়েজ হিসেবে পাইথন এত জনপ্রিয় কেন?

উত্তর: ল্যাংগুয়েজ হিসেবে পাইথন জনপ্রিয় হবার কারণগুলো নিম্নরূপ:

- পাইথন একটি সহজবোধ্য প্রোগ্রামিং ল্যাংগুয়েজ।
- পাইথনে কোডিং করা খুব সহজ।
- এর কোডিং সি বা জাভার চেয়ে তুলনামূলক অনেক ছোট হয়।
- পাইথনকে বিনামূল্যে ডাউনলোড করা যায় ও সহজে ইনস্টল করা যায়।
- পাইথনে লিস্ট, ডিকশনারি ও সেটের মতো চমৎকার সব ডাটা স্ট্রাকচার রয়েছে।
- বিভিন্ন অপারেটিং সিস্টেমে পাইথন ব্যবহার করা যায়।
- বিশাল ও কার্যকর স্ট্যান্ডার্ড লাইব্রেরি রয়েছে।
- শক্তিশালী অনলাইন কমিউনিটি।
- চমৎকার ওয়েব ফ্রেমওয়ার্ক (জ্যাঙ্গো, ফ্লাস্ক ইত্যাদি) বিদ্যমান।
- উইন্ডোজ ছাড়া মোটামুটি অন্য সব অপারেটিং সিস্টেমে (যেমন, লিনাক্স ও ম্যাক অপারেটিং সিস্টেম) পাইথন বিল্ট-ইন থাকে অর্থাৎ আলাদাভাবে ইনস্টল করতে হয় না।

২. পাইথনে কमेंট এর ব্যবহার দেখাও।

[বাকশির্বো: '১৯, '২০]

উত্তর: কमेंটস (Comments in Python): হ্যাশ চিহ্ন (#) ব্যবহার করে যদি কোন বাক্য লেখা হয়, তবে সেটি 'কमेंট হিসেবে গণ্য হয়। সহজ কথায়, # চিহ্নের পরে ব্যবহৃত একই লাইনের সকল অক্ষর একসাথে একট কमेंট হিসেবে বিবেচিত হবে, এবং Python interpreter সব ধরনের কमेंটকে ignore করবে। যেমন:

```
#This is an example of Comment
```

```
print "Hello, Python!"
```

এই স্ক্রিপ্টটি রান করলে নিম্নোক্ত ফলাফল প্রদান করবে

Hello, Python!

যেকোনো স্টেটমেন্ট, কিংবা এক্সপ্রেশন এর পরেও কমেন্ট সংযোজন করা যায়।

```
name "Muzahid" #This is a comment also
```

প্রয়োজন হলে প্রোগ্রামে একাধিক কমেন্টও যোগ করা যায়। যেমন:

```
#This is a comment.
```

```
#This is a comment, too
```

```
#Hello! I am Imrul Kayes
```

৩. পাইথনে আইডেন্টিফায়ার নামকরণের নিয়মাবলি উল্লেখ কর। [বাকাশির্বা: '১৯, '২২, '২৪]

অথবা, পাইথন আইডেন্টিফায়ার তৈরির নিয়মাবলি লেখ। [বাকাশির্বা: '১৭, '১৮]

উত্তর: পাইথনে আইডেন্টিফায়ার নামকরণের নিয়মগুলো হলো:

- আইডেন্টিফায়ারের নাম 'A' থেকে 'Z' অথবা 'a' থেকে 'z' অথবা underscore () দিয়ে শুরু হয় এবং এরপর যে-কোনো অক্ষর, underscore কিংবা সংখ্যা (0-9) ব্যবহৃত হয়।
- Python এর Identifier হিসেবে @, \$, % ইত্যাদি বিশেষ চিহ্নগুলো ব্যবহার করা যায়না।
- Python কেইজ সেন্সিটিভ ল্যাংগুয়েজ, অর্থাৎ A এবং a এখানে আলাদা অর্থ বহন করে। যেমন, Python identifier-এ Computer এবং coputer-এর দুটি শব্দ আলাদা অর্থ বহন করে এবং তারা ভিন্ন ভিন্ন আইডেন্টিফায়ার।
- শুধুমাত্র Class Name-গুলো বড় হাতের অক্ষর দিয়ে শুরু হয়, অন্য সবকিছুর নামকরণ শুরু হয় ছোট হাতের অক্ষর দিয়ে।

- কোন Identifier যদি শুধুমাত্র একটি underscore () দিয়ে শুরু হয় তবে তার অর্থ হচ্ছে এটি একটি "Private Identifier".
- যদি কোনো Identifier পর পর দুটি underscore () দিয়ে শুরু হয় তবে সেটি "Strongly Private Identifier".
- যদি কোনো Identifier পর পর দুটি underscore() দিয়ে শেষ হয়, তবে তাকে "Language-Defined Special Name" বলে।

৪. পাইথন প্রোগ্রাম স্ট্রাকচারের উল্লেখযোগ্য অংশগুলো কী কী?

উত্তর: পাইথন প্রোগ্রাম স্ট্রাকচারের উল্লেখযোগ্য অংশসমূহ হচ্ছে-

- ইমপোর্ট স্টেটমেন্টস (Import Statements)
- ফাংশন ও ক্লাস ডেফিনিশন (Function & Class Definition)
- মডিউল স্কোপ ভেরিয়েবলস (Module Scope Variables) এবং
- মেইন প্রোগ্রাম (Main Program) ।

৫. এক নজরে পাইথন ব্যাখ্যা কর।

উত্তর: এক নজরে পাইথন নিম্নরূপ:

সর্বপ্রথম প্রকাশ	২০ ফেব্রুয়ারি, ১৯৯১
প্যারাডাইম	মাল্টি প্যারাডাইম
জনক	গুইডো ভ্যান রোয়াম
ডেভলপার	পাইথন সফটওয়্যার ফাউন্ডেশন
অধুনা প্রকাশ	৩.৬.৪/১৬ অক্টোবর ২০১৭
টাইপিং ডিসিপ্লিন	কঠোর, চলমান ("ডাক টাইপিং")

প্রধান বাস্তবায়নসমূহ	সিপাইথন, জাইথন, আয়রন পাইথন, পাইপাই
যার দ্বারা প্রভাবিত	এবিসি, সি, হ্যাস্কেল, আইকন, লিম্প, মডুলা-৩ পার্লজাভা
বৈধপত্র	পাইথন সফটওয়্যার ফাউন্ডেশন লাইসেন্স
ওয়েবসাইট	http://www.python.org/

রচনামূলক প্রশ্নোত্তর :

১. পাইথনে ব্যবহৃত লাইন, মার্জিন, মাল্টি-লাইন স্টেটমেন্ট (Lines, Indentation, Multi-Line Statements in Python) বর্ণনা কর।

উত্তর: স্টেটমেন্ট (Statement): পাইথনের একটি স্টেটমেন্ট হলো একটি লজিক্যাল ইনস্ট্রাকশন, যা পাইথন ইন্টারপ্রেটার Read এবং Execute করতে পারে। অর্থাৎ পাইথনে ব্যবহৃত প্রতিটি expression এক একটি স্টেটমেন্ট।

এক্সপ্রেশন (Expression): এক্সপ্রেশন হলো একপ্রকার পাইথন স্টেটমেন্ট, যা numbers, strings, objects এবং operators-সমূহের একটি লজিক্যাল ক্রম ধারণ করে। এক্সপ্রেশন ব্যবহার করে যোগ, বিয়োগ, Concatenation-সহ অনেক কাজ করতে পারি।

লাইন (Line): পাইথন প্রম্পট (>>>)-এর ডানে লিখিত প্রতিটি স্টেটমেন্টকেই লাইন বলে। যেমন-
Line

মার্জিন (Indentation): কোনো একটি কোডের আগে স্পেস দেওয়া বা খালি জায়গা রাখাকে Indentation বলে। While loop/for loop বা if statement-সহ বিভিন্ন স্টেটমেন্টে Indentation-এর প্রয়োজন হয়। প্রয়োজন অনুযায়ী চারটি অথবা আটটি Indentation ব্যবহার করতে হয়। পাইথনে Indentation-কে Whitespace-ও বলা হয়।

আসলে পাইথনে আলাদা করে Indentation দিতে হয় না। কারণ পাইথন নিজেই Indentation তৈরি করে ফেলে। তাই প্রোগ্রাম লেখার সময় Indentation-এর কথা ভুলে গেলেও সমস্যা নেই। কখনো প্রোগ্রাম লেখার সময় যদি আলাদা করে স্পেস চলে আসে তখন বুঝতে হবে এখানে Indentation হবে বলেই স্পেস চলে আসছে। একই ব্লকের continuous line ভিন্ন ভিন্ন মার্জিনে শুরু হলে প্রোগ্রামিং-এ error দেখা দিবে। যেমন-

```
if True:

    print ("Answer")

    print ("True")

else:

    print ("Answer")

    print ("False")
```

উপরোক্ত কোডটি সঠিক নয়। কারণ এখানে একই ব্লকের পর পর দুটি স্টেটমেন্ট print "Answer" এবং print "False" এর মার্জিন ভিন্ন ভিন্ন।

উদাহরণ (সঠিক মার্জিন):

```
if True:

    print ("Answer")

    print ("True")

else:

    print ("Answer")

    print ("False")
```

মাল্টি-লাইন স্টেটমেন্ট (Multi line statement): যদি কোনো স্টেটমেন্ট বুঝানোর জন্য একাধিক লাইনের কোড ব্লক তৈরি করার প্রয়োজন হয় তখন তাকে মাল্টি-লাইন স্টেটমেন্ট বলে।

Python-এর Statement সাধারণত নতুন লাইনে গিয়ে শেষ হয়। তবে কোনো স্টেটমেন্টের শেষে continuation character (/) দিয়ে স্টেটমেন্টটি আরও চলবে (continued statement) বুঝানো হয়। যেমন-

```
total = item_one + \
        item_two + \
        item_three
```

কিন্তু, কোনো স্টেটমেন্ট যদি [], {}, বা () এর ভিতর থাকে তবে continuation character (\) দেয়ার দরকার নেই। যেমন-

```
days = ['Monday', 'Tuesday', 'Wednesday',
        'Thursday', 'Friday']
```

একাধিক স্টেটমেন্ট গ্রুপ (Suites): আলাদা আলাদা স্টেটমেন্ট এর একটিমাত্র 'গ্রুপ' যখন একসাথে একটি কোড ব্লক তৈরি করে তখন তাকে Suites বলে। জটিল/যৌগিক/বড় স্টেটমেন্ট-এর ক্ষেত্রে if, while, def অথবা class-এর ক্ষেত্রে একটি হেডার লাইন এবং পরে একটি suite-এর প্রয়োজন হয়। হেডার লাইনের আগে একটি কী-ওয়ার্ড থাকে এবং পরে কোলন চিহ্ন (:) থাকে, যার পরে এক বা একাধিক লাইনের suite ব্যবহার করা হয়। যেমন-

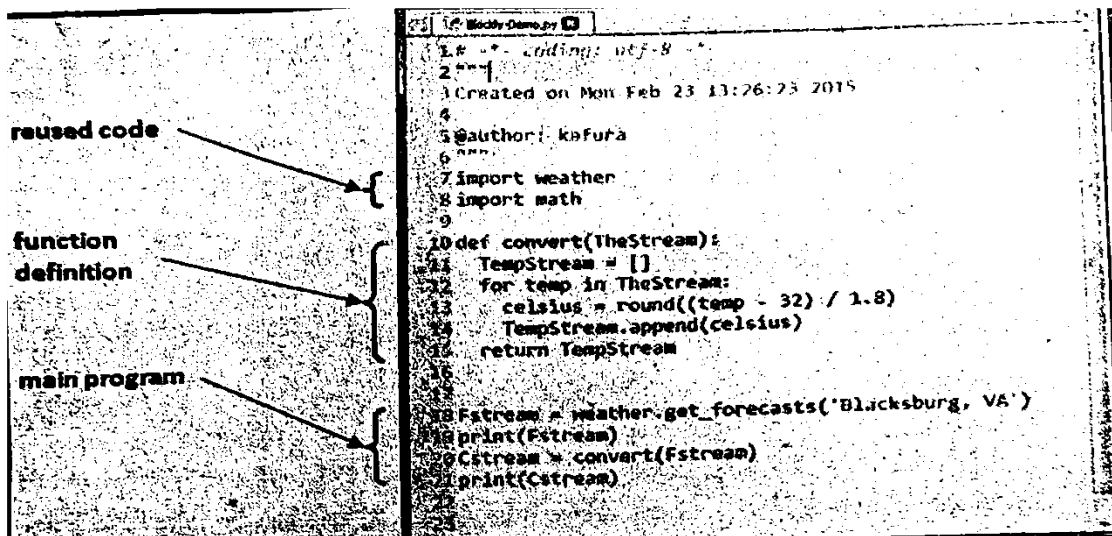
```
If expression:
    suite
elif expression:
    suite
else:
    suite
```


একই লাইনে একাধিক স্টেটমেন্ট (Multiple statements on a line): পাইথনে অনেক সময় একই লাইনে একাধিক স্টেটমেন্টও ব্যবহার করা হয়। এজন্য সেমিকোলন চিহ্ন ব্যবহার করা হয়। যেমন-

```
>>> a=2; b=3; c=a+b; print(c)
```

২. উদাহরণসহ পাইথনের প্রোগ্রাম স্ট্রাকচার ব্যাখ্যা কর।

উত্তর: প্রোগ্রাম স্ট্রাকচার (Program structure):



চিত্র: পাইথন প্রোগ্রাম স্ট্রাকচার

উপরোক্ত চিত্রটি লক্ষ্য করলে দেখা যায় যে, পাইথন প্রোগ্রাম স্ট্রাকচারের উল্লেখযোগ্য অংশসমূহ হচ্ছে-

- > ইমপোর্ট স্টেটমেন্টস (Import statements)
- > ফাংশন ও ক্লাস ডেফিনিশন (Function & Class definition)
- > মডিউল স্কোপ ভেরিয়েবলস (Module scope variables) এবং
- > মেইন প্রোগ্রাম (Main program) ।

ইমপোর্ট স্টেটমেন্টস (Import statements): কোনো লাইব্রেরি ক্লাসকে প্রোগ্রামে ব্যবহার করতে চাইলে সে ক্লাসটি বা ক্লাসগুলো যে প্যাকেজে আছে তা ইমপোর্ট স্টেটমেন্টস (Import statements)-এর মাধ্যমে ডিক্লেয়ার করতে হয়।

যেমন- import random, Import math ইত্যাদি।

ফাংশন ও ক্লাস ডেফিনিশন (Function & Class definition): এই অংশে প্রোগ্রামে ব্যবহৃত যাবতীয় ফাংশন ও ক্লাসসমূহকে ডিফাইন বা বর্ণনা করা হয়। যেমন-

```
def recur_fibo(n):
    If n 1:
    else:
    return in
    return(recur_fibo(n-1) + recur_fibo(n-2))
```

মডিউল স্কোপ ভেরিয়েবলস (Module scope variables): পাইথনের স্ট্যান্ডার্ড লাইব্রেরিতে বিভিন্ন ধরনের মডিউল আছে এবং প্রতিটি মডিউলের মধ্যে একাধিক মেথড রয়েছে। এই অংশে আমরা প্রয়োজনে কোনো মডিউল বা তার মেথডকে কল করতে পারি। যেমন-

```
Import math
print (math.cell(7.7))
```

মেইন প্রোগ্রাম (Main program): প্রোগ্রামের এই অংশে মূল কোডিং লেখা হয়। প্রোগ্রামে ব্যবহৃত অন্যান্য মেথড ও অবজেক্টসমূহ এই সেকশনেই কল করতে হয়।

অধ্যায়
০৩

ভেরিয়েবল ও ডাটা টাইপ

১৩ অধ্যায়ের প্রধান সূচিপত্র (contents)

বিষয়	পৃষ্ঠা নং
❖ সূত্রাবলী ও শর্টকাট টেকনিক	
❖ অতি সংক্ষিপ্ত প্রশ্নাবলি	
❖ সংক্ষিপ্ত প্রশ্নাবলি	
❖ রচনামূলক প্রশ্নাবলি	

বোর্ড বোর্ড প্রশ্নাবলির বিশ্লেষণ (Board Questions Analysis) প্রশ্নাবলির বিশ্লেষণ (Board Questions Analysis)

এই অধ্যায় থেকে বিগত পাঁচ বছরের ডিপ্লোমা ইন ইঞ্জিনিয়ারিং, ডিপ্লোমা ইন টেক্সটাইল, ডিপ্লোমা ইন এগ্রিকালচার, ডিপ্লোমা ইন মেরিন পরীক্ষায় মোট প্রশ্ন। নিচে ‘Board Questions Analysis’ অংশে এই অধ্যায় থেকে কোন সালে কোন বোর্ডে কতটি প্রশ্ন হয়েছে তা দেখানো হলো।

সাল \ বোর্ড	অতি সংক্ষিপ্ত	সংক্ষিপ্ত প্রশ্ন	রচনামূলক প্রশ্ন
২০২৪	২টি	১টি	—
২০২৩	২টি	২টি	—
২০২২	১টি	১টি	—
২০২১	—	—	—



“প্রত্যেকেরই প্রোগ্রাম শেখা উচিত, কারণ এটা তোমাকে চিন্তা করতে শেখায়।”

- স্টিভ জবস (Steve Jobs)

অতি সংক্ষিপ্ত প্রশ্নোত্তর :

১. স্ট্রিং কী?

[বাকশির্বো: '২০, '২০R, '২১]

উত্তর: সিঙ্গেল কোট বা ডাবল কোটের ভিতরে যাই থাকে, তাই স্ট্রিং।

২. $x = 50$ এবং $y = 3$ হলে x/y এবং $x\%y$ এর মান কত?

[বাকশির্বো: '২০R]

উত্তর: $x / y = 16$ $x\% * y = 2$

৩. Datatype conversion কী?

[বাকশির্বো: '২০R]

অথবা, টাইপ কাস্টিং বলতে কী বোঝায়?

উত্তর: এক টাইপ ভেরিয়েবলকে অন্য টাইপ ভেরিয়েবলে কনভার্ট করার প্রক্রিয়াকে ডাটা টাইপ কনভার্সন বা টাইপ কাস্টিং বলে।

৪. $\text{int } x = 3$ $y = 2$ হলে x/y ; $x\%y$ এবং $x*y$ -এর মান কত?

[বাকশির্বো: '১৯]

উত্তর: 1, 1 এবং 6.

৫. Local ও Global variable কী?

[বাকশির্বো: '১৯]

উত্তর: লোকাল ভেরিয়েবল (Local of Variable): যেসব ভেরিয়েবলের কার্যকারিতা শুধুমাত্র কোন নির্দিষ্ট অংশ বা ফাংশনের মধ্যে সীমাবদ্ধ থাকে তাকে লোকাল ভেরিয়েবল বলে। যখন কোনো ভেরিয়েবল মান ও অস্তিত্ব শুধুমাত্র সংশ্লিষ্ট ফাংশনের মধ্যে সীমাবদ্ধ থাকে। এই মান অন্য ফাংশনে সরাসরি ব্যবহার করা যায় না। যতক্ষণ একটি ফাংশন নিয়ে কাজ করা হয় ততক্ষণ পর্যন্ত ঐ ফাংশনের লোকাল ভেরিয়েবলগুলো সক্রিয় থাকে। কোনো ফাংশনের কার্যক্রম শেষে স্বয়ংক্রিয়ভাবে লোকাল ভেরিয়েবলগুলোর জন্য বরাদ্দকৃত মেমোরি পরিসর খালি হয়ে যায়। ফলে দুই বা ততোধিক ফাংশনে একই নাম এবং ডাটা টাইপের লোকাল ভেরিয়েবল ব্যবহার করা যেতে পারে এবং তাতে কোনো সমস্যা হয় না।

গ্লোবাল ভেরিয়েবল (Global Variable): যখন কোন ভেরিয়েবলের কার্যকারিতা কোন নির্দিষ্ট ফাংশনের মধ্যে সীমাবদ্ধ না থেকে এর মান সকল ফাংশনেই ব্যবহার করা যায় তাকে গ্লোবাল ভেরিয়েবল বলে। গ্লোবাল ভেরিয়েবলের মান ও অস্তিত্ব কোনো নির্দিষ্ট ব্লক বা কোনো নির্দিষ্ট ফাংশনের মধ্যে সীমাবদ্ধ না থেকে পুরো প্রোগ্রামে বিস্তৃত থাকে। এ ধরনের ভেরিয়েবল ফাংশনের মধ্যে নয়; ফাংশনের উপর ঘোষণা করা হয়।

৬. ভেরিয়েবল কী?

[বাকশির্বো: '১৮']

উত্তর: ভেরিয়েবল হচ্ছে কম্পিউটার মেমোরির সেই নির্ধারিত জায়গা (স্পেস), যেখানে বিভিন্ন মান (value) জমা করে রাখা যায়। ভেরিয়েবল তৈরি করা মানেই কম্পিউটারের মেমোরিতে একটি নির্দিষ্ট স্পেসে ডাটা সঞ্চয় করে রাখা। কোনো ভেরিয়েবল কী ধরনের ডাটা রেকর্ড বা স্টোর করবে সেটা ঐ ভেরিয়েবলের উপর নির্ভর করে।

৭. ডাটা বলতে কী বোঝায়?

উত্তর: প্রোগ্রামে ব্যবহৃত যে-কোনো মানই (value) ডাটা। প্রোগ্রামিং-এ বিভিন্ন প্রকার ডাটা নিয়ে কাজ করা হয়। প্রোগ্রাম চালনার সময় সব ডাটাই মেমোরিতে সংরক্ষিত হয় এবং বিভিন্ন টাইপের ডাটাই মেমোরিতে ভিন্ন ভিন্ন বাইটের জায়গা দখল করে।

৮. টোকেন কী?

[বাকশির্বো: '২৩']

উত্তর: 'Token' ইংরেজি শব্দটির বাংলা অর্থ হচ্ছে প্রতীক বা চিহ্ন। যে-কোনো প্রোগ্রাম কতকগুলো Statement নিয়ে গঠিত। আবার প্রতিটি Statement কতকগুলো Word এবং Character এর সমষ্টি। পাইথন প্রোগ্রামে ব্যবহৃত এরূপ Word বা Character ও Symbol-সমূহকে একত্রে Token বলে।

৯. ভেরিয়েবলের মান পুনঃনির্ধারণ বলতে কী বোঝায়?

উত্তর: পাইথনে কোনো ভেরিয়েবলের মধ্যে একাধিবার নতুন নতুন ভ্যালু স্টোর করা যায়। এভাবে কোনো ভেরিয়েবলের মান বার বার নির্ধারণ করার এ প্রক্রিয়াকে রিঅ্যাসাইনমেন্ট অব ভেরিয়েবল

বা ভেরিয়েবলের মান পুনঃনির্ধারণ বলে। এক্ষেত্রে সর্বশেষ স্টোরকৃত মানটিই ভেরিয়েবলের মান হিসেবে জমা থাকে। যেমন-

```
>>> x = 110
>>> X 110
>>> x = 154.32
>>> x 154.32
>>> x 'Tamim Al Iqbal'
>>> X
Tamim Al Iqbal
```

১০. মাল্টিপল ভেরিয়েবল অ্যাসাইনমেন্ট বলতে কী বোঝায়?

উত্তর: একই লাইনে একাধিক ভেরিয়েবল ডিক্লারেশনের পদ্ধতিকে মাল্টিপল ভেরিয়েবল অ্যাসাইনমেন্ট বলে। যেমন-

```
a = b = c = 10
এবং
a, b, c = 10,20, "Tamim Al Iqbal"
```

১১. পাইথনে ডাটা টাইপ কত প্রকার ও কী কী?

অথবা, পাইথনে ডাটা টাইপের প্রকারভেদ উল্লেখ কর।

উত্তর: ডাটার মান, ধরন এবং মেমোরি স্পেস সংরক্ষণের ভিত্তিতে পাইথনে পাঁচ ধরনের স্ট্যান্ডার্ড ডাটা টাইপ রয়েছে, যেমন-

- Numbers
- String

- List
- Tuple
- Dictionary

১২. অপারেটরের পেসিডেন্স বলতে কি বুঝায়?

[বাকাশিবো: '২২, '২৩]

উত্তর: কোনো এক্সপ্রেশনে একাধিক অপারেটর থাকলে, কোন অপারেটরটি আগে কার্যকর হবে তার অগ্রাধিকারের ক্রম (order of priority)।

সংক্ষিপ্ত প্রশ্নোত্তর :

১. ভেরিয়েবল ঘোষণা করার নিয়ম লেখ।

[বাকাশিবো: '২১]

অথবা, Variable লেখার নিয়ম লেখ।

[বাকাশিবো: '১৯]

অথবা, পাইথনে ভেরিয়েবল নামকরণের নিয়মাবলি উল্লেখ কর।

উত্তর: নিচে ভেরিয়েবলের নাম নির্ধারণের নিয়মাবলি দেওয়া হলো-

পাইথনে ভ্যারিয়েবল লেখার সময় কিছু নিয়ম মেনে ভ্যারিয়েবল ডিফাইন করা হয়।

- ভ্যারিয়েবলের নাম অবশ্যই এক শব্দের হতে হবে। অর্থাৎ এরকমভাবে ভ্যারিয়েবল লেখা যাবে না:

my variable 10

- প্রথম অক্ষর অবশ্যই একটি alphabetic letter (uppercase or lowercase) অথবা underscore (_) হতে হবে। যেমন nafis, a, b, variable লেখা যাবে ভ্যারিয়েবল হিসেবে কিন্তু 1name, @nafis, 7a, %b এইভাবে লেখা যাবে না। প্রথম অক্ষর ছাড়া পরে letter, underscore, number ব্যবহার করা যাবে, কিন্তু পাইথনের কনভেনশন হচ্ছে ভ্যারিয়েবলের নাম সবসময় lowercase letter দিয়ে শুরু করা।

- পাইথন Case Sensitive অর্থাৎ $a = 4$ এবং $A = 4$ একই ভ্যারিয়েবল নয়।
- পাইথনের কিছু reserved কী-ওয়ার্ড আছে এগুলো ব্যবহার করা যাবে না।

যেমন: if, else, elif, for, while, break, continue, except, as, in, is, True, False, yield, None, def, del, class ইত্যাদি।

- ভেরিয়েবলের মান হিসেবে অর্থবোধক নাম ব্যবহার করাই যুক্তযুক্ত। অনর্থক নাম পরিহার করাই শ্রেয়। যেমন: আয়তক্ষেত্রের ক্ষেত্রফল নির্ণয় করার জন্য Length, Width, Rectangle_Area ব্যবহার করাই শ্রেয়; A, B, R নয়।

উদাহরণ,

```
>>> this is_a_normal_name = 7
```

```
>> 123abc =
```

```
SyntaxError: invalid syntax
```

```
>>> space are not allowed
```

```
SyntaxError: invalid syntax
```

২. বিভিন্ন প্রকার টোকেন ও তাদের ব্যবহার লেখ।

[বাকাশিবো: '১৯]

উত্তর: পাইথনে ব্যবহৃত বিভিন্ন প্রকার টোকেন ও তাদের ব্যবহার নিম্নরূপ:

টোকেন	ব্যবহার
কীওয়ার্ড (Keyword)	প্রোগ্রাম কোড লেখার জন্য।
আইডেন্টিফায়ার (Identifier)	ভেরিয়েবল, ফাংশন, স্ট্রাকচার, ক্লাস ইত্যাদি নামকরণের জন্য।
স্ট্রিং (String)	প্রোগ্রামে একগুচ্ছ ক্যারেক্টার নিয়ে কাজ করার জন্য।

পাঙ্কচুয়েটর (Punctuator)	কীওয়ার্ড, আইডেন্টিফায়ার, অপারেটর, অপারেন্ডের মধ্যে পার্থক্য করার জন্য।
স্পেশাল সিম্বল (Special Symbol)	কতগুলো বিশেষ কাজে ব্যবহার করার জন্য।
অপারেটর, অপারেন্ড এবং এক্সপ্রেশন	বিভিন্ন রকম গাণিতিক ও যৌক্তিক অপারেশন সম্পন্ন করার জন্য।

৩. প্রোগ্রামে ভেরিয়েবলের প্রয়োজনীয়তা লেখ।

[বাকশিবো: '২৩]

উত্তর:

- প্রোগ্রামে ভেরিয়েবল হলো ডেটা বা মান সংরক্ষণের জন্য ছোট বক্সের মত। এতে সংখ্যা, লেখা, সত্য-মিথ্যা ইত্যাদি রাখা যায়।

- ভেরিয়েবল হলো তথ্য ধারণ, পরিবর্তন, এবং ব্যবস্থাপনার জন্য প্রোগ্রামের অপরিহার্য অংশ। একবার ভেরিয়েবল ব্যবহার করলে এর মান পরিবর্তন করা যায়। যেমন, বয়স বারবার বাড়তে পারে, তাই এক্সপ্রেশনে পরিবর্তন সম্ভব।
- ভেরিয়েবল নাম দেখে বুঝতে পারা যায় সেটার মান কী, তাই কোড পড়া ও লেখা সহজ হয়।
- ভেরিয়েবল ব্যবহার করে সহজে যোগ, বিয়োগ, গুণ, ভাগ ইত্যাদি করা যায়। ব্যবহারকারীর কাছ থেকে তথ্য নিতে এবং ফলাফল দেখাতে ভেরিয়েবল প্রয়োজন।
- ভেরিয়েবল ছাড়া প্রোগ্রাম ডেটা প্রসেস করতে পারে না, তাই এটি প্রোগ্রামের মূল অংশ।

৪. লোকাল ও গ্লোবাল ভেরিয়েবলের পাথক্য লিখ।

[বাকশিবো: '২৩]

উত্তর:

i. স্কোপ (Scope): * **লোকাল ভেরিয়েবল:** এর স্কোপ সীমিত। এটি কেবল তার নিজস্ব ফাংশন বা ব্লকের মধ্যে অ্যাক্সেসযোগ্য। ফাংশনের বাইরে এই ভেরিয়েবল ব্যবহার করার চেষ্টা করলে এরর (Error) দেখাবে। * **গ্লোবাল ভেরিয়েবল:** এর স্কোপ গ্লোবাল বা বৈশ্বিক। এটি প্রোগ্রামের যেকোনো ফাংশন থেকে অ্যাক্সেস এবং পরিবর্তন করা যেতে পারে।

ii. **জীবনকাল (Lifetime):** * **লোকাল ভেরিয়েবল:** এটি যখন ফাংশন কল হয় বা ব্লক শুরু হয়, তখন অস্তিত্ব লাভ করে এবং ফাংশন বা ব্লকটি শেষ হলেই এর অস্তিত্ব শেষ হয়ে যায় (মেমরি থেকে মুছে যায়)। * **গ্লোবাল ভেরিয়েবল:** প্রোগ্রাম এক্সিকিউশন (execution) শুরু হওয়ার সময় এটি তৈরি হয় এবং প্রোগ্রামটি সম্পূর্ণভাবে শেষ না হওয়া পর্যন্ত এটি মেমরিতে বিদ্যমান থাকে।

iii. **নিরাপত্তা ও রক্ষণাবেক্ষণ (Security & Maintenance):** * **লোকাল ভেরিয়েবল:** কোডের নিরাপত্তা বেশি, কারণ কোনো ফাংশন ভুল করে অন্য ফাংশনের ভেরিয়েবল পরিবর্তন করতে পারে না। এটি কোড রক্ষণাবেক্ষণ এবং ডিবাগিং (Debugging) সহজ করে। * **গ্লোবাল ভেরিয়েবল:** এটি যেকোনো জায়গা থেকে পরিবর্তিত হতে পারে, যা অপ্রত্যাশিত ফলাফল তৈরি করতে পারে এবং কোড ডিবাগিং জটিল করে তোলে। তাই এদের ব্যবহার সতর্কতার সাথে করা উচিত

৫. Quotation ও Comment এর ব্যবহার লিখ।

[বাকশিবো: '২৪]

উত্তর:

i. Quotation-এর ব্যবহার (উদ্ধৃতি চিহ্ন):

Quotation বা উদ্ধৃতি চিহ্ন (যেমন: একক উদ্ধৃতি চিহ্ন ' অথবা দ্বৈত উদ্ধৃতি চিহ্ন ") ব্যবহৃত হয় কোনো স্ট্রিং লিটারেল বা টেক্সট ডেটা চিহ্নিত করার জন্য।

মূল কাজ:

কোড-এর মধ্যে যখন কোনো টেক্সট বা শব্দগুচ্ছকে সাধারণ প্রোগ্রামিং নির্দেশনা (Instruction) হিসেবে নয়, বরং মান (Value) হিসেবে দেখাতে হয়, তখন তাকে উদ্ধৃতি চিহ্নের মধ্যে রাখতে হয়।

ii. Comment-এর ব্যবহার (মন্তব্য);

Comment বা মন্তব্য হলো কোডের সেই অংশ যা কম্পাইলার বা ইন্টারপ্রেটার দ্বারা বাদ দেওয়া হয় এবং কার্যকর (Execute) করা হয় না। এটি শুধুমাত্র মানুষের পড়ার জন্য ব্যবহৃত হয়।

মূল কাজ:

কমেন্ট ব্যবহার করা হয় কোডকে ব্যাখ্যা করা, নোট রাখা এবং ভবিষ্যতে সহজে বোঝার জন্য।

৬. $x = 15$ ও $Y = 4\frac{x}{y}$ ও $\frac{x^0}{y}$ এর মান কত?

[বাকশির্বো: '২৪]

উত্তর:

প্রোগ্রাম:

$X = 15$

$Y = 4$

$\text{division_result} = X / Y$

$\text{modulo_result} = X \% Y$

$\text{print}(\text{division_result})$

$\text{print}(\text{modulo_result})$

প্রোগ্রামটির আউটপুট:

3.75

3

রচনামূলক প্রশ্নোত্তর :

১. উদাহরণসহ ভেরিয়েবলের মান নির্ধারণ ও মান পুনঃনির্ধারণ পদ্ধতি বর্ণনা কর।

উত্তর: পাইথন Statically Typed ল্যাংগুয়েজ নয়। স্ট্যাটিক্যালি টাইপড ল্যাংগুয়েজগুলোতে ভেরিয়েবল তৈরির সময় কোন টাইপের ডাটা রাখব ঐ ভেরিয়েবলে, তা বলে দিতে হয়। পাইথনে সেভাবে বলে দিতে হয় না। শুধু ভেরিয়েবলের একটা নাম দিয়ে যে-কোনো ডাটা অ্যাসাইন করা যায়। পাইথনে কোনো ভেরিয়েবলে ডাটা অ্যাসাইন করলে তা অটোমেটিক ডাটা টাইপ সেট করে নেয়।

পাইথনে সমতা চিহ্নের (=) মাধ্যমে কোনো ধরনের মান (value) ভেরিয়েবল হিসেবে সেট করা যায়, এক্ষেত্রে আলাদা করে কোনো ডিক্লারেশনের প্রয়োজন নেই। সমতা চিহ্নের বাম পাশে ব্যবহৃত শব্দটি ভেরিয়েবলের নাম এবং ডান পাশের সংখ্যাটি ভেরিয়েবলের মান নির্দেশ

করে। যেমন-

```
>>> num = 10

>>> print(num)

10

>>> print(num*2)

20

>>> print(num+5)

15
```

উপরোক্ত উদাহরণে প্রথমে ভেরিয়েবল num-এর মান ১০ জমা রাখা হয়েছে। এরপরের লাইনে print() ফাংশনের আর্গুমেন্ট হিসেবে সেই a-কেই পাঠানো হয়েছে। প্রিন্ট ফাংশনের আর্গুমেন্ট হিসেবে কিছু পাঠালে তাই প্রিন্ট হয়। সেজন্য a-এর মান হিসেবে স্ক্রিনে ১০ প্রিন্ট হয়েছে। পরবর্তী লাইনে num-এর মানকে ২ দিয়ে গুণ করলে ২০ পাওয়া যায়। তাই আউটপুটে ২০ দেখাচ্ছে। এমনভাবে পরবর্তী লাইনের আউটপুট ১৫ দেখানো হয়েছে।

ভেরিয়েবলের মান পুনঃনির্ধারণ (Re-assignment of values to variables): পাইথনে কোনো ভেরিয়েবলের মধ্যে একাধিকবার নতুন নতুন ভ্যালু স্টোর করা যায়। এভাবে কোনো ভেরিয়েবলের মান বার বার নির্ধারণ করার এই প্রক্রিয়াকে রি-অ্যাসাইনমেন্ট অফ ভেরিয়েবল বা ভেরিয়েবলের মান পুনঃনির্ধারণ বলে। তবে এক্ষেত্রে সর্বশেষ স্টোরকৃত মানটিই ভেরিয়েবলের মান হিসেবে জমা থাকে। যেমন-

```
>>> x=110

>>> X

110

>>> x=154.32

>>> X

154.32

>>> x = "Mahi"

>>> X

Mahi
```

উপরোক্ত উদাহরণে x-এর মান হিসেবে একটি পূর্ণসংখ্যা ১১০ প্রিন্ট হবে। পরবর্তী দুটো কমান্ডে এর X মান পরিবর্তিত হয়ে যথাক্রমে দশমিক সংখ্যা ১৫৪.৩২ এবং স্ট্রিং Mahi-কে আউটপুট হিসেবে প্রিন্ট করার জন্য ব্যবহার করা হয়েছে। পাইথনে ভেরিয়েবলে নির্দিষ্ট কোনো ডাটা টাইপ নেই। তাই একই ভ্যারিয়েবল প্রথমে একটি নাম্বার এবং পরবর্তীতে সেটাতে একটি স্ট্রিং জমা রাখা গেছে।

উল্লেখ্য, আমরা প্রথম যখন কোনো একটা ভেরিয়েবলে কোনো ভ্যালু অ্যাসাইন করি তখন সেই ভেরিয়েবলটা initialize হয়। পরবর্তীতে আমরা ঐ ভেরিয়েবলটাতেই আবার বিভিন্ন ভ্যালু রেখে

কাজ করতে পারি। কিন্তু আমরা যদি ভ্যালু অ্যাসাইন না করে ঐ ভেরিয়েবল নিয়ে কাজ করার চেষ্টা করি তাহলে প্রোগ্রামে এরর দেখাবে। যেমন-

```
>>> variable
```

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

NameError: name 'variable' is not defined

একাধিক ভেরিয়েবলের মান নির্ধারণ (Multiple variable assignment): পাইথন একই লাইনে একাধিক ভেরিয়েবল ডিক্লারেশনের সুবিধা প্রদান করে। এতে একাধিক ভেরিয়েবলের জন্য একটি নির্দিষ্ট মান স্টোর করা যায়। আবার একাধিক ভেরিয়েবলের জন্য যথাক্রমে একাধিক মানও একসাথে স্টোর করা যায়। পাইথনে ভেরিয়েবল ডিক্লারেশনের এ পদ্ধতিকে মাল্টিপল ভেরিয়েবল অ্যাসাইনমেন্ট বলে। যেমন-

```
a = b = c = 10
```

এবং

```
a, b, c = 10, 20, "Mahi"
```

প্রথমোক্ত উদাহরণে a, b এবং c প্রতিটি ভেরিয়েবলের মান ১০ কিন্তু দ্বিতীয় উদাহরণে a, b এবং c-এর মান যথাক্রমে ১০, ২০ এবং Mahi নির্ধারণ করা হয়েছে।

উদাহরণ:

```
>>> x, y = 2, 3
```

```
>>> X
```

```
2
```

```
>>> y
```

২. উদাহরণসহ বিভিন্ন প্রকার ডাটা টাইপ বর্ণনা কর।

উত্তর: প্রোগ্রামিং মানেই হলো ডাটা প্রসেসিং। ডাটা ছাড়া প্রোগ্রামিং অসম্ভব। কম্পিউটারে কোনো প্রোগ্রাম লিখতে হলে কম্পিউটারকে অবশ্যই ঐ নির্দিষ্ট প্রোগ্রামের জন্য ডাটা ইনপুট দিতে হবে। প্রোগ্রামে ব্যবহৃত যে-কোনো মানই ডাটা। পাইথনে ইন্টিজার, ফ্লোটিং পয়েন্ট, ক্যারেক্টার, স্ট্রিং, লজিক্যাল ইত্যাদি প্রায় সব ধরনের ডাটা নিয়েই কাজ করা যায়। তবে ডাটার মান, ধরন এবং মেমরি স্পেস সংরক্ষণের ভিত্তিতে পাইথনে পাঁচ ধরনের স্ট্যান্ডার্ড ডাটা টাইপ রয়েছে। যেমন-

- (i) Numbers
- (ii) String
- (iii) List
- (iv) Tuple
- (v) Dictionary

অধ্যায়
০৪

পাইথন অপারেটর

২৪ অধ্যায়ের প্রধান সূচিপত্র (contents)

বিষয়	পৃষ্ঠা নং
❖ সূত্রাবলী ও শর্টকাট টেকনিক	
❖ অতি সংক্ষিপ্ত প্রশ্নাবলি	
❖ সংক্ষিপ্ত প্রশ্নাবলি	
❖ রচনামূলক প্রশ্নাবলি	

বোর্ড বোর্ড প্রশ্নাবলির বিশ্লেষণ (Board Questions Analysis) প্রশ্নাবলির বিশ্লেষণ (Board Questions Analysis)

এই অধ্যায় থেকে বিগত পাঁচ বছরের ডিপ্লোমা ইন ইঞ্জিনিয়ারিং, ডিপ্লোমা ইন টেক্সটাইল, ডিপ্লোমা ইন এগ্রিকালচার, ডিপ্লোমা ইন মেরিন পরীক্ষায় মোট প্রশ্ন। নিচে ‘Board Questions Analysis’ অংশে এই অধ্যায় থেকে কোন সালে কোন বোর্ডে কতটি প্রশ্ন হয়েছে তা দেখানো হলো।

সাল \ বোর্ড	অতি সংক্ষিপ্ত	সংক্ষিপ্ত প্রশ্ন	রচনামূলক প্রশ্ন
২০২৪	—	৩টি	১টি
২০২৩	২টি	৩টি	১টি
২০২২	১টি	২টি	—
২০২১	—	—	—



“অপর্যাপ্ত তথ্য দিয়ে ভুল করাও ভালো, কোনো তথ্য ছাড়াই সিদ্ধান্ত নেওয়ার চেয়ে।”

- চার্লস ব্যাবেজ (Charles Babbage)

অতি সংক্ষিপ্ত প্রশ্নোত্তর :

১. Assignment Operator-এর কাজ লেখ।

[বাকশির্বো: '২১]

অথবা, অ্যাসাইনমেন্ট অপারেটর কেন ব্যবহার করা হয়?

উত্তর: কোনো Expression বা Variable এর মানকে অপর কোনো Variable এর মান হিসেবে নির্ধারণের জন্য অ্যাসাইনমেন্ট অপারেটর ব্যবহার করা হয়।

২. Operator বলতে কী বুঝায়?

[বাকশির্বো: '২০]

অথবা, উদাহরণসহ অপারেটর ও অপারেণ্ড এর সংজ্ঞা লেখ।

অথবা, অপারেটর ও অপারেণ্ড বলতে কী বুঝায়?

অথবা, Operator ও Operand বলতে কী বুঝায়?

উত্তর: অপারেটর (Operator): যে সকল বিশেষ চিহ্ন বা Symbol ব্যবহার করে Data, Variable বা Expression এর উপর গাণিতিক, তুলনামূলক বা যৌক্তিক কার্য সম্পন্ন করা হয়, তাদেরকে Operator বলে।

অপারেণ্ড (Operand): Operator যে-সব Data, Variable বা Expression এর উপর কাজ করে, তাদেরকে Operand বলে।

উদাহরণ:

$x + y$ এ x এবং y হলো অপারেণ্ড এবং '+' হলো অপারেটর।

৩. রিলেশনাল অপারেটরগুলোর নাম লেখ।

[বাকশির্বো: '১৯]

অথবা, পাইথন Programming-এ ব্যবহৃত Relational operator-সমূহ কী কী?

অথবা, প্রতীকের সাহায্যে রিলেশনাল অপারেটরগুলো উল্লেখ কর।

অথবা, তুলনামূলক অপারেটরগুলো বর্ণনা কর।

উত্তর: পাইথনে মোট ৬টি Relational Operator আছে। যথা:

Operator (অপারেটর)	Symbol (চিহ্ন)	Example (উদাহরণ)	Operation (কাজ)
Equal	==	$a == b$	a এর মান b এর সমান।
Greater than	>	$a > b$	a এর মান b এর চেয়ে বড়।
Less than	<	$a < b$	a এর মান b এর চেয়ে ছোট।
Greater than or equal to	>=	$a >= b$	a এর মান b এর চেয়ে বড় / অথবা সমান।
Less than or equal to	<=	$a <= b$	a এর মান b এর চেয়ে ছোট অথবা সমান।
Not equal	!=	$a \neq b$	a এর মান b এর সমান নয়।

৪. অপারেটর প্রেসিডেন্স বলতে কী বুঝায়?

[বাকশিবি: '১৮, '১৯, '২২, '২৩]

অথবা, Operator Precedence বলতে কী বুঝায়?

অথবা, Precedence বলতে কী বুঝায়?

উত্তর: কোনো এক্সপ্রেশনে একাধিক অপারেটর ব্যবহৃত হলে কম্পাইলার যে অনুক্রমের (Order) ভিত্তিতে অপারেটরের অগ্রগণ্যতা (কোনটার কাজ আগে, কোনটার কাজ পরে হবে) নির্ধারণ করে, তাকে অপারেটরের প্রেসিডেন্স (Precedence) বলে।

৫. $22\% 7, 22\% 7$ এবং $22\% 5$ এর মান কত এবং কেন?

উত্তর: $22\% 7 = 1$

$$-22\% 7 = 6$$

$$-22\% 5 = 3$$

এদের প্রত্যেকের মান আলাদা আলাদা। ২য় ও ৩য় রাশিটির 1st operand এর চিহ্নটি যেহেতু Minus (-) এর, তাই result এর বেলায় 1st operand এর সঙ্গে আর যত যোগ করলে নিঃশেষে বিভাজ্য হবে, তাই হবে এর মান।

৬. $17\% 3 =$ কত?

উত্তর: $17\% 3 = 2$

৭. এই সমীকরণটি পাইথন এক্সপ্রেশনে রূপান্তর কর। $x = \frac{a(b/c)}{2a^2}$

উত্তর: $x = (a * (b/c)) / (2 * a * a)$

সংক্ষিপ্ত প্রশ্নোত্তর :

১. পাইথন প্রোগ্রামে বিভিন্ন প্রকার কম্পারিজম অপারেটরগুলোর সিঙ্গেলসহ নাম লেখ।

[বাকশির্ষো: '১৭, '১৯, '২০R, '২২]

অথবা, রিলেশনাল বা তুলনামূলক অপারেটরসমূহ বর্ণনা কর।

উত্তর: পাইথনে রিলেশনাল বা তুলনামূলক অপারেটর ছয় ধরনের। যথা-

Python Comparison Operators		
অপারেটর	বর্ণনা	উদাহরণ
<code>==</code> (ইকুয়াল)	যদি দুই পাশের অপারেন্ড দুইটি সমান হয় তবে শর্তটি সত্যি হয়।	<code>(a == b)</code>
<code>!=</code> (নট ইকুয়াল)	যদি দুই পাশের অপারেন্ড দুইটি সমান না হয় তবে শর্তটি সত্যি হয়।	<code>a! = b</code>
<code>></code> (গ্রেটার দ্যান)	যদি ডান পাশের অপারেন্ডের চেয়ে বাম পাশের অপারেন্ড বড় হয় তবে শর্তটি সত্যি হয়।	<code>(a > b)</code>
<code>>=</code> (গ্রেটার দ্যান অর ইকুয়াল)	যদি ডান পাশের অপারেন্ডের চেয়ে বাম পাশের অপারেন্ড বড় বা সমান সমান হয় তবে শর্তটি সত্যি হয় তবে শর্তটি সত্যি হয়।	<code>(a >= b)</code>
<code><</code> (লেস দ্যান)	যদি ডান পাশের অপারেন্ডের চেয়ে বাম পাশের অপারেন্ড ছোট হয় তবে শর্তটি সত্যি হয়।	<code>(a < b)</code>
<code><=</code> (লেস দ্যান অর ইকুয়াল)	যদি ডান পাশের অপারেন্ডের চেয়ে বাম পাশের অপারেন্ড ছোট বা সমান হয় তবে শর্তটি সত্যি হয় তবে শর্তটি সত্যি হয়।	<code>(a <= b)</code>

২. পাইথন অপারেটর কত প্রকার ও কী কী?

[বাকশির্বো: '২০, '২৩]

অথবা, Operator এর প্রকারভেদ লেখ।

[বাকশির্বো: '২০০৪]

উত্তর: পাইথন অপারেটর আট প্রকার। যথা:

- গাণিতিক অপারেটর (Arithmetic Operator)
- তুলনামূলক অপারেটর (Comparison Operator)
- যৌক্তিক অপারেটর (Logical Operator)
- অ্যাসাইনমেন্ট অপারেটর (Assignment Operator)

- বুলিয়ান অপারেটর (Boolean Operator)
- মেম্বারশিপ অপারেটর (Membership Operator)
- আইডেন্টিটি অপারেটর (Identity Operator)
- বিটওয়াইজ অপারেটর (Bitwise Operator)

৩. গাণিতিক অপারেটরসমূহের নাম লেখ।

[বাকশির্বো: '২০, '২৪]

অথবা, অ্যারিথমেটিক অপারেটরের বর্ণনা দাও।

[বাকশির্বো: '১৮]

অথবা, Mathematical operator এর বর্ণনা দাও।

অথবা, গাণিতিক অপারেটরের বর্ণনা দাও।

উত্তর: যে-সব অপারেটর বিভিন্ন ধরনের গাণিতিক কাজ (যেমন- যোগ, বিয়োগ, গুণ, ভাগ, ভাগশেষ ইত্যাদি) করার জন্য ব্যবহৃত হয়, তাদেরকে গাণিতিক অপারেটর (Arithmetic operator) বলে।

Operator-এর নাম	Operator	Operation-এর কাজ
Addition	+	দুটি Operand কে যোগ করে। যেমন: $a + b$
Subtraction	-	১ম Operand থেকে ২য় Operand বিয়োগ করে। যেমন: $a - b$
Multiplication	*	দুটি Operand কে গুণ করে। যেমন: $a * b$
Division	/	১ম Operand কে ২য় Operand দিয়ে ভাগ করে। যেমন: a / b
Modulus	%	১ম Operand কে ২য় Operand দিয়ে ভাগ করার পর যা ভাগশেষ থাকে। তা দেখায়। যেমন: $a \% b$

Exponent (পাওয়ার)	**	বাম পাশের অপারেন্ড হলো বেস আর ডান পাশের অপারেটর হলো পাওয়ার। বেস ও পাওয়ারকে ব্যবহার করে পাওয়ার মান নির্ণয় করে। যেমন: $a**b=32$
Floor Division (পূর্ণ সংখ্যার ভাগফল)	//	অপারেটরের বাম পাশের মান থেকে ডান পাশের মান দিয়ে ভাগ করে ভাগফল নির্ণয় করে এবং সাধারণত দশমিকের পরের সংখ্যা স্কিপ করে যায়। যেমন: $13//3=4$ এবং $13.0//3.0=4.0$

৪. লজিক্যাল অপারেটরের বর্ণনা দাও।

[বাকশিবো: '২৩]

অথবা, লজিক্যাল Operator-গুলোর ব্যবহার লেখ।

অথবা, লজিক্যাল অপারেটরসমূহের নাম লেখ।

উত্তর: পাইথনে লজিক্যাল অপারেটর তিন ধরনের। যথা-

Python Logical Operators		
অপারেটর	বর্ণনা	উদাহরণ
and (অ্যান্ড)	যদি দুটি অপারেন্ডই সত্যি হয়, 'কেবল তবেই শর্তটি সত্যি হয়।	(a and b)
or (অর)	দুটি অপারেন্ডের যে-কোনো একটি সত্যি হলেই শর্তটি সত্যি হয়।	(a or b)
not (নট)	না-বোধক বুঝায়, সত্য অপারেন্ডের আগে not দিলে তা মিথ্যা হয়ে যায় আর মিথ্যা অপারেন্ডের আগে not দিলে তা সত্য হয়ে যায়।	(a not b)

৫. Bitwise operator-সমূহ বর্ণনা কর।

উত্তর: পাইথনে বিটওয়াইজ অপারেটর ছয় ধরনের, যথা-

Python Bitwise Operators	
অপারেটর	বর্ণনা
& (বিটওয়াইজ অ্যান্ড)	মূলত দুটো বিটকে গুণ করার জন্য বিটওয়াইজ অ্যান্ড ব্যবহার করা হয়।
(বিটওয়াইজ অর)	দুটো বিটকে যোগ করার জন্য বিটওয়াইজ অ্যান্ড ব্যবহার করা হয়।
^ (বিটওয়াইজ অক্স অর)	উভয় অপারেন্ড যদি একই হয় তাহলে আউটপুট হবে ০। অন্যথায় ১।
- (বিটওয়াইজ নেগেশন)	একটিমাত্র অপারেন্ড নিয়ে কাজ করে। অপারেন্ডের মান ০ হলে আউটপুট ১, অপারেন্ডের মান ১ হলে আউটপুট ০।
<< (লেফট শিফট)	ডাটা বিটকে বামদিকে শিফট করার জন্য লেফট শিফট অপারেটর ব্যবহার করা হয়।
>> (রাইট শিফট)	ডাটা বিটকে ডানদিকে শিফট করার জন্য রাইট শিফট অপারেটর ব্যবহার করা হয়।

৬. আসাইনমেন্ট অপারেটর কাকে বলে এবং প্রোগ্রামে এটি কেন ব্যবহৃত হয়? [বাকশিবো: '২৩]

উত্তর: আসাইনমেন্ট অপারেটর (Assignment Operator) হলো এমন একটি প্রতীক বা অপারেটর যা একটি ভেরিয়েবলের মধ্যে কোনো মান (Value) বা কোনো রাশির ফলাফল (Expression Result) সংরক্ষণ করার জন্য ব্যবহৃত হয়। প্রোগ্রামিংয়ে এর প্রধান প্রতীক হলো সমান চিহ্ন (=)।

প্রোগ্রামে ব্যবহারের কারণ (Why it is Used in Programs):

প্রোগ্রামিংয়ে অ্যাসাইনমেন্ট অপারেটর ব্যবহার করার প্রধান উদ্দেশ্য এবং কারণগুলো নিচে দেওয়া হলো:

ডেটা সংরক্ষণ (Data Storage)

এটি প্রোগ্রামের সবচেয়ে মৌলিক কাজ—ডেটা মেমরিতে সংরক্ষণ করা। ব্যবহারকারীর ইনপুট, গণনা করা ফলাফল বা স্থির মানকে ভেরিয়েবলে ধরে রাখার জন্য এটি অপরিহার্য।

ভেরিয়েবল ইনিশিয়ালাইজেশন (Variable Initialization)

প্রোগ্রাম শুরু করার সময় যেকোনো ভেরিয়েবলকে একটি প্রারম্ভিক মান (Starting Value) দেওয়ার জন্য এটি ব্যবহৃত হয়।

স্টেট আপডেট (State Update)

প্রোগ্রামের কার্যকরকরণের সময় (Runtime) কোনো ভেরিয়েবলের মান পরিবর্তন বা আপডেট করার জন্য এটি লাগে। যেমন, লুপ (Loop) চালানোর সময় একটি কাউন্টার ভেরিয়েবলের মান বাড়ানোর জন্য:

- `count = count + 1`

কোড ছোট করা (Shorthand/Compact Code)

অ্যাসাইনমেন্ট অপারেটরটির বিভিন্ন সংক্ষিপ্ত রূপ (Shorthand Operators) আছে, যা কোডকে আরও সংক্ষিপ্ত এবং পঠনযোগ্য করে তোলে। যখন একটি ভেরিয়েবলের মান, সেই ভেরিয়েবলটির মাধ্যমেই আপডেট করা হয়, তখন এটি ব্যবহৃত হয়।

৬. তিনটি সংখ্যার মাঝে বড় সংখ্যা নির্ণয়ের প্রোগ্রাম লিখ।

[বাকশিবো: '২৪]

উত্তর:


```
num1 = int(input("প্রথম সংখ্যাটি প্রবেশ করান (A): "))
num2 = int(input("দ্বিতীয় সংখ্যাটি প্রবেশ করান (B): "))
num3 = int(input("তৃতীয় সংখ্যাটি প্রবেশ করান (C): "))
```

```
largest = 0
```

```
if (num1 >= num2) and (num1 >= num3):
```

```
    largest = num1
```

```
elif (num2 >= num1) and (num2 >= num3):
```

```
    largest = num2
```

```
else:
```

```
    largest = num3
```

```
print(f"\nপ্রদত্ত সংখ্যাগুলোর মধ্যে সবচেয়ে বড় সংখ্যাটি হলো: {largest}")
```

৬. $x = \frac{a(\frac{b}{c})}{2a^2}$ এই সমীকরণটি পাইথন এক্সপ্রেশনে রূপান্তর কর। [বাকশির্বো: '২৩]

উত্তর:

গাণিতিক নিয়ম অনুসারে, $x = \frac{a(\frac{b}{c})}{2a^2}$ কে $\frac{ab}{2a^2c}$ এবং অবশেষে $\frac{b}{2ac}$ এ সরল করা যায়।

এই সরলীকৃত রূপের পাইথন এক্সপ্রেশন:

```
X = b / (2 * a * c)
```

রচনামূলক প্রশ্নোত্তর :

১. পাইথনের বিভিন্ন প্রকার Operator-গুলো বর্ণনা কর।

উত্তর: . পাইথনের বিভিন্ন প্রকার Operator-গুলো বর্ণনা নিম্নরূপ;

ক্র.	অপারেটরের ধরন	কাজের সংক্ষিপ্ত বর্ণনা	উদাহরণ
১	Arithmetic Operators	গাণিতিক কাজ করে	+, -, *, /, //, %, **
২	Assignment Operators	মান সংরক্ষণ ও পরিবর্তন	=, +=, -=, *=, /=, %=
৩	Comparison Operators	দুটি মান তুলনা করে	==, !=, >, <, >=, <=
৪	Logical Operators	শর্ত যাচাই করে	and, or, not
৫	Bitwise Operators	বাইনারি সংখ্যায় কাজ করে	&
৬	Membership Operators	কোনো মান কোনো list/string-এ আছে কি না দেখে	in, not in
৭	Identity Operators	দুটি অবজেক্ট একই কি না যাচাই করে	is, is not

২. দ্বিঘাত সমীকরণের মান নির্ণয়ের জন্য একটি প্রোগ্রাম লিখ।

[বাকশি: '২৩', '২৪']

উত্তর:

```
import math
```

```
a = float(input("a: "))
```

```
b = float(input("b: "))
```

```
c = float(input("c: "))
```

$$d = (b^2) - (4 * a * c)$$

if d >= 0:

$$\text{root1} = (-b + \text{math.sqrt}(d)) / (2 * a)$$

$$\text{root2} = (-b - \text{math.sqrt}(d)) / (2 * a)$$

print("মূল 1:", root1)

print("মূল 2:", root2)

else:

print("নিশ্চায়ক ঋণাত্মক। মূলগুলি জটিল (অবাস্তব)।")

অধ্যায়
০৫

ব্রাঞ্চিং স্ট্রাকচার

৫.১ অধ্যায়ের প্রধান সূচিপত্র (contents)

বিষয়	পৃষ্ঠা নং
❖ সূত্রাবলী ও শর্টকাট টেকনিক	
❖ অতি সংক্ষিপ্ত প্রশ্নাবলি	
❖ সংক্ষিপ্ত প্রশ্নাবলি	
❖ রচনামূলক প্রশ্নাবলি	

বোর্ড বোর্ড প্রশ্নাবলির বিশ্লেষণ (Board Questions Analysis) প্রশ্নাবলির বিশ্লেষণ (Board Questions Analysis)

এই অধ্যায় থেকে বিগত পাঁচ বছরের ডিপ্লোমা ইন ইঞ্জিনিয়ারিং, ডিপ্লোমা ইন টেক্সটাইল, ডিপ্লোমা ইন এগ্রিকালচার, ডিপ্লোমা ইন মেরিন পরীক্ষায় মোট প্রশ্ন। নিচে ‘Board Questions Analysis’ অংশে এই অধ্যায় থেকে কোন সালে কোন বোর্ডে কতটি প্রশ্ন হয়েছে তা দেখানো হলো।

সাল \ বোর্ড	অতি সংক্ষিপ্ত	সংক্ষিপ্ত প্রশ্ন	রচনামূলক প্রশ্ন
২০২৪	১টি	১টি	—
২০২৩	১টি	—	১টি
২০২২	১টি	—	—
২০২১	—	—	—



“একটি কম্পিউটার তখনই বুদ্ধিমান বলা যায়, যখন এটি মানুষকে প্রভাবিত করে নিজেকে মানুষ ভাবতে পারে।”

- অ্যালান টিউরিং (Alan Turing)

অতি সংক্ষিপ্ত প্রশ্নোত্তর :

১. While loop-এর অসুবিধা লেখ।

[বাকশিৰো: '২০R]

উত্তর: While লুপে প্রথমেই শর্তের মান পরীক্ষিত হয়, শর্ত সত্য হলে তবেই লুপ বড়ি সম্পাদিত হয়। অসুবিধা হল, শর্ত সত্য না হলে while লুপের সাথে সংশ্লিষ্ট স্টেটমেন্ট সম্পাদিত হয় না।

২. Nested if statement বলতে কী বোঝায়?

[বাকশিৰো: '১৯, '২০]

উত্তর: 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 কোন ব্লককে রান করানো যায়। যদি কন্ডিশনটি বা এক্সপ্রেশনটি সত্য হয় তাহলে এর আওতাভুক্ত স্টেটমেন্ট বা স্টেটমেন্টসমূহ রান হয় আর মিথ্যা হলে পরবর্তী কোনো অ্যাকশন সম্পন্ন হবে না।

৩. if-else স্টেটমেন্টের সাধারণ ফরম্যাট লেখ।

[বাকশিৰো: '১৮, '১৯, '২০]

অথবা, if-else statement-এর Syntax লেখ।

অথবা, if-else statement-এর Formula লেখ।

উত্তর: if-else স্টেটমেন্টের সাধারণ ফরম্যাট হলো:

if Condition:

Statements

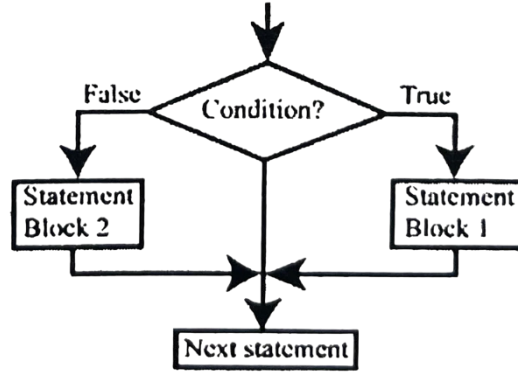
else:

Statements

৪. if-else statement-এর Flowchart অঙ্কন কর।

[বাকশিৰো: '২০R]

উত্তর: if-else statement-এর Flowchart নিম্নরূপ:



৫. প্রোগ্রাম ইনডেন্টেশন কী?

[বাকশির্বো: '১৭]

উত্তর: প্রোগ্রামের কোন স্টেটমেন্ট এর সামনে হোয়াইট স্পেস (সাধারণত চারটি স্পেস)-কে ইনডেন্টেশন বলে। ট্যাব বাটন ব্যবহার করেও প্রোগ্রামকে সহজে ইনডেন্ট করা যায়।

৬. স্টেটমেন্ট কাকে বলে?

উত্তর: কতিপয় টোকেন, কী-ওয়ার্ড, আইডেন্টিফায়ার, অপারেটর ও অপারেন্ড-এর সমন্বয়ে গঠিত এক্সপ্রেশনকেই স্টেটমেন্ট বলে।

৭. কন্ডিশনাল ও আনকন্ডিশনাল ব্রাঞ্চিং স্টেটমেন্টের উদাহরণ দাও।

অথবা, কন্ডিশনাল ও আনকন্ডিশনাল ব্রাঞ্চিং এর ১টি করে উদাহরণ দাও।

উত্তর: কন্ডিশনাল ব্রাঞ্চিং স্টেটমেন্টের উদাহরণ:

☆ if স্টেটমেন্ট

☆ if-else স্টেটমেন্ট

☆ else if স্টেটমেন্ট

আনকন্ডিশনাল ব্রাঞ্চিং স্টেটমেন্টের উদাহরণ:

☆ for স্টেটমেন্ট

☆ while স্টেটমেন্ট

☆ do-while স্টেটমেন্ট

☆ continue স্টেটমেন্ট

☆ go to স্টেটমেন্ট ইত্যাদি।

৮. Nested if else স্টেটমেন্টটি লেখ।

অথবা, Nested if else statement এর গঠন লেখ।

অথবা, Nested if else এর ফরম্যাট লেখ।

উত্তর: nested if else স্টেটমেন্টটি নিম্নরূপ:

if Condition1:

Statements

else:

if Condition2:

Statements

.....

.....

else:

if Condition n:

Statements

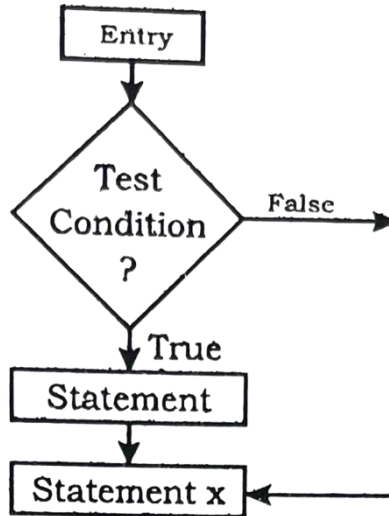
else:

Statements

৯. if statement-এর Flowchart অঙ্কন কর।

[বাকশিবো: '২২]

উত্তর: if statement-এর Flowchart নিম্নরূপ:



১০. If Statement এর সিনট্যাক্স লেখ।

[বাকশিবো: '২৩]

উত্তর:

if শর্ত:

শর্ত সত্য হলে এই কোড কার্যকর হবে (ইন্ডেন্ট করা অংশ)

বিবৃতি_১

বিবৃতি_২

ইন্ডেন্টেশন শেষ হলে if ব্লকের সমাপ্তি বোঝায়

১১. Branching কয়প্রকার ও কিকি?

[বাকশিবো: '২৪]

উত্তর:

প্রোগ্রামিংয়ে ব্রাঞ্চিং (Branching) বা শর্তাধীন নিয়ন্ত্রণ কাঠামো (Conditional Control Structure) মূলত দুই প্রকার। এগুলো হলো:

- দ্বি-মুখী ব্রাঞ্চিং (Two-way Branching)
- বহু-মুখী ব্রাঞ্চিং (Multi-way Branching)

সংক্ষিপ্ত প্রশ্নোত্তর :

১. if else statement-এর গঠন ও ফ্লোচার্ট বর্ণনা কর।

[বাক্যাশির্বো: '১৮, '২০R]

অথবা, if else statement-এর ফ্লোচার্ট বর্ণনা কর।

[বাক্যাশির্বো: '২০R]

অথবা, if else statement-এর Syntax লেখ।

[বাক্যাশির্বো: '১৯]

উত্তর:

নিচে if else statement-এর গঠন ও ফ্লোচার্ট বর্ণনা করা হলো-

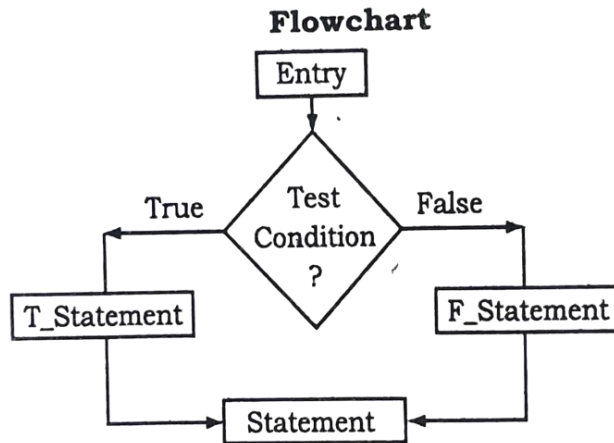
এটি if statement এরই একটি বর্ধিত রূপ। এই statement-টি test expression-এর ফলাফল true হতে true-block এর স্টেটমেন্টগুলো নির্বাহ করবে আর false হলে false-block এর স্টেটমেন্টগুলোর নির্বাহ করবে। নিম্নে সিনটেক্স দেওয়া হলো:

if (test-expression)

 true-block statement

else

 false-block-statement



চিত্র: if else statement এর flowchart

text-expression-এর ফলাফল true হলে true-block এর statement গুলোর কাজ করবে আর false হলে else এর পরে অর্থাৎ false-block এর কাজ করবে।

২. Leap yeat যাচাই করার জন্য একটি Program লেখ।

[বাকশিবো: '২০R]

উত্তর: Leap yeat যাচাই করার জন্য একটি Program নিচে দেওয়া হলো-

```

1  # code: Learn With Hafiz Vaiya
2  year = int(input("Type Your Year Here: "))
3  if (year % 400 == 0) or (year % 100 != 0) and (year % 4 == 0):
4      print("This Year Is Leap Year:", year)
5  else:
6      print("This Year Is Not Leap Year:", year)
7
8  # output:
9  # type Your Year here: 2000
10 # This Year Is Leap Year: 2000
11
12
  
```

৩. N একটি সংখ্যা। এটি জোড় না বিজোড়, তা নির্ণয় করার জন্য if-else ব্যবহার করে একটি প্রোগ্রাম লেখ।

[বাকশিবো: '১৭]

উত্তর:

```
x = int(input('Enter the number:'))
```

```
    if x %2 == 0:
```

```
        print ('Even')
```

```
    else:
```

```
        print('Odd')
```

8. if-else ব্যবহার করে দুটি সংখ্যার মাঝে বড় সংখ্যাটি প্রিন্ট করার একটি প্রোগ্রাম লেখ।

[বাকশির্বো: ১৭]

উত্তর: নিচে if-else ব্যবহার করে দুটি সংখ্যার মাঝে বড় সংখ্যাটি প্রিন্ট করার একটি প্রোগ্রাম দেওয়া হলো-

```
a = 20
```

```
b = 30
```

```
if a > b:
```

```
    print ("a is larger than b")
```

```
else:
```

```
    print ("b is larger than a")
```

b is larger than a

৫. Branching Looping বলতে কী বোঝায়?

অথবা, ব্রাঞ্চিং ও লুপিং এর পার্থক্য লেখ।

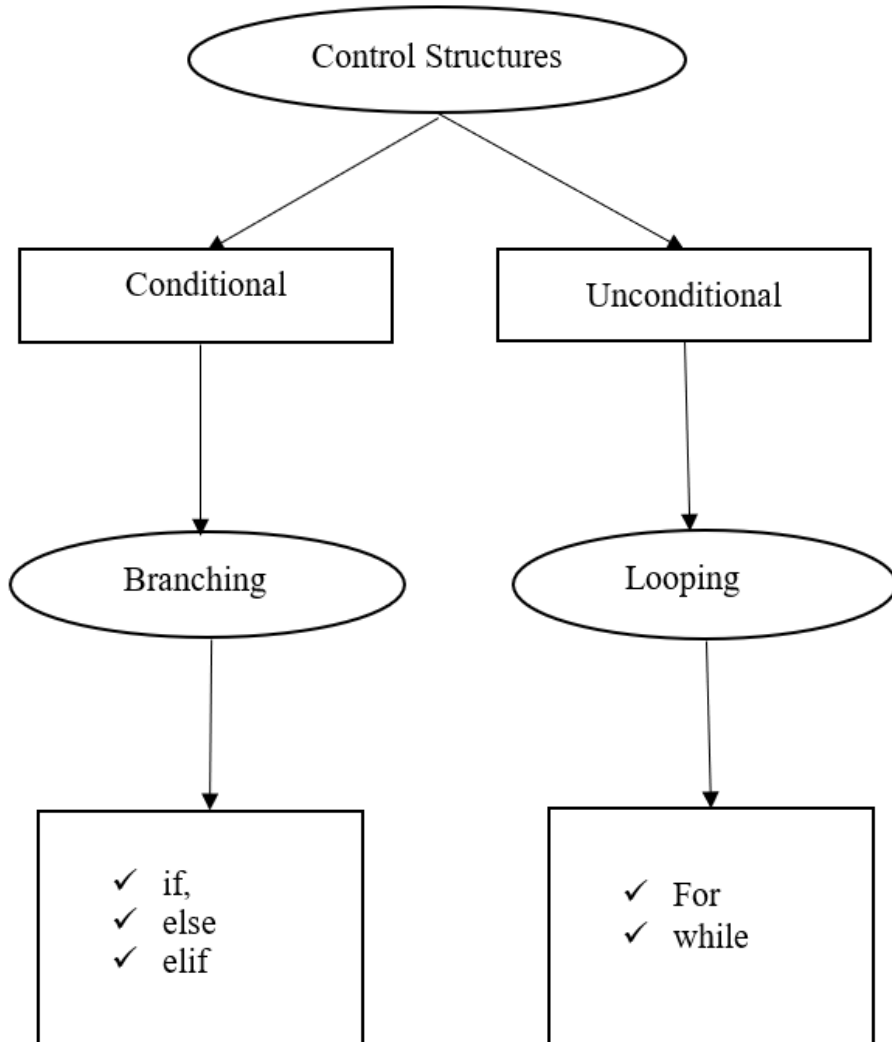
উত্তর: ব্রাঞ্চিং (Branching): প্রোগ্রামের কোনো অংশবিশেষ যদি নির্দিষ্ট শর্ত সত্য হলে এক ধরনের ফলাফল এবং মিথ্যা ০ হলে অন্য ধরনের ফলাফল প্রদান করে, তবে তাকে ব্রাঞ্চিং বলে।

লুপিং (Looping): প্রোগ্রামের কোনো অংশবিশেষ নির্দিষ্ট সংখ্যকবার বা প্রদত্ত কোনো শর্তে না পৌঁছা পর্যন্ত পুনরাবৃত্তি করাকে লুপিং বলা হয়।

৬. Conditional branching এর flowchart অংকন করো।

[বাকশির্বো: ২৪]

উত্তর:



রচনামূলক প্রশ্নোত্তর :

১. দ্বিঘাত সমীকরণের মূল বের করার Program লেখ।

[বাকশিৰো: '১৮, '১৯]

উত্তর: 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**2) - (4*a*c)

if(d==0)

    x=(-b/(2*a))

    print("Roots are real and value is: ", x)
elif(d>0)

    x1= ((-b+math.sqrt(d))/(2*a))
    x2= ((-b-math.sqrt(d))/(2*a))

    print("Roots are real and values are ",x1, "And", x2)
else

    print("Roots are imaginary")
```

আউটপুট:

Enter the value of a: 2

Enter the value of b: 4

Enter the value of c: 2

Roots are real and value is. -1.0

২.তিনটি সংখ্যার মাঝে বড় সংখ্যাটি বের করার পাইথন Program লেখ। [বাকশিৰো: '১৮, '২০]

উত্তর: প্রোগ্রাম:

```
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(a, "is the largest number.")
    else:
        print(c, "is the largest number.")
elif(b>c):
    print(b, "is the largest number.")
else:
    print(c, "is the largest number.")
```

আউটপুট:

Enter the value of a: 15

Enter the value of b: 30

Enter the value of c: 9

30 is the largest number.

৩. তিন বাহুর মান দিয়ে ত্রিভুজের সম্ভাব্যতা যাচাইপূর্বক ক্ষেত্রফল নির্ণয়ের একটি প্রোগ্রাম তৈরি কর।

[বাকশিৰো: '১৭, ১৮, ২০, ২১']

উত্তর:

```
import math
```

```
a = float(input("বাহু a: "))
```

```
b = float(input("বাহু b: "))
```

```
c = float(input("বাহু c: "))
```

```
if a + b > c and a + c > b and b + c > a:
```

```
s = (a + b + c) / 2
```

```
area = math.sqrt(s * (s - a) * (s - b) * (s - c))
```

```
print(f"ত্রিভুজের ক্ষেত্রফল: {area:.2f}")
```

```
else:
```

```
print("ত্রিভুজ গঠন সম্ভব নয়।")
```

অধ্যায়
০৬

লুপিং স্ট্রাকচার

৬.১ অধ্যায়ের প্রধান সূচিপত্র (contents)

বিষয়	পৃষ্ঠা নং
❖ সূত্রাবলী ও শর্টকাট টেকনিক	
❖ অতি সংক্ষিপ্ত প্রশ্নাবলি	
❖ সংক্ষিপ্ত প্রশ্নাবলি	
❖ রচনামূলক প্রশ্নাবলি	

বোর্ড বোর্ড প্রশ্নাবলির বিশ্লেষণ (Board Questions Analysis) প্রশ্নাবলির বিশ্লেষণ (Board Questions Analysis)

এই অধ্যায় থেকে বিগত পাঁচ বছরের ডিপ্লোমা ইন ইঞ্জিনিয়ারিং, ডিপ্লোমা ইন টেক্সটাইল, ডিপ্লোমা ইন এগ্রিকালচার, ডিপ্লোমা ইন মেরিন পরীক্ষায় মোট প্রশ্ন। নিচে ‘Board Questions Analysis’ অংশে এই অধ্যায় থেকে কোন সালে কোন বোর্ডে কতটি প্রশ্ন হয়েছে তা দেখানো হলো।

সাল \ বোর্ড	অতি সংক্ষিপ্ত	সংক্ষিপ্ত প্রশ্ন	রচনামূলক প্রশ্ন
২০২৪	১টি	১টি	—
২০২৩	২টি	২টি	—
২০২২	১টি	১টি	—
২০২১	—	—	—



“প্রোগ্রাম আসলে মানুষের পড়ার জন্য লেখা হয়, কম্পিউটার তো শুধু সেটি চালায় মাত্র।”

- ডোনাল্ড নুথ (Donald Knuth)

অতি সংক্ষিপ্ত প্রশ্নোত্তর :

১. break স্টেটমেন্টের কাজ কী?

[বাকাশিবো: '২০R, '২১, '২৩]

উত্তর: 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।

২. continue স্টেটমেন্টের কাজ কী?

[বাকাশিবো: '১৭, '২১]

উত্তর: Continue স্টেটমেন্টের কাজ হলো Continue স্টেটমেন্ট পাওয়া মাত্রই লুপের এক্সিকিউশন তাৎক্ষণিকভাবে থেমে যাবে ঠিকই কিন্তু লুপ থেকে বের হবে না। শুধুমাত্র কন্ডিশনের অংশটুকু বাদ দেবে এবং প্রোগ্রামের বাকি অংশের কাজ সম্পন্ন করবে।

৩. ইনফিনিট লুপিং বলতে কী বোঝায়?

[বাকাশিবো: '২১]

উত্তর: প্রোগ্রামে এক বা একাধিক স্টেটমেন্টকে শর্তহীনভাবে বা অনিদিষ্ট সংখ্যকবার আবর্তন করানোর প্রক্রিয়াকে ইনফিনিট (Infinite) লুপিং বলে।

৪. while লুপের সিনট্যাক্স লেখ।

[বাকাশিবো: '১৮, '২০, '২২]

উত্তর: while লুপের সাধারণ গঠন হচ্ছে:

while Condition:

Statements

while লুপের কন্ডিশনটি যতক্ষণ পর্যন্ত সত্য হবে, ততক্ষণ পর্যন্ত while লুপটি চলবে। কন্ডিশন মিথ্যা হলেই প্রোগ্রামের এক্সিকিউশন শেষ হয়ে যাবে।

৫. দুটি লুপিং স্টেটমেন্টের নাম লেখ।

[বাকাশিবো: '১৭]

উত্তর: পাইথনে ব্যবহৃত দুটি লুপিং ফ্লো স্টেটমেন্ট হলো:

> for স্টেটমেন্ট এবং

> while স্টেটমেন্ট

৬. লুপিং বলতে কী বোঝায়?

[বাকশির্বো: '২৩]

উত্তর: প্রোগ্রামের কোনো অংশবিশেষ নির্দিষ্ট সংখ্যক বার বা প্রদত্ত কোনো শর্তে না পৌঁছা পর্যন্ত অথবা অনির্দিষ্ট সংখ্যক বার পুনরাবৃত্তি করাকে লুপিং বলা হয়।

৭. লুপিং এর প্রকারভেদ উল্লেখ কর।

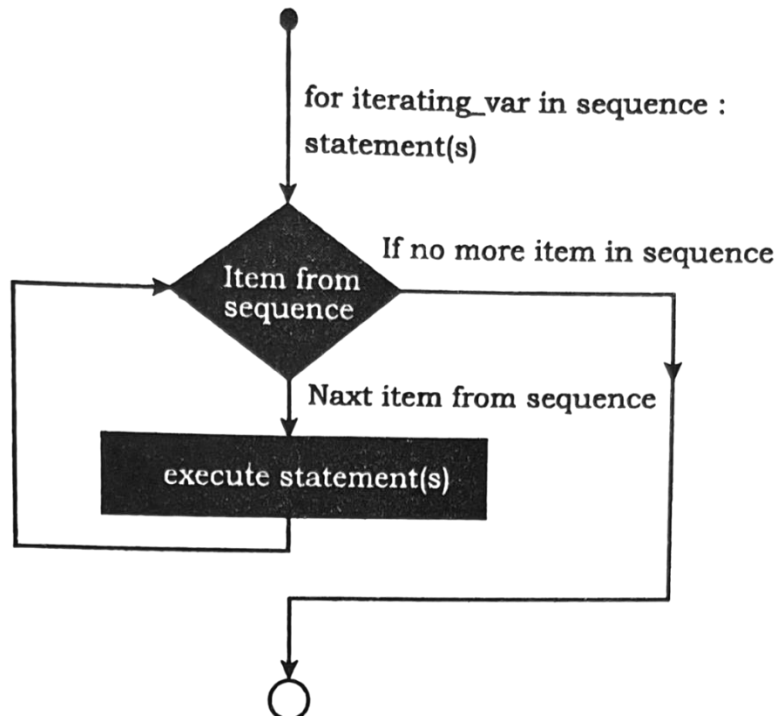
উত্তর: লুপিং দুই প্রকার, যথা:

> কন্ডিশনাল লুপিং ফ্লো (Conditional Looping Flow) ও

> আনকন্ডিশনাল লুপিং ফ্লো (Unconditional Looping Flow)

৮. for লুপের ফ্লোচার্ট উল্লেখ কর।

উত্তর: for লুপের ফ্লোচার্ট হলো:



৯. range () ফাংশনে কয় ধরনের ও কী কী প্যারামিটার আছে?

উত্তর: range () Function-এ দুই সেট প্যারামিটার আছে, যথা:

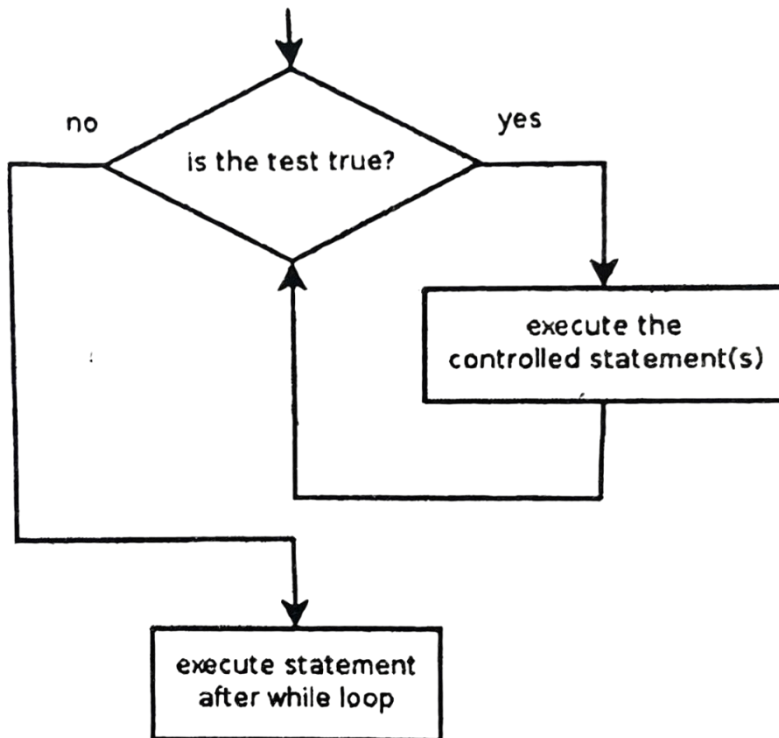
- range (Stop).
- range(Start, Stop, Step)

এখানে

- Start হলো সিকোয়েন্সের ইনিশিয়াল ভ্যালু (শুরুর মান)
- Stop হলো সর্ববিশেষ মানের পূর্বের মান এবং
- Step হলো কাউন্টার ভেরিয়েবলের ইনক্রিমেন্ট বা ডিক্রিমেন্ট (অর্থাৎ মান কত করে বাড়বে বা কমবে) ভ্যালুকে রাখায়।

১০. while লুপের ফ্লোচার্ট উল্লেখ কর।

উত্তর: while লুপের ফ্লোচার্ট হলো:



১১. Nasted loop কি?

[বাকশিবো: '২৪]

উত্তর: নেস্টেড লুপ (Nested Loop) হলো প্রোগ্রামিং-এর একটি কাঠামো, যেখানে একটি লুপের (Loop) বডির ভেতরে অন্য একটি বা একাধিক লুপ ব্যবহার করা হয়। সহজ কথায়, একটি লুপের মধ্যে আরেকটি লুপ থাকাই হলো নেস্টেড লুপ।

সংক্ষিপ্ত প্রশ্নোত্তর :

১. while লুপ ব্যবহার করে 1 থেকে 100 পর্যন্ত সংখ্যাগুলোর যোগফল নির্ণয়ের প্রোগ্রাম লেখ।

[বাকশিবো: '১৮, '২২]

উত্তর: 1. থেকে 100 পর্যন্ত সংখ্যাগুলোর যোগফল নির্ণয়ের প্রোগ্রাম নিম্নরূপ:

```
n = 1
Sum = 0
while n < 100
    Sum = Sum + n
    n = n + 1
print (Sum)
```

২. for ও while লুপের মূল পার্থক্য কী?

[বাকশিবো: '১৭]

উত্তর: for ও while লুপের মূল পার্থক্য হলো:

(i) লুপটি কতবার নির্বাহ হবে। লুপটি কতবার নির্বাহ হবে তা পূর্ব থেকে জানা না থাকলে for লুপ ব্যবহার করা যায় না। while লুপ ব্যবহার করা যায়।

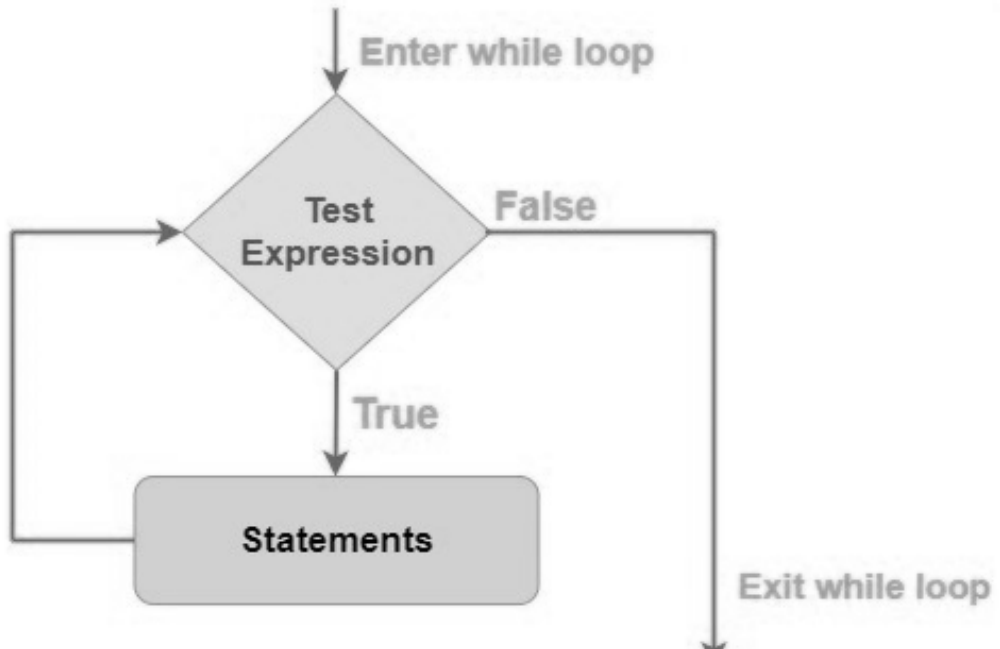
(ii) while লুপে কন্ডিশনের বাইরে কাউন্টারকে ইনিশিয়ালাইজ করতে হয়। কিন্তু for লুপে কন্ডিশনের ভেতরে এবং বাইরে সর্বত্রই কাউন্টারকে ইনিশিয়ালাইজ করা যায়।

(iii) while লুপ স্ট্রীকচারের সংক্ষিপ্ত উপস্থাপনই হচ্ছে for লুপ স্ট্রীকচার।

৩. while loop এর ফ্লোচার্ট অঙ্কন কর।

[বাকশির্বো: '২৩]

উত্তর:



৪. nested loop এর syntax লেখ।

[বাকশির্বো: '২৩]

উত্তর: for outer_variable in outer_sequence:

Outer loop body

for inner_variable in inner_sequence:

Inner loop body

Code here runs inside the inner loop

রচনামূলক প্রশ্নোত্তর :

১. কোন একটি সংখ্যা ইনপুট নিয়ে এর নামতা প্রিন্ট করার প্রোগ্রাম লেখ।

উত্তর: for লুপ ব্যবহার করে একটি পূর্ণসংখ্যা ইনপুট নিয়ে ঐ পূর্ণসংখ্যার নামতা প্রিন্ট করার

প্রোগ্রাম:

```
number = Int(input("Enter the number: "))
```

```
for x in range(1,11):
```

```
    print(number, "*",x "=", number*x)
```

আউটপুট;

Enter the number: 17

17*1 = 17

17*2 = 34

17*3 = 51

17*4 = 68

17*5 = 85

17*6 = 102

17*7 = 119

17*8 = 135

17*9 = 153

17*10 = 170

২. লিস্টের সঙ্গে for লুপের ব্যবহার দেখাও।

উত্তর: for এবং While স্টেটমেন্টের সিনটাক্স (The syntax of for & while statements):

For বাপ স্টেটমেন্ট (for loop statement) পাইথন প্রোগ্রামে এক বা একাধিক স্টেটমেন্ট একটি নির্দিষ্ট সংখ্যকবার এক্সিকিউট করার অন্য বা স্ট্রিং) পুনরাবৃত্তি করার জন্য for ব্যবহৃত লুপ হয়। সাধারণ কোনো পেরিয়েবল ব্যবহার করার পুর আবর্তন সংখ্যা গণনা করা হয়। এরূপ ভেরিয়েবলকে কাউন্টার ভেরিয়েবল বলে।

সিনটাক্স (Syntax):

```
for iterator_var in sequence range (Number of Times):
```

statements(s)

এখানে, for = keyword

Iterator_var = iterate করার জন্য ভেরিয়েবল

in = keyword

sequence = লুপ-এর সীমা, অর্থাৎ যতক্ষণ লুপ চলবে।

উল্লেখ, for লুপ-এর statements(s) অংশ অবশ্যই ইন্ডেন্টেশন দিতে হবে।

উদাহরণ-১:

for name in range (5):

print("TAMIM AL IQBAL")

উদাহরণ-১ প্রোগ্রামে for লুপ ব্যবহার করে TAMIM AL IQBAL নামটি ৫ বার প্রিন্ট করা হয়েছে। এক্ষেত্রে range() ফাংশনে আর্গুমেন্ট ৫ ব্যবহার করা হয়েছে।

উদাহরণ-২:

1+2+3+.....+9+10=? নির্ণয়ের প্রোগ্রাম।

প্রোগ্রাম; sum = 0

for i in range (1, 11)

sum = sum + i

print("sum of first 10 number is", sum)

আউটপুট:

sum of first 10 number is 55

অধ্যায়
০৭

লিস্ট স্ট্রাকচার

৭.১ অধ্যায়ের প্রধান সূচিপত্র (contents)

বিষয়	পৃষ্ঠা নং
❖ সূত্রাবলী ও শর্টকাট টেকনিক	
❖ অতি সংক্ষিপ্ত প্রশ্নাবলি	
❖ সংক্ষিপ্ত প্রশ্নাবলি	
❖ রচনামূলক প্রশ্নাবলি	

বোর্ড বোর্ড প্রশ্নাবলির বিশ্লেষণ (Board Questions Analysis) প্রশ্নাবলির বিশ্লেষণ (Board Questions Analysis)

এই অধ্যায় থেকে বিগত পাঁচ বছরের ডিপ্লোমা ইন ইঞ্জিনিয়ারিং, ডিপ্লোমা ইন টেক্সটাইল, ডিপ্লোমা ইন এগ্রিকালচার, ডিপ্লোমা ইন মেরিন পরীক্ষায় মোট প্রশ্ন। নিচে ‘Board Questions Analysis’ অংশে এই অধ্যায় থেকে কোন সালে কোন বোর্ডে কতটি প্রশ্ন হয়েছে তা দেখানো হলো।

সাল \ বোর্ড	অতি সংক্ষিপ্ত	সংক্ষিপ্ত প্রশ্ন	রচনামূলক প্রশ্ন
২০২৪	১টি	—	—
২০২৩	—	—	১টি
২০২২	১টি	—	—
২০২১	—	—	—



“কথা সহজ—আমাকে কোড দেখাও।”

- লিনাস টরভাল্ডস (Linus Torvalds)

অতি সংক্ষিপ্ত প্রশ্নোত্তর :

১. লিস্ট ও অ্যারের মধ্যে পার্থক্য লেখ।

[বাকশির্বো: '২০R]

উত্তর: সি, সি++ বা জাভা ল্যাংগুয়েজে যাকে অ্যারে বলা হয় পাইথনে তাকেই লিস্ট বলে। তবে লিস্ট অ্যারের চেয়েও স্মার্ট। কারণ- প্রোগ্রামিং এ লিস্টের প্রায়োগিক সক্ষমতা অনেক বেশি।

২. লিস্ট কাকে বলে?

[বাকশির্বো: '২০]

অথবা, লিস্ট বলতে কী বোঝায়?

[বাকশির্বো: '১৮, '১৯]

উত্তর: লিস্ট হচ্ছে পাইথনের সবচেয়ে বৈচিত্র্যপূর্ণ ডাটা টাইপ, যা স্কয়ার ব্রাকেট {} এর ভেতর কমার সাহায্যে উপাদানসহ প্রকাশ করা হয়। অন্যান্য ল্যাংগুয়েজে যাকে অ্যারে বলা হয় পাইথনে তাকেই লিস্ট বলে।

৩. all() ও any() ফাংশনের মধ্যে পার্থক্য কী?

[বাকশির্বো: '১৯]

উত্তর: লিস্টের সকল উপাদানের মান সত্য হলে আউটপুট হিসেবে all() ফাংশনটি True রিটার্ন করে। অপরদিকে লিস্টের যে-কোনো একটি উপাদানের মান সত্য হলেও আউটপুট হিসেবে any() ফাংশনটি True রিটার্ন করে। লিস্টটি empty হলেই কেবল আউটপুট হিসেবে False রিটার্ন করে।

৪. স্লাইসিং বলতে কী বোঝায়?

[বাকশির্বো: '১৯]

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

৫. লিস্ট এলিমেন্ট কত প্রকার ও কী কী?

উত্তর: পাইথনের লিস্ট এলিমেন্ট বিভিন্ন টাইপের হতে পারে, যেমন-

- ইন্টিজার টাইপ ডাটা
- ফ্লোটিং পয়েন্ট টাইপ ডাটা

- স্ট্রিং
- লিস্ট
- টাপল
- ডিকশনারি ও
- মিক্সড ডাটা টাইপ ইত্যাদি।

৬. লিস্টের উপাদানে অ্যাক্সেসিং পদ্ধতিসমূহের নাম লেখ।

উত্তর: লিস্টের উপাদানসমূহকে অ্যাক্সেস করার জন্য পাইথনে বিভিন্ন পদ্ধতি প্রচলিত আছে।
পদ্ধতিসমূহ হচ্ছে-

- ইনডেক্সিং (Indexing) পদ্ধতি
- নেগেটিভ ইনডেক্সিং (Negative Indexing) পদ্ধতি এবং
- স্লাইসিং (Slicing) পদ্ধতি।

৭. ইনডেক্সিং বলতে কী বোঝায়?

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

৮. 'পাইথনের লিস্ট অপারেশন কত প্রকার ও কী কী?

উত্তর: পাইথনে মোট তিন ধরনের লিস্ট অপারেশন আছে। যথা:

- কনক্যাটেনেশন অপারেটর
- রিপিটিশন অপারেটর এবং
- মেম্বারশিপ অপারেটর।

৯. len() ফাংশনের কাজ কী?

[বাকশিবো, ২২, '২৩,২৪]

উত্তর: লিস্টে বিদ্যমান সকল উপাদানের সংখ্যা জানার জন্য len() ফাংশন ব্যবহৃত হয়।

সংক্ষিপ্ত প্রশ্নোত্তর :

১. উদাহরণসহ স্লাইসিং ব্যাখ্যা কর।

[বাকশিবো, ২২]

অথবা, লিস্টে স্লাইসিং পদ্ধতির ব্যবহার দেখাও।

উত্তর: স্লাইসিং (Slicing) পদ্ধতি: স্লাইসিং অপারেটর (:) বা কোলন ব্যবহার করে লিস্টের আইটেমকে অ্যাক্সেস স্লাইসিং বলে।

উদাহরণ-১:

```
b = ['Faruk', 'Muzahid', 'Asadul', 'Sabbir', 35409, 27.003]
```

```
print(b[1 : 5])
```

```
print(b[ : 5])
```

```
print(b[ 2 : ])
```

আউটপুট:

```
['Muzahid', 'Asadul', 'Sabbir', 35409]
```

```
['Faruk', 'Muzahid', 'Asadul', 'Sabbir', 35409]
```

```
['Asadul', 'Samsul', 35409, 27.003]
```

২. সিনট্যাক্স ও উদাহরণসহ pop() মেথডের ব্যবহার দেখাও।

[বাকশিবো, ২১]

উত্তর: pop() মেথড:

সিনট্যাক্স:

```
List_pop(index)
```

উদাহরণ:

```
language ['Python', 'Java', 'C++', 'Cobol', 'C']  
  
return_value = language.pop(3)  
  
print('Return Value: ', return_value)  
  
print('Updated List: ',language)
```

৩. উদাহরণসহ লিস্ট ইন্ডেক্সিং পদ্ধতি সংক্ষেপে বর্ণনা কর।

উত্তর: ইন্ডেক্সিং (Indexing) পদ্ধতি: যে পদ্ধতিতে লিস্টের কোনো আইটেমকে অ্যাক্সেস করার জন্য ইন্ডেক্স নম্বর ব্যবহার করা হয় তাতে ইন্ডেক্সিং পদ্ধতি বলে। লিস্টের ইন্ডেক্স নম্বর শুরু হয় ০ থেকে।

উদাহরণ: (i)

```
my_list = ['p','r','o','b','e']  
print(my_list[0])  
print(my_list[2])  
print(my_list[4])  
  
n_list = ["Happy",[2,0,1,5]]  
print(n_list[0][1])  
print(n_list[1][3])
```

আউটপুট:

P

0

e

a

5

৪. লিস্টে ব্যবহৃত যে-কোনো তিনটি বিল্ট ইন ফাংশনের কাজ লেখ।

উত্তর: পাইথন প্রোগ্রামে লিস্ট নিয়ে দ্রুততার সঙ্গে এবং সুনির্দিষ্ট কোনো কাজ সম্পাদনের জন্য এতে বেশ কিছু বিল্ট ইন ফাংশন রয়েছে। নিম্নে লিস্টে ব্যবহৃত এরূপ তিনটি বিল্ট ইন ফাংশন ও তাদের ব্যবহার উল্লেখ করা হলো:

পাইথনের লিস্টে ব্যবহৃত বিল্ট ইন ফাংশনসমূহ;	
বিল্ট ইন ফাংশন	বর্ণনা
all()	লিস্টের সকল উপাদানের মান সত্য হলে আউটপুট হিসেবে এই ফাংশনটি True রিটার্ন করে।
any()	লিস্টের যে-কোনো একটি উপাদানের মান সত্য হলেও আউটপুট হিসেবে এই ফাংশনটি True রিটার্ন করে। লিস্টটি empty হলেই কেবল আউটপুট হিসেবে False রিটার্ন করে।
len()	লিস্টে বিদ্যমান সকল উপাদানের সংখ্যা জানার জন্য এই ফাংশনটি ব্যবহৃত হয়।

রচনামূলক প্রশ্নোত্তর :

১. লিস্টে কোনো উপাদান সংযোজন, বিয়োজন ও আপডেটিং পদ্ধতি বর্ণনা কর।

[বাকশির্বো: '১৯, '২৩]

উত্তর: লিস্টে নতুন কোনো উপাদান সংযোজন (Inserting new item in the list): লিস্টে নতুন কোনো আইটেম সংযোজন করতে চাইলে পাইথনে তিন ধরনের ফাংশন ব্যবহার করা হয়; যথা- append() ফাংশন, insert() ফাংশন ও extend() ফাংশন। append() ফাংশন দিয়ে কোনো আইটেম যোগ করা হলে সেটা সবসময় লিস্টের শেষে যোগ হবে। কিন্তু insert() ফাংশন ব্যবহার করে নতুন আইটেমটিকে যে-কোনো ইনডেক্সে সংযোজন করা যায়। সেক্ষেত্রে নতুন আইটেমটিকে যে ইনডেক্সে সংযোজন করতে চাই তার ইনডেক্স নম্বর আর নতুন আইটেমটি বলে দিতে হবে। কিন্তু একাধিক আইটেম একসঙ্গে সংযোজন করার জন্য extend() ফাংশন ব্যবহার করা হয়। তবে এক্ষেত্রে আইটেমগুলো লিস্ট হিসেবে নয় বরং সিঙ্গেল আইটেম হিসেবে সংযোজিত হয়।

উদাহরণ:

```
lst = [1, 2, 3]
```

```
lst.append(4)          # [1, 2, 3, 4]
```

```
lst.insert(1, 10)     # [1, 10, 2, 3, 4]
```

```
lst.extend([5, 6])    # [1, 10, 2, 3, 4, 5, 6]
```

লিস্টের উন্নাদান আপডেটিং (Updating list elements): লিস্টের উপাদান আপডেটিং বলতে লিস্টের ইনডেক্সে যে আইটেমগুলো আছে সেগুলোর মধ্যে কোনো পরিবর্তন, পরিবর্ধন, সংযোজন, বিয়োজন ইত্যাদি অপারেশনকে বুঝায়। লিস্টের কোনো একটি আইটেম যদি Mahi হয় তাহলে তা পরিবর্তন করে Mahdoo তে রূপান্তর করার প্রক্রিয়াই হচ্ছে আপডেটিং। লিস্ট আপডেটিং-এর ক্ষেত্রে স্ট্রিংয়ের পরিবর্তে স্ট্রিং, ইন্টিজারের পরিবর্তে ইন্টিজার আর ফ্লোটের পরিবর্তে ফ্লোট

টাইপের ডাটাই ব্যবহার করতে হবে- এ রকম কোনো বিধিনিষেধ নেই। যে কোনো ডাটা টাইপের পরিবর্তে অন্য যে-কোনো ডাটা টাইপ আমরা ব্যবহার করতে পারি। লিস্ট এলিমেন্টকে আপডেট করার জন্য অ্যাসাইনমেন্ট অপারেটর ('=') ব্যবহার করা হয়।

উদাহরণ:

```
lst = [5, 6, 7]
```

```
lst[1] = 60      # [5, 60, 7]
```

লিস্টের উপাদান বিয়োজন (Deleting list elements): লিস্ট থেকে এর কোনো উপাদান রিমুভ করার জন্য পাইথনে দুই ধরনের ফাংশন রয়েছে। যথা- `del()` ফাংশন ও `remove()` ফাংশন। যে আইটেমটিকে বাস দিতে চাই তার ইনডেক্স জানা থাকলে `del()` ফাংশন ও ইনডেক্স নম্বর জানা না থাকলে `remove()` ফাংশন ব্যবহার করা হয়।

উদাহরণ:

```
lst = [1, 2, 3, 4]
```

```
lst.remove(2)      # [1, 3, 4]
```

```
lst.pop(1)         # [1, 4]
```

```
lst.clear()        # []
```

২. উদাহরণসহ বিভিন্ন প্রকার লিস্ট অপারেশন ব্যাখ্যা কর।

[বাকশিবো: '২০]

অথবা, List এর বিভিন্ন অপারেশন গুলো বর্ণনা কর।

[বাকশিবো: '১৯]

উত্তর: পাইথনে মোট তিন ধরনের লিস্ট অপারেশন আছে। যথা-

- কনক্যাটেনেশন অপারেশন (Concatenation operation)
- রিপিটিশন অপারেশন (Repetition operation)
- মেম্বারশিপ অপারেশন (Membership operation)

কনক্যাটেনেশন অপারেশন (Concatenation operation): দুটি 'List'-এর উপাদানসমূহকে সংযুক্ত করে নতুন কোনো List করাকে কনক্যাটেনেশন বলে। লিস্ট-এর উপাদানসমূহ বিভিন্ন টাইপের হলেও কোনো সমস্যা নেই। এটি কনক্যাটেনেশন অপারেটর (+) ব্যবহার করে করা হয়।

উদাহরণ:

প্রোগ্রাম :

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

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

```
List3=List1+List2
```

```
print(List3)
```

আউটপুট :

```
['Mahbub', 960021, 'Mahi', 23022015, 19072017, 'Mahdee', 'Tahsin']
```

রিপিটিশন অপারেশন (Repetition operation): পাইথনে কোনো 'List'-এ বিদ্যমান উপাদানসমূহকে একাধিকবার রিপিট করার প্রক্রিয়াকে রিপিটিশন অপারেশন বলে। রিপিটিশন অপারেটর (*) ব্যবহার করে এটি করা হয়। এক্ষেত্রেও List-এর উপাদানসমূহ বিভিন্ন টাইপের হলেও কোনো সমস্যা নেই।

উদাহরণ:

প্রোগ্রাম :

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

```
print(List1*2)
```

আউটপুট :

```
['Mahbub', 960021, 'Mahi', 23022015, 'Mahbub', 960021, 'Mahi', 23022015]
```


মেম্বারশিপ অপারেশন (Membership operation): কোনের লিস্টের মধ্যে নির্দিষ্ট কোনো এলিমেন্টের উপস্থিতি বা অনুপস্থিতি চেক করারসহব মেম্বারশিপ অপারেটর ব্যবহার করা হয়। পাইথনের লিস্ট দুই ধরনের মেম্বারশিপ অপারেটর ব্যবহৃত হয়। যেমন- `in` অপারেটর ও `not in` অপারেটর। কোনো লিস্টের মধ্যে নির্দিষ্ট কোনো এলিমেন্ট উপস্থিত আছে কি না সেটা চেক করার জন্য `in` অপারেটর ব্যবহার করা হয়। যদি এলিমেন্টটি লিস্টের মধ্যে এক বা একাধিকবার থাকে তাহলে এটি `True` রিটার্ন করে, অন্যথায় `False` রিটার্ন করে। একইভাবে লিস্টের সাথে `not in` অপারেটর ব্যবহার করে কোনো এলিমেন্টের অনুপস্থিতি চেক করা যায়।

উদাহরণ:

প্রোগ্রাম :

```
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)
```

আউটপুট :

```
False
False
True
False
```

অধ্যায়
০৮

টপল স্ট্রাকচার

১১ অধ্যায়ের প্রধান সূচিপত্র (contents)

বিষয়	পৃষ্ঠা নং
❖ সূত্রাবলী ও শর্টকাট টেকনিক	
❖ অতি সংক্ষিপ্ত প্রশ্নাবলি	
❖ সংক্ষিপ্ত প্রশ্নাবলি	
❖ রচনামূলক প্রশ্নাবলি	

বোর্ড বোর্ড প্রশ্নাবলির বিশ্লেষণ (Board Questions Analysis) প্রশ্নাবলির বিশ্লেষণ (Board Questions Analysis)

এই অধ্যায় থেকে বিগত পাঁচ বছরের ডিপ্লোমা ইন ইঞ্জিনিয়ারিং, ডিপ্লোমা ইন টেক্সটাইল, ডিপ্লোমা ইন এগ্রিকালচার, ডিপ্লোমা ইন মেরিন পরীক্ষায় মোট প্রশ্ন। নিচে ‘Board Questions Analysis’ অংশে এই অধ্যায় থেকে কোন সালে কোন বোর্ডে কতটি প্রশ্ন হয়েছে তা দেখানো হলো।

সাল \ বোর্ড	অতি সংক্ষিপ্ত	সংক্ষিপ্ত প্রশ্ন	রচনামূলক প্রশ্ন
২০২৪	—	১টি	—
২০২৩	১টি	২টি	—
২০২২	১টি	—	—
২০২১	—	—	—



“ডিবাগ করা মানে কোড লেখার চেয়েও দ্বিগুণ কঠিন কাজ।”

- ব্রায়ান কার্নিহান (Brian Kernighan)

অতি সংক্ষিপ্ত প্রশ্নোত্তর :

১. টাপল বলতে কী বোঝায়?

[বাকাশিৰো: '২০, '২২, '২৩]

অথবা, টাপল কী?

উত্তর: টাপল হচ্ছে পাইথনের অন্যতম বৈচিত্র্যপূর্ণ ডাটা টাইপ, প্যারেনথেসিস () এর ভেতর কমার সাহায্যে উপাদানসহ প্রকাশ করা হয়।

২. টাপল ও লিস্টের মধ্যে পার্থক্য কী?

উত্তর: লিস্টকে যেমন ইচ্ছেমতো পরিবর্তন, পরিবর্ধন করা যায় কিন্তু টাপলকে ইচ্ছেমতো পরিবর্তন, পরিবর্ধন করা যায় না। সাধারণত লিস্ট একই ধরনের ডাটা টাইপ ব্যবহৃত হয় কিন্তু টাপলে ভিন্ন ভিন্ন ধরনের ডাটা টাইপ ব্যবহার করা যায়।

৩. টাপলের উপাদান অ্যাক্সেস করার জন্য পাইথনে কয় ধরনের ও কী কী রয়েছে?

উত্তর: টাপলের উপাদানসমূহ-

(১) ইনডেক্সিং (Indexing) পদ্ধতি

(২) নেগেটিভ ইনডেক্সিং (Negative Indexing) পদ্ধতি এবং

(৩) স্লাইসিং (Slicing) পদ্ধতি।

রচনামূলক প্রশ্নোত্তর :

১. টাপলে ব্যবহৃত বিল্ট ইন ফাংশনসমূহ বর্ণনা কর।

[বাকশিবো: '১৮']

উত্তর:

টাপলে ব্যবহৃত বিল্ট-ইন ফাংশন (Built-in functions): পাইথন প্রোগ্রামে টাপল নিয়ে দ্রুততার সঙ্গে এবং সুনির্দিষ্ট কোনো কাজ সম্পাদনের জন্য এতে বেশ কিছু বিল্ট-ইন ফাংশন রয়েছে। নিম্নে টাপলে ব্যবহৃত এ সকল বিল্ট-ইন ফাংশনসমূহ ও তাদের ব্যবহার উল্লেখ করা হলো-

পাইথনের টাপলে ব্যবহৃত বিল্ট-ইন ফাংশনসমূহ;	
বিল্ট-ইন ফাংশন	বর্ণনা
sum()	টাপলে বিদ্যমান উপাদানসমূহের যোগফল নির্ণয়ের জন্য এই ফাংশনটি ব্যবহৃত হয়।
any()	টাপলের যে-কোনো একটি উপাদানের মান সত্য হলে ও আউটপুট হিসেবে এই ফাংশনটি True রিটার্ন করে। টাপলটি empty হলেই কেবল আউটপুট হিসেবে False রিটার্ন করে।
max()	টাপলে বিদ্যমান সকল উপাদানের মধ্যে সর্ববৃহৎ আইটেমটি জানার জন্য এই ফাংশনটি ব্যবহৃত হয়।
enumerate()	এই ফাংশনটি টাপলের সকল উপাদানের ইনডেক্স নম্বর ও তাদের মান Pair হিসেবে ধারণ করে।
len()	টাপলে বিদ্যমান সকল উপাদানের সংখ্যা জানার জন্য এই ফাংশনটি ব্যবহৃত হয়।
all()	টাপলের সকল উপাদানের মান সত্য হলে আউটপুট হিসেবে এই ফাংশনটি True রিটার্ন করে।
list()	পাইথনের ইটারেবল ডাটা স্ট্রাকচারসমূহকে (list, string, set, dictionary) টাপলে রূপান্তরের জন্য এই ফাংশনটি ব্যবহৃত হয়।
min()	টাপলে বিদ্যমান সকল উপাদানের মধ্যে সবচেয়ে ছোট আইটেমটি জানার জন্য এই ফাংশনটি ব্যবহৃত হয়।
sorted()	সর্টেড টাপল রিটার্নের জন্য এই ফাংশনটি ব্যবহৃত হয়।

২. টাপল অপারেশন কত প্রকার?

[বাকশির্ষো: '২৩']

উত্তর:

টাপলের উপর প্রধানত তিন প্রকার মৌলিক অপারেশন করা হয়-

অ্যাক্সেস অপারেশন (Access Operations):

টাপলের উপাদানগুলো অ্যাক্সেস বা পুনরুদ্ধার করার জন্য ব্যবহৃত হয়।

মডিফাই করা ছাড়া অপারেশন (Non-Mutating Operations):

এগুলো টাপলটিকে পরিবর্তন না করে নতুন টাপল তৈরি করে বা কোনো তথ্য দেয়।

বিল্ট-ইন ফাংশন ও মেথড (Built-in Functions & Methods):

টাপলের ডেটার উপর তথ্য জানার জন্য ব্যবহৃত হয়।

৩. লিস্টের তুলনায় টাপল সুবিধা কী?

[বাকশির্ষো: '২৩']

উত্তর:

- **ডেটা সুরক্ষিত রাখা:** যেহেতু টাপল অপরিবর্তনশীল, তাই ভুলবশত বা অসাবধানতাবশত এর উপাদানগুলো পরিবর্তিত হওয়ার কোনো ভয় থাকে না। যেসব ডেটার মান স্থির থাকা প্রয়োজন (যেমন: মাসের নাম, ডিকশনারির 'কী' বা কোনো ডেটা সেটের স্থানাঙ্ক), সেগুলোর জন্য টাপল ব্যবহার করা নিরাপদ।
- **ডিকশনারির কী হিসেবে ব্যবহার:** পাইথনে শুধুমাত্র অপরিবর্তনশীল ডেটা টাইপই ডিকশনারির কী (Key) হিসেবে ব্যবহার করা যায়। লিস্ট পরিবর্তনশীল হওয়ায় তা কী হিসেবে ব্যবহার করা যায় না, কিন্তু টাপল অপরিবর্তনশীল হওয়ায় এটি কী হিসেবে ব্যবহৃত হতে পারে।
- **দ্রুত কার্যকরীকরণ:** টাপল অপরিবর্তনশীল হওয়ায় পাইথনের ইন্টারপ্রেটার এটি নিয়ে কাজ করার সময় লিস্টের চেয়ে কম মেমরি ব্যবহার করে এবং দ্রুত কার্যকর (Execute)

করতে পারে। যখন ডেটা পরিবর্তন করার প্রয়োজন হয় না, তখন টাপল ব্যবহার করলে প্রোগ্রাম দ্রুত চলে।

- **মেমরি দক্ষতা:** টাপল তৈরি করার জন্য লিস্টের চেয়ে কম মেমরি ওভারহেড (Memory Overhead) লাগে।
- **একাধিক মান ফেরত:** একটি ফাংশন থেকে একসাথে একাধিক মান ফেরত দেওয়ার জন্য টাপল ব্যবহার করা একটি সাধারণ ও আদর্শ পদ্ধতি। পাইথন স্বয়ংক্রিয়ভাবে একাধিক কমা-সেপারেটেড মানকে একটি টাপল হিসেবে ফেরত দেয়।

৪. লিস্ট ও টাপল মাঝে পার্থক্য লেখ।

[বাকশির্বো: '২৪]

উত্তর:

লিস্ট (List) এবং টাপল (Tuple)-এর মধ্যে প্রধান পার্থক্যগুলো নিচে সারণী বা টেবিল ছাড়া বিস্তারিতভাবে দেওয়া হলো:

i. পরিবর্তনশীলতা (Mutability)

এটিই লিস্ট ও টাপলের মধ্যে সবচেয়ে গুরুত্বপূর্ণ পার্থক্য:

- **লিস্ট:** হলো পরিবর্তনশীল (Mutable)। এর অর্থ হলো লিস্ট তৈরি হওয়ার পরেও আপনি এর উপাদানগুলো যোগ করতে, মুছে ফেলতে বা পরিবর্তন করতে পারবেন।
 - **প্রতীক:** লিস্ট সংজ্ঞায়িত করা হয় স্কোয়ার ব্র্যাকেট ([]) ব্যবহার করে।
 - **ব্যবহার:** যেসব ডেটা সেটে ঘন ঘন পরিবর্তন আনা প্রয়োজন (যেমন: একটি শপিং কার্ট, বা একটি ডাইনামিক স্ট্যাক)।
- **টাপল:** হলো অপরিবর্তনশীল (Immutable)। একবার টাপল তৈরি হয়ে গেলে, এর ভেতরের কোনো উপাদান পরিবর্তন করা, যোগ করা বা মুছে ফেলা সম্ভব নয়।
 - **প্রতীক:** টাপল সংজ্ঞায়িত করা হয় প্যারেনথেসিস বা প্রথম বন্ধনী (()) ব্যবহার করে।

- ব্যবহার: যেসব ডেটা স্থিতিশীল থাকা প্রয়োজন (যেমন: \$X, Y\$ স্থানাঙ্ক, বা একটি ফাংশনের রিটার্ন করা একাধিক মান)।

ii. কার্যকারিতা ও গতি (Performance & Speed)

- টাপল সাধারণত লিস্টের চেয়ে দ্রুত কাজ করে। যেহেতু টাপল অপরিবর্তনশীল, পাইথনের ইন্টারপ্রেটার এটি নিয়ে কাজ করার সময় লিস্টের চেয়ে কম মেমরি ওভারহেড রাখে এবং দ্রুত প্রক্রিয়া সম্পন্ন করতে পারে।
- লিস্ট পরিবর্তনশীল হওয়ায় এর আকার এবং উপাদান যেকোনো সময় পরিবর্তিত হতে পারে। এই কারণে, এটিকে রক্ষণাবেক্ষণ করতে টাপলের চেয়ে বেশি মেমরি এবং কিছুটা বেশি প্রক্রিয়াকরণের সময় লাগে।

iii. নিরাপত্তা ও মেমরি ব্যবহার (Safety & Memory)

- টাপল: অপরিবর্তনশীলতার কারণে এটি ডেটার নিরাপত্তা নিশ্চিত করে। ভুলবশত কেউ এর মান পরিবর্তন করতে পারবে না। এছাড়া এটি ডিকশনারির কী (Key) হিসেবেও ব্যবহার করা যায়।
- লিস্ট: এটি পরিবর্তনশীল হওয়ায় ডেটা নিরাপদ রাখার জন্য এটি টাপলের মতো উপযুক্ত নয়। লিস্ট ডিকশনারির কী হিসেবে ব্যবহার করা যায় না।

অধ্যায়
০৯

সেট স্ট্রাকচার

১১ অধ্যায়ের প্রধান সূচিপত্র (contents)

বিষয়	পৃষ্ঠা নং
❖ সূত্রাবলী ও শর্টকাট টেকনিক	
❖ অতি সংক্ষিপ্ত প্রশ্নাবলি	
❖ সংক্ষিপ্ত প্রশ্নাবলি	
❖ রচনামূলক প্রশ্নাবলি	

বোর্ড বোর্ড প্রশ্নাবলির বিশ্লেষণ (Board Questions Analysis) প্রশ্নাবলির বিশ্লেষণ (Board Questions Analysis)

এই অধ্যায় থেকে বিগত পাঁচ বছরের ডিপ্লোমা ইন ইঞ্জিনিয়ারিং, ডিপ্লোমা ইন টেক্সটাইল, ডিপ্লোমা ইন এগ্রিকালচার, ডিপ্লোমা ইন মেরিন পরীক্ষায় মোট প্রশ্ন। নিচে ‘Board Questions Analysis’ অংশে এই অধ্যায় থেকে কোন সালে কোন বোর্ডে কতটি প্রশ্ন হয়েছে তা দেখানো হলো।

সাল \ বোর্ড	অতি সংক্ষিপ্ত	সংক্ষিপ্ত প্রশ্ন	রচনামূলক প্রশ্ন
২০২৪	১টি	—	১টি
২০২৩	—	—	—
২০২২	১টি	১টি	১টি
২০২১	—	—	—



“কম্পিউটার শুধু সংখ্যার ব্যাপার নয়, এটি যুক্তির জগৎ।”

- জন ভন নিউম্যান (John von Neumann)

অতি সংক্ষিপ্ত প্রশ্নোত্তর :

১. পাইথনে সেট কী?

অথবা, পাইথনে সেট কাকে বলে?

উত্তর: সেট হচ্ছে একসেট অবজেক্টের কালেকশন। এই অবজেক্টসমূহকে সেট এর মেম্বার অথবা ইলিমেন্ট বলা হয়। সেটের ইলিমেন্টসমূহ ডুপ্লিকেট হতে পারবে না। সেট এর ইলিমেন্টসমূহ যে কোন নাম্বার, ক্যারেক্টার অথবা কোন ডাটা ভ্যালু হতে পারে। সেট এর অবজেক্টসমূহ আনঅর্ডার অবস্থায় থাকে।

২. লিস্ট ও সেটের মধ্যে পার্থক্য কী?

[বাকশিবো: '২২]

উত্তর: লিস্টে একই আইটেম একাধিকবার থাকতে পারে কিন্তু সেটে একটি আইটেম কেবলমাত্র একবারই থাকতে পারে।

৩. সেট ব্যবহারের সুবিধা লেখ।

[বাকশিবো: '২৪]

উত্তর:

অনন্যতা নিশ্চিত: সেটের সবচেয়ে বড় সুবিধা হলো এটি স্বয়ংক্রিয়ভাবে ডুপ্লিকেট (Duplicate) বা সদৃশ উপাদানগুলিকে দূর করে শুধুমাত্র অনন্য (Unique) উপাদান সংরক্ষণ করে।

দ্রুত পরীক্ষা: কোনো উপাদান সেটে আছে কি না, তা পরীক্ষা করার জন্য (in অপারেটর) সেট লিস্টের চেয়ে অনেক দ্রুত কাজ করে।

সেট অপারেশন: সেট সরাসরি ইউনিয়ন, ইন্টারসেকশন এবং পার্থক্য-এর মতো গাণিতিক সেট অপারেশনগুলো সম্পাদনে সাহায্য করে।

অবিন্যস্ত ডেটা: যখন ডেটাগুলির কোনো নির্দিষ্ট ক্রমের (Order) প্রয়োজন হয় না, তখন সেট একটি দক্ষ ডেটা স্ট্রাকচার হিসেবে কাজ করে।

সংক্ষিপ্ত প্রশ্নোত্তর :

১. বিভিন্ন প্রকার সেট অপারেশনগুলো লেখ।

উত্তর: পাইথনে মোট পাঁচ ধরনের সেট অপারেশন রয়েছে। যথা:

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

২. python ভাষায় ব্যবহৃত চারটি built in function উল্লেখ কর।

[বাকশিবো: '২২']

উত্তর:

পাইথন (Python) ভাষায় ব্যবহৃত চারটি গুরুত্বপূর্ণ বিল্ট-ইন ফাংশন (Built-in Function) নিচে উল্লেখ করা হলো:

print(): এটি একটি ফাংশন যা আউটপুট ডিভাইস, সাধারণত স্ক্রিন বা কনসোলে, ডেটা প্রিন্ট করার জন্য ব্যবহৃত হয়।

* উদাহরণ: `print("Hello World")`

input(): এটি ব্যবহারকারীর কাছ থেকে ইনপুট গ্রহণ করার জন্য ব্যবহৃত হয়। এই ফাংশনটি সবসময় ইনপুটকে একটি স্ট্রিং (String) হিসেবে রিটার্ন করে।

* উদাহরণ: `name = input("Enter name: ")`

len(): এই ফাংশনটি একটি সিকোয়েন্স (Sequence) বা কালেকশনের (Collection)—যেমন: লিস্ট, টাপল, সেট বা স্ট্রিং—এর উপাদানের মোট সংখ্যা বা দৈর্ঘ্য নির্ণয় করে।

* উদাহরণ: `length = len([1, 2, 3])`

int() / str() / float() (টাইপ কনভার্সন ফাংশন): এই ফাংশনগুলো এক ডেটা টাইপকে অন্য ডেটা টাইপে রূপান্তর করতে ব্যবহৃত হয়।

* উদাহরণ: int('10') স্ট্রিং '10' কে পূর্ণসংখ্যা 10 এ রূপান্তর করে।

* উদাহরণ: str(42) সংখ্যা 42 কে স্ট্রিং '42' এ রূপান্তর করে।

রচনামূলক প্রশ্নোত্তর :

১. পাইথনে সেট অপারেশনগুলো বর্ণনা কর।

[বাকশির্বো: '২৪]

উত্তর: পাইথনে মোট পাঁচ ধরনের সেট অপারেশন রয়েছে। যথা:

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

ইউনিয়ন অপারেশন (Set Union): পাইথনে Union () ফাংশন ব্যবহার করে দুইটি সেটের উপাদানগুলোকে একত্রিত করে নতুন কোনো ইউনিয়ন সেট গঠন করা হয়। ইউনিয়ন অপারেশনে '|' চিহ্নও ব্যবহার করা হয়।

ইন্টারসেকশন অপারেশন (Intersection): পাইথনে intersection () ফাংশন ব্যবহার করে দুইটি সেটের কমন উপাদানগুলোকে নিয়ে নতুন কোনো সেট গঠন করা হয়। ইন্টারসেকশন অপারেশনে '&' চিহ্ন ও ব্যবহার করা হয়।

ডিফারেন্স অপারেশন (Difference): পাইথনে Difference () ফাংশন ব্যবহার করে দুইটি সেটের আনকমন উপাদানগুলোকে নিয়ে নতুন কোনো সেট গঠন করা হয়। তাছাড়া ডিফারেন্স অপারেশনে '-' চিহ্ন ও ব্যবহার করা হয়।

সিমেট্রিক ডিফারেন্স অপারেশন (Symmetric Difference): পাইথনে সিমেট্রিক ডিফারেন্স অপারেশনের মাধ্যমে দুইটি সেটের উপাদানগুলোর মধ্যে প্রতিটি সেটের অনুপস্থিত উপাদানগুলো নিয়ে নতুন কোনো সেট গঠন করা হয়। সিমেট্রিক ডিফারেন্স অপারেশনে '^' চিহ্ন ব্যবহার করা হয়।

মেম্বারশিপ অপারেশন (Membership operation): কোনো সেটের মধ্যে নির্দিষ্ট কোনো এলিমেন্টের উপস্থিতি বা অনুপস্থিতি চেক করার জন্য মেম্বারশিপ অপারেটর ব্যবহার করা হয়। পাইথনের সেটে ও দুই ধরনের মেম্বারশিপ অপারেটর ব্যবহৃত হয়, যেমন- 'in'। অপারেটর এবং not অপারেটর। কোনো সেটের মধ্যে নির্দিষ্ট কোনো এলিমেন্ট উপস্থিত আছে কি না সেটা চেক করার জন্য 'in' অপারেটর ব্যবহার করা হয়। যদি এলিমেন্টটি সেটের মধ্যে এক বা একাধিকবার থাকে তাহলে এটি True রিটার্ন করে, অন্যথায় False রিটার্ন করে। একইভাবে সেটের সাথে not in অপারেটর ব্যবহার করে কোনো এলিমেন্টের অনুপস্থিতি চেক করা যায়

২. পাইথনে সেটের বৈশিষ্ট্যগুলো লিখ।

[বাকশিবো: '২২']

উত্তর: পাইথনে সেট (Set) হলো একটি গুরুত্বপূর্ণ বিন্ট-ইন ডেটা স্ট্রাকচার। এর প্রধান বৈশিষ্ট্যগুলো নিচে উল্লেখ করা হলো:

অনন্য উপাদান (Unique Elements)

সেটের সবচেয়ে মৌলিক বৈশিষ্ট্য হলো এটি শুধুমাত্র অনন্য (Unique) উপাদান ধারণ করে। যদি আপনি সেটে একাধিকবার একই মান যোগ করার চেষ্টা করেন, তবে সেটি শুধুমাত্র একবারই সংরক্ষিত হবে।

অবিন্যস্ত (Unordered)

সেটের উপাদানগুলি নির্দিষ্ট কোনো ক্রমে (Order) সংরক্ষিত হয় না। ফলে আপনি ইনডেক্সিং (my_set[0]) বা স্লাইসিং ব্যবহার করে সেটের কোনো উপাদান সরাসরি অ্যাক্সেস করতে পারবেন না।

পরিবর্তনশীল (Mutable)

সেট নিজেই একটি পরিবর্তনশীল (Mutable) অবজেক্ট। এর মানে হলো আপনি সেট তৈরি করার পরেও এতে নতুন উপাদান যোগ (add) করতে বা বিদ্যমান উপাদান মুছে ফেলতে (remove/discard) পারবেন।

অপরিবর্তনশীল উপাদান গ্রহণ (Only Immutable Elements Allowed)

যদিও সেট নিজে পরিবর্তনশীল, তবে সেটের উপাদানগুলি অবশ্যই অপরিবর্তনশীল (Immutable) হতে হবে।

- গ্রহণযোগ্য: সংখ্যা, স্ট্রিং, এবং টাপল।
- অগ্রহণযোগ্য: লিস্ট এবং ডিকশনারি, কারণ তারা পরিবর্তনশীল।

গাণিতিক সেট অপারেশন

সেট সরাসরি গাণিতিক সেট অপারেশনগুলো সমর্থন করে, যেমন:

- ইউনিয়ন (Union): দুটি সেটের সব উপাদান নিয়ে নতুন সেট তৈরি।
- ইন্টারসেকশন (Intersection): দুটি সেটের সাধারণ উপাদানগুলো নিয়ে নতুন সেট তৈরি।
- ডিফারেন্স (Difference): একটি সেটে আছে কিন্তু অন্যটিতে নেই এমন উপাদান নিয়ে নতুন সেট তৈরি।

প্রতীক: সেট সাধারণত কার্লি ব্রেসেস বা দ্বিতীয় বন্ধনী ({}) ব্যবহার করে সংজ্ঞায়িত করা হয়।

অধ্যায়
১০

ডিকশনারি স্ট্রাকচার

১০.১ অধ্যায়ের প্রধান সূচিপত্র (contents)

বিষয়	পৃষ্ঠা নং
❖ সূত্রাবলী ও শর্টকাট টেকনিক	
❖ অতি সংক্ষিপ্ত প্রশ্নাবলি	
❖ সংক্ষিপ্ত প্রশ্নাবলি	
❖ রচনামূলক প্রশ্নাবলি	

বোর্ড বোর্ড প্রশ্নাবলির বিশ্লেষণ (Board Questions Analysis) প্রশ্নাবলির বিশ্লেষণ (Board Questions Analysis)

এই অধ্যায় থেকে বিগত পাঁচ বছরের ডিপ্লোমা ইন ইঞ্জিনিয়ারিং, ডিপ্লোমা ইন টেক্সটাইল, ডিপ্লোমা ইন এগ্রিকালচার, ডিপ্লোমা ইন মেরিন পরীক্ষায় মোট প্রশ্ন। নিচে ‘Board Questions Analysis’ অংশে এই অধ্যায় থেকে কোন সালে কোন বোর্ডে কতটি প্রশ্ন হয়েছে তা দেখানো হলো।

সাল \ বোর্ড	অতি সংক্ষিপ্ত	সংক্ষিপ্ত প্রশ্ন	রচনামূলক প্রশ্ন
২০২৪	১টি	—	—
২০২৩	১টি	—	—
২০২২	—	—	—
২০২১	—	—	—



“ওয়েব শুধু মেশিনকে নয়, মানুষকেও সংযুক্ত করে।”

- টিম বার্নার্স-লি (Tim Berners-Lee, ওয়েবের উদ্ভাবক)

অতি সংক্ষিপ্ত প্রশ্নোত্তর :

১. পাইথন প্রোগ্রামে ডিকশনারি কেন ব্যবহার করা হয়?

[বাকশির্বো : '১৭, '১৯, '২৩]

অথবা, ডিকশনারি বলতে কী বোঝায়?

উত্তর: ডিকশনারি এমন এক ধরনের ডাটা স্ট্রাকচার, যার মধ্যেও লিস্টের মতো বিভিন্ন রকম এলিমেন্ট বা অবজেক্ট স্টোর করা যায়। কিন্তু এক্ষেত্রে এর এলিমেন্টগুলোকে ম্যানুয়ালি ইনডেক্সিং করতে হয়। অর্থাৎ প্রোগ্রামারকেই প্রতিটি এলিমেন্টের জন্য একটি key বা ইনডেক্স নির্ধারণ করে দিতে হয়।

২. ডিকশনারিতে clear() মেথডের কাজ কী?

উত্তর: ডিকশনারিতে বিদ্যমান সকল উপাদানকে রিমুভ করার জন্য এই মেথড ব্যবহার করা হয়।

৩. ডিকশনারিতে itemes() মেথডের কাজ কী?

উত্তর: ডিকশনারি আইটেমগুলোর একটি নতুন ভিউ রিটার্ন করার জন্য এই মেথড ব্যবহার করা হয়।

৪. pop() ফাংশন কি কাজে ব্যবহার করা হয়?

[বাকশির্বো : '২৪]

উত্তর: ডিকশনারির ক্ষেত্রে pop() ফাংশনটি একটি নির্দিষ্ট কী (Key) এবং তার সংশ্লিষ্ট মান (Value) অপসারণের জন্য ব্যবহৃত হয়।

সংক্ষিপ্ত প্রশ্নোত্তর :

১. Python-এ Dictionary ব্যাখ্যা কর।

[বাকাশিবো: ১৯]

অথবা, পাইথনে ডিকশনারি 'কি' বর্ণনা কর?

উত্তর: লিস্টের সাথে এর অনেক পার্থক্য রয়েছে। উত্তর: ডিকশনারি পাইথনের সবচেয়ে স্মার্ট ডাটা টাইপ। ডিকশনারিকে আমরা লিস্টের সাথে তুলনা করতে পারি। কিন্তু লিস্টের সাথে এর অনেক পার্থক্য রয়েছে।

ডিকশনারি এমন এক ধরনের ডাটা স্ট্রাকচার যার মধ্যেও লিস্টের মত বিভিন্ন রকম এলিমেন্ট বা অবজেক্ট স্টোর করা যায়- কিন্তু এ ক্ষেত্রে ওই এলিমেন্ট গুলোকে ম্যানুয়ালি ইনডেক্স করতে হয়। অন্যভাবে বলতে গেলে, আমাদের নিজেরদেরকেই প্রত্যেকটা এলিমেন্টের বা value এর জন্য একটি key বা ইনডেক্স নির্ধারণ করে দিতে হয়। অতঃপর একটি key-value জোড় ওয়ালা এলিমেন্টের কালেকশন তৈরি হয়। দুটি কালী ব্র্যাকেট {} মধ্যে কোলন চিহ্ন দিয়ে key-value জোড় তৈরি করে এবং প্রত্যেক জোড় কে কমা, দিয়ে আলাদা করে একটি ডিকশনারি তৈরি করা যায়। নিচের মত করে। my_marks ("Bengali": 80, "English": 85, "Math": 90)

আবার ফাকা ডিকশনারি তৈরির জন্য এভাবে লিখলেই সেটি ইনিসিয়ালাইজ হয়ে যায়-

```
my_dictionary = {}
```

২. ডিকশনারি 'কী' এর বৈশিষ্ট্য লেখ।

উত্তর: নিচে ডিকশনারি 'কী'-এর বৈশিষ্ট্য উল্লেখ করা হলো-

- ডিকশনারি কী-ভ্যালু জোড় অবস্থায় থাকে কিন্তু কী-ভ্যালুগুলোর কোনো অর্ডার (Order) নেই।
- ডিকশনারির প্রতিটি 'কী' ইউনিক হতে হবে।

(iii) ডিকশনারির মধ্যে যেকোনো টাইপের অবজেক্ট বা এলিমেন্টকেই স্টোর করা যায় শুধু এর key গুলো হতে হবে Immutable (অপরিবর্তনীয়) টাইপের যেমন নিচের মত করে একটি ডিকশনারি তৈরি করা যেতে পারে।

```
my_marks ("Bengali": [30, 35, 32], "English": [45, 52, 33], "Math":
[60, 74, 58]) print(my_marks["Math"])
```

আউটপুট:

যার আউটপুট আসবে,

```
[60, 74, 58]
```

(iv) কিন্তু নিচের মত একটি ডিকশনারি হতে পারে না-

```
my_marks = {[30, 35, 32]: "Bengali", [45, 52, 33]: "English", [60, 74, 58]:
"Math"}
```

এর আউটপুট হবে,

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

TypeError: unhashable type: 'list'

অর্থাৎ একটি লিস্ট যা কিনা একধরনের Mutable টাইপ তাকে কোন ডিকশনারির key হিসেবে ব্যবহার করা যাবে না।

রচনামূলক প্রশ্নোত্তর :

১. ডিকশনারিতে ব্যবহৃত বিল্ট-ইন ফাংশন এবং মেথডের বর্ণনা কর।

উত্তর: ডিকশনারিতে ব্যবহৃত বিল্ট-ইন ফাংশন (Built-in functions): পাইথন প্রোগ্রামের ডিকশনারিতে দ্রুততার সঙ্গে এবং সুনির্দিষ্ট কোনো কাজ সম্পাদনের জন্য এতে বেশ কিছু বিল্ট-ইন ফাংশন রয়েছে। নিম্নে ডিকশনারিতে ব্যবহৃত এ সকল বিল্ট-ইন ফাংশনসমূহ ও তাদের ব্যবহার উল্লেখ করা হলো-

পাইথনের ডিকশনারিতে ব্যবহৃত বিল্ট-ইন ফাংশনসমূহ	
বিল্ট-ইন ফাংশন	বর্ণনা
all()	ডিকশনারি empty হলেই কেবল আউটপুট হিসেবে False রিটার্ন করে।
any()	ডিকশনারির সকল উপাদানের মান সত্য হলে আউটপুট হিসেবে এ ফাংশনটি True রিটার্ন করে। ডিকশনারির যে-কোনো একটি উপাদানের মান সত্য হলে ও আউটপুট হিসেবে এ ফাংশনটি True রিটার্ন করে।
len()	ডিকশনারিতে বিদ্যমান সকল উপাদানের সংখ্যা জানার জন্য এই ফাংশনটি ব্যবহৃত হয়।
comp()	দুটি ডিকশনারির আইটেমগুলোর মধ্যে তুলনা করার জন্য এই ফাংশনটি ব্যবহৃত হয়।
sorted()	ডিকশনারির জন্য একটি নতুন সর্টেড কী লিস্ট রিটার্ন করার জন্য এই ফাংশনটি ব্যবহৃত হয়।

উদাহরণ:

প্রোগ্রাম:

```
squares = {1: 1, 3: 9, 5: 25, 7: 49, 9:81}
```

```
print(len(squares))
```

```
print(sorted(squares))
```

আউটপুট:

5

[1, 3, 5, 7, 9]

ডিকশনারিতে ব্যবহৃত বিল্ট-ইন মেথডসমূহ (Set methods): ডিকশনারি ম্যানিপুলেশনের জন্য পাইথনে মেথডসমূহ ব্যবহার করা হয়। পাইথনে যে-সকল বিল্ট-ইন মেথডসমূহ ব্যবহার করা হয় তার একটি তালিকা ও তাদের বর্ণনা নিম্নে উল্লেখ করা হলো-

পাইথনের ডিকশনারিতে ব্যবহৃত মেথডসমূহ	
মেথড	বর্ণনা
popitem()	আইটেমকে রিমুভ করা ও কোনো অরবিটারি আইটেম (কী, ভ্যালু) রিটার্ন করার জন্য এই মেথড ব্যবহার করা হয়। আর ডিকশনারিটি যদি ফাঁকা হয় তাহলে কী (Key) এরর দেখাবে।
copy()	ডিকশনারির একটি কপি তৈরির জন্য এই মেথড ব্যবহার করা হয়।
values()	ডিকশনারি ভ্যালুগুলোর একটি নতুন ভিউ রিটার্ন করার জন্য এই মেথড ব্যবহার করা হয়।
fromkeys(seq[,v])	seq হতে v এর সমান কী ভ্যালুসহ নতুন একটি ডিকশনারি রিটার্ন করার জন্য এ মেথড ব্যবহার করা হয়।
get(key[,d])	কী ভ্যালু রিটার্ন করার জন্য এই মেথড ব্যবহার করা হয়। আর যদি কী অনুপস্থিত থাকে তবে d-এর মান (defaults to none) প্রদর্শন করে।
items()	ডিকশনারি আইটেমগুলোর একটি নতুন ভিউ রিটার্ন করার জন্য এই মেথড ব্যবহার করা হয়।
keys()	ডিকশনারি কী-গুলোর একটি নতুন ভিউ রিটার্ন করার জন্য এই মেথড ব্যবহার করা হয়।
pop(key[,d])	কী ভ্যালুসহ কোনো আইটেম রিমুভ করা এবং এর ভ্যালু রিটার্ন করার জন্য এই মেথড ব্যবহার করা হয়। আর যদি কী অনুপস্থিত থাকে ও ১-এর মান সরবরাহ না করা হয় তবে কী এরর প্রদর্শন করে।

setdefault(key[,d])	কী'টি যদি ডিকশনারিতে বিদ্যমান থাকে তবে এই ফাংশনের মাধ্যমে এর মান প্রদর্শন করা যায়। আর যদি কী' টি ডিকশনারিতে অনুপস্থিত থাকে তবে d-এর মানসহ কী ইনসার্ট করা এবং d-এর মান রিটার্ন করা যায়।
update([other])	অন্য কোনো ডিকশনারির (কী/ভ্যালু) pair-সহ ডিকশনারি আপডেট করার জন্য এই মেথড ব্যবহার করা হয়।
clear()	ডিকশনারিতে বিদ্যমান সকল উপাদানকে রিমুভ করার জন্য এই মেথড ব্যবহার করা হয়।

অধ্যায়
১১

ফাংশন অপারেশন

১১.১ অধ্যায়ের প্রধান সূচিপত্র (contents)

বিষয়	পৃষ্ঠা নং
❖ সূত্রাবলী ও শর্টকাট টেকনিক	
❖ অতি সংক্ষিপ্ত প্রশ্নাবলি	
❖ সংক্ষিপ্ত প্রশ্নাবলি	
❖ রচনামূলক প্রশ্নাবলি	

বোর্ড বোর্ড প্রশ্নাবলির বিশ্লেষণ (Board Questions Analysis) প্রশ্নাবলির বিশ্লেষণ (Board Questions Analysis)

এই অধ্যায় থেকে বিগত পাঁচ বছরের ডিপ্লোমা ইন ইঞ্জিনিয়ারিং, ডিপ্লোমা ইন টেক্সটাইল, ডিপ্লোমা ইন এগ্রিকালচার, ডিপ্লোমা ইন মেরিন পরীক্ষায় মোট প্রশ্ন। নিচে ‘Board Questions Analysis’ অংশে এই অধ্যায় থেকে কোন সালে কোন বোর্ডে কতটি প্রশ্ন হয়েছে তা দেখানো হলো।

সাল \ বোর্ড	অতি সংক্ষিপ্ত	সংক্ষিপ্ত প্রশ্ন	রচনামূলক প্রশ্ন
২০২৪	—	—	১টি
২০২৩	—	১টি	২টি
২০২২	১টি	১টি	১টি
২০২১	—	—	—



“আমরা সবাই এখন ইন্টারনেটের মাধ্যমে সংযুক্ত — যেন এক বিশাল মস্তিষ্কের স্নায়ু কোষ।”

- স্টিফেন হকিং (Stephen Hawking)

অতি সংক্ষিপ্ত প্রশ্নোত্তর :

১. Function Calling বলতে কী বোঝায়?

[বাকশিবো: '২১]

অথবা, Calling Function Called Function কাকে বলে?

অথবা, কলিং প্রোগ্রাম ও কন্ড প্রোগ্রাম বলতে কী বোঝায়?

উত্তর: যখন একটি ফাংশন অপর কোনো ফাংশনকে ব্যবহার করে, তখন তাকে কলিং ফাংশন বা মূল ফাংশন এবং যে ফাংশনকে কল করা হয়, তাকে ব্যবহৃত বা কন্ড ফাংশন বলা হয়।

২. পাঁচটি Library Function-এর নাম লেখ।

[বাকশিবো: '২১]

অথবা, চারটি লাইব্রেরি ফাংশনের নাম লেখ।

উত্তর: কিছু লাইব্রেরি ফাংশন হচ্ছে:

input (), print (), type (), pow (), round (), max(), min (), sum () ইত্যাদি।

৩. লাইব্রেরি ফাংশন কাকে বলে?

[বাকশিবো: '১৯, '২০]

অথবা, বিল্ট-ইন ফাংশন বলতে কী বোঝায়?

উত্তর: যে সকল ফাংশন ডেভেলপার কর্তৃক পূর্ব থেকেই নির্ধারিত কাজের জন্য ডিফাইনকৃত, কম্পাইলারের আওতাভুক্ত এবং নির্দিষ্ট নামসম্পন্ন, ব্যবহারকারী চাইলেও যাদের নাম পরিবর্তন করতে পারে না সে সকল ফাংশনকে লাইব্রেরি ফাংশন বলে।

৪. ফাংশন Prototype এর কাজগুলো লেখ।

[বাকশিবো: '১৯]

উত্তর: ফাংশন Prototype এর কাজ হলো কোনো ফাংশন ডিক্লেয়ার করার পূর্বে ঐ ফাংশনের ডাটা কী ধরনের হবে ও কয়টি ডাটা হবে এবং ব্যবহারকারী ফাংশনে কোনো ধরনের ডাটা পাঠাবে তা ব্যবহারকারীকে জানিয়ে দেয়া।

৫. ফাংশন আরগুমেন্ট বলতে কী বোঝায়?

[বাকশির্বো: '১৭]

উত্তর: যখন একটি ফাংশন কল করা হয় তখন ঐ ফাংশন এর প্যারামিটারের টাইপ, অর্ডার এবং নাম্বার অনুসারে ভ্যালু পাস করা হয়। এই ভ্যালুকেই ফাংশন এর আরগুমেন্ট বলা হয়।

৬. ফাংশন প্রটোটাইপ বলতে কী বোঝায়?

[বাকশির্বো: '১৯]

অথবা, প্রটোটাইপ বলতে কী বোঝায়?

উত্তর: ফাংশন প্রটোটাইপ হলো- কোনো নির্দিষ্ট ফাংশনের সংক্ষিপ্ত বর্ণনা (ফাংশন কী ধরনের ও কয়টি ডাটা গ্রহণ করবে এবং ব্যবহারকারী ফাংশনে কোনো ধরনের ডাটা পাঠাবে) কম্পাইলারকে জানানোর একটি process যাতে compiler বুঝে নিতে পারে যে, ঐ নামের ও গঠনের একটি ফাংশন পরে প্রোগ্রামের কোথাও define করা হয়েছে।

৭. ফাংশন বলতে কী বোঝায়?

[বাকশির্বো: '২২]

অথবা, ফাংশন কী?

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

৮. ফাংশন হেডার কাকে বলে?

উত্তর: প্রতিটি ফাংশনের একটি নির্দিষ্ট নাম থাকে, একে ফাংশন হেডার বলে।

৯. ফাংশন কত প্রকার ও কী কী?

উত্তর: ফাংশন দুই প্রকার, যথা-

(1) User define function

(ii) Library function.

১০. User defined function কাকে বলে?

উত্তর: প্রোগ্রামে ব্যবহারকারী তার নিজস্ব প্রয়োজন এবং প্রজ্ঞা অনুযায়ী যে-সব ফাংশন ডিফাইন করেন এবং প্রয়োজনে সেগুলো প্রোগ্রামে ব্যবহার করেন, সেগুলোকে ইউজার-ডিফাইন্ড বা ব্যবহারকারী বর্ণিত ফাংশন বলা হয়।

১১. Library function ব্যবহারের সুবিধা কী?

অথবা, Library function ব্যবহারের উদ্দেশ্য কী, লেখ।

উত্তর: লাইব্রেরি ফাংশন ব্যবহারের সুবিধা এই যে, লাইব্রেরি ফাংশন ব্যবহার করার পর উক্ত ফাংশনের কাজ আপনা-আপনি সম্পন্ন হয়। এর জন্য বিস্তারিত প্রোগ্রাম লেখার প্রয়োজন হয় না। কারণ লাইব্রেরি ফাংশনগুলোর প্রটোটাইপ তাদের সংশ্লিষ্ট হেডার ফাইলে বর্ণিত থাকে।

১২. ফাংশন চেনার উপায় কী?

উত্তর: ফাংশন চেনার সহজ উপায় হলো ফাংশনের নামের শেষে একজোড়া প্যারেনথেসিস '()' থাকে। এই প্যারেনথেসিস্টে মধ্যে প্যারামিটার বা আরগুমেন্ট থাকতে পারে বা না ও পারে।

১৩. Recursive function কী?

অথবা, Recursive ফাংশন বলতে কী বোঝায়? উদাহরণ দাও।

অথবা, পিরকার্শ ফাংশন বলতে কী বোঝায়?

উত্তর: প্রোগ্রামে কোনো function নিজেই নিজেকে call করতে পারে। কোনো function নিজেই নিজেকে যখন call করে তখন সেই function কে recursive function বলে।

১৪. পাইথনে ফাংশন তৈরির সিনট্যাক্স লেখ।

উত্তর: পাইথনে ফাংশন তৈরির সাধারণ গঠন বা সিনট্যাক্স বা ফরম্যাট নিম্নরূপ:


```
def Function Name (Parameters):
```

```
    Statemnts
```

```
    return [expression]
```

১৫. লোকাল ভেরিয়েবল ও গ্লোবাল ভেরিয়েবল বলতে কী বোঝায়?

অথবা, লোকাল ও গ্লোবাল ভেরিয়েবল বলতে কী বোঝায়?

অথবা, লোকাল ভেরিয়েবল ও গ্লোবাল ভেরিয়েবল কী?

অথবা, Local ও Global variable কী?

উত্তর:

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

গ্লোবাল ভেরিয়েবল: যখন কোনো ভেরিয়েবলের কার্যকারিতা কোনো নির্দিষ্ট ফাংশনের মধ্যে সীমাবদ্ধ না থেকে এর মান সনফাংশনেই ব্যবহার করা যায়, তাকে গ্লোবাল ভেরিয়েবল বলে।

১৬. ল্যাম্বডা ফাংশন কী?

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

সংক্ষিপ্ত প্রশ্নোত্তর :

১. ফাংশন ব্যবহার করে বৃত্তের ক্ষেত্রফল নির্ণয়ের একটি প্রোগ্রাম তৈরি কর। [বাকশিবো: ১৭]

উত্তর: ফাংশন ব্যবহার করে ত্রিভুজ ক্ষেত্রের ক্ষেত্রফল নির্ণয় করার প্রোগ্রাম লেখ।

```
import math
```

```
def Triangle():
```

```
    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 of the triangle is: ", Area)
```

```
    else:
```

```
        Print ("The triangle is not possible")
```

```
Triangle ()
```

২. ইউজার ডিফাইন্ড ফাংশন ব্যবহার সুবিধাসমূহ লেখ।

উত্তর: পাইথনে ইউজার ডিফাইন্ড ফাংশন ব্যবহারের নানাবিধ সুবিধা রয়েছে। যেমন-

(i) পাইথনে ইউজার ডিফাইন্ড ফাংশন ব্যবহার করে একটি বড় প্রোগ্রামকে ছোট ছোট স্বাধীন প্রোগ্রামে রূপান্তর করা। বলে প্রোগ্রামটি অধিক সহজতর ও আকর্ষণীয় বলে প্রতীয়মান হয়।

- (ii) একটি প্রোগ্রামে ব্যবহৃত ফাংশনকে সহজেই অন্য প্রোগ্রামেও ব্যবহার করা যায়। ফলে উক্ত ফাংশন বডিকে বায়া রিপট করার প্রয়োজন হয় না।
- (iii) উপযুক্ত স্থানে ফাংশনটি ব্যবহার করতে পারলে প্রোগ্রামের আকার ছোট হয়ে যায়।
- (iv) প্রোগ্রাম এক্সিকিউশনে সময় কম লাগে।
- (v) প্রোগ্রামের ভুল-ত্রুটি নির্ণয় ও সংশোধন সহজতর হয়।
- vi) রিকার্সন পদ্ধতি একটি ফাংশন প্রয়োজন নিজে নিজেই পুনঃ পুনঃ এক্সেল করতে পারে।

৩. ফাংশন ব্যবহারের সুবিধাসমূহ লেখ।

[বাকাশিবো: '১৮]

উত্তর: নিচে ফাংশন ব্যবহারের সুবিধাসমূহ দেওয়া হলো-

- ফাংশন ব্যবহার করার ফলে প্রোগ্রামের কোনো অংশ মূল প্রোগ্রামে বারবার লেখার প্রয়োজন হয় না।
- প্রোগ্রামের সাইজ ছোট হয়।
- প্রোগ্রাম লিখতে সময় অপেক্ষাকৃত কম লাগে।
- প্রোগ্রাম পড়ে বুঝা ও সংশোধন করা সহজ হয়।
- বড় প্রোগ্রাম অনেকগুলো মডিউলে ভাগ করে নিয়ে এক-এক মডিউল আলাদাভাবে বিভিন্ন প্রোগ্রামার দিয়ে প্রেমা করানো হলে এবং এগুলো ফাংশন হিসেবে ব্যবহার করা হলে স্বল্পতম সময়ে প্রোগ্রাম সম্পন্ন করা সম্ভব হয়।
- অত্যন্ত জরুরি প্রোগ্রামকে ফাংশন হিসেবে লাইব্রেরিতে জমা করে রাখলে অন্য ব্যবহারকারী প্রয়োজনে তা ব্যবহার করতে পারে।

8. $2+4+6+ \dots + 100$ সিরিজের যোগফল বের করার Program লেখ।

[বাকাশিবো: '২০,১৭, '২১]

অথবা, জোড় সংখ্যার সিরিজের যোগফল নির্ণয়ের প্রোগ্রাম লেখ।

উত্তর:

#Python Program to Calculate Sum of Even Numbers from 1 to 100

```
minimum = int(input(" Please Enter the Minimum Value: "))
```

```
maximum = int(input(" Please Enter the Maximum Value: "))
```

```
total = 0
```

```
for number in range(minimum, maximum+1);
```

```
    if(number % 2 == 0):
```

```
        print("{0}".format(number))
```

```
        total = total + number
```

```
print("The Sum of Even Numbers from {0} to {1} = {2}".format(minimum,  
number, total))
```

৫. আয়তক্ষেত্রের ক্ষেত্রফল বের করার Program লেখ।

[বাকশিৰো: '২০]

উত্তর:

```
def Area():
```

```
    length=int(input("Enter the value of Length:"))
```

```
    width=int(input("Enter the value of Width:"))
```

```
    RectangleArea = length*width
```

```
    print("Area of Rectangle is", RectangleArea)
```

```
Area()
```

৬. পাস বাই রেফারেন্স ও পাস বাই ভ্যালু বর্ণনা কর।

[বাকশিবা: '২০]

উত্তর:

পাস বাই ভ্যালু (Pass by Value)

- যখন কোনো ফাংশনে একটি ভেরিয়েবল পাস করা হয়, তখন সেই ভেরিয়েবলের কপি পাঠানো হয়।
- ফাংশনের ভিতরে পরিবর্তন করলে মূল ভেরিয়েবলের মানে কোনো প্রভাব পড়ে না।
- অর্থাৎ, ফাংশনের ভিতরে পরিবর্তন লোকাল বা স্থানীয়।

পাস বাই রেফারেন্স (Pass by Reference)

- যখন কোনো ফাংশনে ভেরিয়েবল পাস করা হয়, তখন সেই ভেরিয়েবলের ঠিকানা (reference) পাঠানো হয়।
- ফলে ফাংশনের ভিতরে পরিবর্তন করলে মূল ভেরিয়েবলের মানও পরিবর্তিত হয়।
- অর্থাৎ, ফাংশনের ভিতরের পরিবর্তন মূল ডেটাতে সিঙ্গেল প্রভাব ফেলে।

৭. পাইথনে ফাংশন তৈরির ক্ষেত্রে কী কী নিয়ম মেনে চলতে হয় লেখ।

উত্তর: পাইথনে ফাংশন তৈরি বা ফাংশন ডিফাইন করার ক্ষেত্রে সুনির্দিষ্ট কিছু নিয়মকানুন মেনে চলতে হয়। যেমন-

- ফাংশন ব্লক অবশ্যই def কীওয়ার্ড দিয়ে শুরু হবে। def এর পার একটা স্পেস দিয়ে ফাংশনের নাম থাকবে। ফাংশনের নামকরণের ক্ষেত্রে আইডেন্টিফায়ার নামকরণের নিয়মাবলি মানতে হবে। তবে ফাংশনের নাম আন্ডারস্কোর দিয়ে শুরু করা যেতে পারে।

- ফাংশনের নামের পর প্যারেনথেসিস '()' থাকবে। প্যারেনথেসিসেস ভিতর এক বা একাধিক প্যারামিটার (parameter) বা আরগুমেন্ট (argument) (argume কমা দিয়ে ব্যবহার করা যাবে। প্যারামিটার-(parameter) বা আরগুমেন্ট (argument) হচ্ছে ফাংশন দিয়ে কী কী ডাটা পাস করবে। ফাংশনে এক বা একাধিক parameter পাস করা যেতে পারে। আবার কোনো কোনো ফাংশনে কোনো Parameter নাও থাকতে পারে। এটা নির্ভর করে কী ধরনের ফাংশন লেখা হচ্ছে তার উপর। একের অধিক Parameter থাকলে তাদেরকে কমা দিয়ে দিয়ে লেখতে হয়।
- প্যারেনথেসিসের পর কোলন (:) চিহ্ন থাকবে।
- ফাংশনের সব স্টেটমেন্ট ইনডেন্টেড থাকবে। প্রথম স্টেটমেন্ট কমেন্ট ব্যবহার করলে ভালো হয় যাতে ফাংশনের একটা সংক্ষিপ্ত পরিচয় থাকবে। তবে এটা বাধ্যতামূলক নয়।

সাধারণত return কীওয়ার্ড ব্যবহার করে ফাংশনের এক্সিকিউশন শেষ করা হয়। কোনো ডাটা রিটার্ন করার ক্ষেত্রে তার কীওয়ার্ডের পর একটা স্পেস দিতে হবে। বস্তুত কাজ শেষে ফাংশনটি কী রিটার্ন করবে তাই return দিয়ে পাস করা হয়।

ফাংশন যদি কোনো কিছু রিটার্ন না করে তাহলে return 0 বা না return লিখলেও হবে।

৮. Local এবং global variable এর মাঝে পার্থক্য লেখ।

[বাকশির্বো: '২২]

উত্তর:

i. স্কোপ (Scope): * **লোকাল ভেরিয়েবল:** এর স্কোপ সীমিত। এটি কেবল তার নিজস্ব ফাংশন বা ব্লকের মধ্যে অ্যাক্সেসযোগ্য। ফাংশনের বাইরে এই ভেরিয়েবল ব্যবহার করার চেষ্টা করলে এরর (Error) দেখাবে। * **গ্লোবাল ভেরিয়েবল:** এর স্কোপ গ্লোবাল বা বৈশ্বিক। এটি প্রোগ্রামের যেকোনো ফাংশন থেকে অ্যাক্সেস এবং পরিবর্তন করা যেতে পারে।

ii. **জীবনকাল (Lifetime):** * **লোকাল ভেরিয়েবল:** এটি যখন ফাংশন কল হয় বা ব্লক শুরু হয়, তখন অস্তিত্ব লাভ করে এবং ফাংশন বা ব্লকটি শেষ হলেই এর অস্তিত্ব শেষ হয়ে যায় (মেমরি থেকে

মুছে যায়)। * **গ্লোবাল ভেরিয়েবল:** প্রোগ্রাম এক্সিকিউশন (execution) শুরু হওয়ার সময় এটি তৈরি হয় এবং প্রোগ্রামটি সম্পূর্ণভাবে শেষ না হওয়া পর্যন্ত এটি মেমরিতে বিদ্যমান থাকে।

iii. **নিরাপত্তা ও রক্ষণাবেক্ষণ (Security & Maintenance):** * **লোকাল ভেরিয়েবল:** কোডের নিরাপত্তা বেশি, কারণ কোনো ফাংশন ভুল করে অন্য ফাংশনের ভেরিয়েবল পরিবর্তন করতে পারে না। এটি কোড রক্ষণাবেক্ষণ এবং ডিবাগিং (Debugging) সহজ করে। * **গ্লোবাল ভেরিয়েবল:** এটি যেকোনো জায়গা থেকে পরিবর্তিত হতে পারে, যা অপ্রত্যাশিত ফলাফল তৈরি করতে পারে এবং কোড ডিবাগিং জটিল করে তোলে। তাই এদের ব্যবহার সতর্কতার সাথে করা উচিত

রচনামূলক প্রশ্নোত্তর :

১. ফাংশন ব্যবহার করে ত্রিভুজক্ষেত্রের ক্ষেত্রফল নির্ণয়ের প্রোগ্রাম লেখ।

[বাকশির্বা: '১৭, '২১, '২২]

উত্তর:

```
import math
```

```
def area(a, b, c):
```

```
    if a+b<=c or a+c<=b or b+c<=a:
```

```
        return 0
```

```
    s = (a+b+c)/2
```

```
    return math.sqrt(s*(s-a)*(s-b)*(s-c))
```

```
a, b, c = map(float, input("তিন বাহুর মান দিন: ").split())
```

```
ar = area(a, b, c)
```

```
print("ত্রিভুজ সম্ভব না" if ar==0 else f"ক্ষেত্রফল: {ar:.2f}")
```

২. ১ থেকে ১০০ পর্যন্ত মৌলিক সংখ্যা বের করার Program লেখ।

[বাকশির্বো: ২০R]

উত্তর:

```
for num in range(2, 101):  
  
    is_prime = True  
  
    for i in range(2, int(num**0.5) + 1):  
  
        if num % i == 0:  
  
            is_prime = False  
  
            break  
  
if is_prime:  
  
    print(num, end=' ')
```

ব্যাখ্যা:

- ২ থেকে ১০০ পর্যন্ত প্রতিটি সংখ্যার জন্য চেক করা হয় সে সংখ্যা মৌলিক কি না।
- মৌলিক সংখ্যা হলো এমন সংখ্যা যা ১ এবং নিজ ছাড়া অন্য কোন সংখ্যা দ্বারা বিভাজ্য নয়।
- ভাগ করার জন্য শুধু $\sqrt{\text{num}}$ পর্যন্ত পরীক্ষা করলে পারফর্মেন্স ভালো হয়।

৩. ফাংশন ব্যবহার করে কোনো সংখ্যার ফ্যাক্টোরিয়াল মান নির্ণয়ের প্রোগ্রাম লেখ।

[বাকশিৰো: '১৯, '২২, '২৩]

উত্তর:

```
def factorial(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        return n * factorial(n - 1)  
  
num = int(input("সংখ্যা দিন: "))  
print(f"{num} এর ফ্যাক্টোরিয়াল: {factorial(num)}")
```

৪. ফাংশন ব্যবহার করে কোনো সংখ্যা মৌলিক কি না তা নির্ণয়ের প্রোগ্রাম লেখ।

[বাকশিৰো: '১৯R, '১৯, '২০, '২০R]

উত্তর:

```
def is_prime(n):  
    if n <= 1:  
        return False  
    for i in range(2, int(n**0.5) + 1):  
        if n % i == 0:
```

```
return False
```

```
return True
```

```
num = int(input("সংখ্যা দিন: "))
```

```
if is_prime(num):
```

```
    print(f"{num} একটি মৌলিক সংখ্যা।")
```

```
else:
```

```
    print(f"{num} মৌলিক সংখ্যা নয়।")
```

ব্যাখ্যা:

- is_prime ফাংশনটি ২ থেকে \sqrt{n} পর্যন্ত ভাগ করে চেক করে।
- যদি কোন সংখ্যায় বিভাজ্যতা পাওয়া যায়, তাহলে False রিটার্ন করে।
- না হলে True রিটার্ন করে।

৫. Function ব্যবহার করে তিনটি সংখ্যার মাঝে বড় সংখ্যা নির্ণয়ের অ্যালগরিদম, ফ্লোচার্ট ও প্রোগ্রাম লেখ।

[বাকশিবো: '১৯]

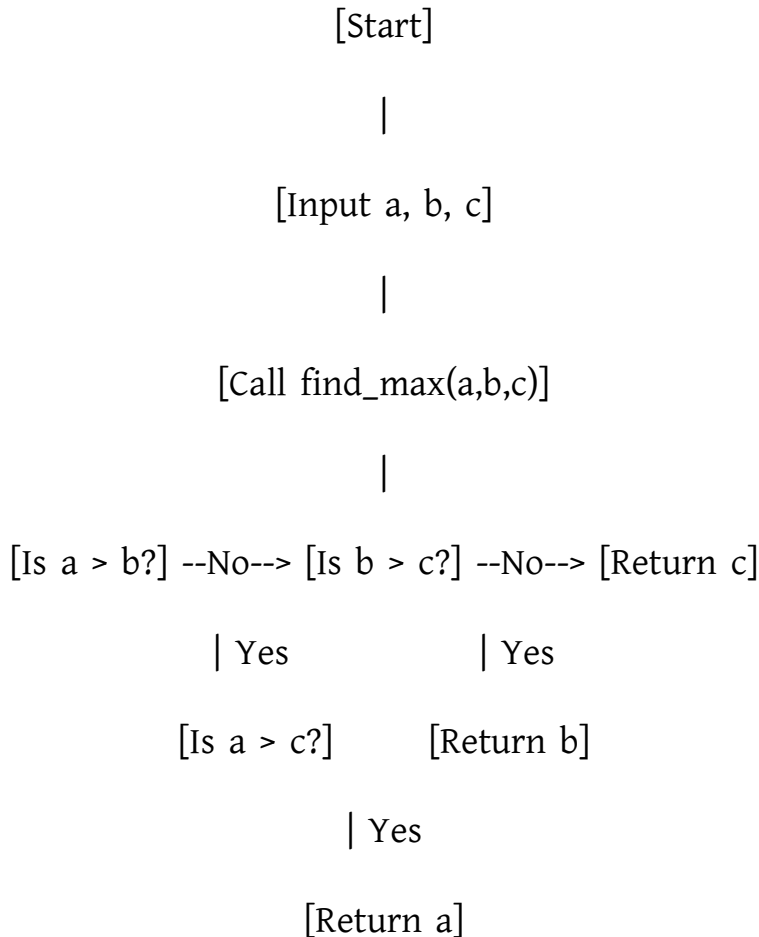
উত্তর:

অ্যালগরিদম (Algorithm)

1. শুরু
2. তিনটি সংখ্যা ইনপুট নাও: a, b, c

3. একটি ফাংশন তৈরি করো: `find_max(a, b, c)`
4. ফাংশনের ভিতরে:
 - যদি $a > b$ এবং $a > c$ তাহলে a রিটার্ন করো
 - নাহলে যদি $b > c$ তাহলে b রিটার্ন করো
 - অন্যথায় c রিটার্ন করো
5. ফাংশন কল করো এবং ফলাফল প্রিন্ট করো
6. শেষ

ফ্লোচার্ট (Flowchart) :



|
[Print max]

|
[End]

প্রোগ্রাম (Python code) :

```
def find_max(a, b, c):  
    if a > b and a > c:  
        return a  
    elif b > c:  
        return b  
    else:  
        return c  
  
a = float(input("প্রথম সংখ্যা: "))  
b = float(input("দ্বিতীয় সংখ্যা: "))  
c = float(input("তৃতীয় সংখ্যা: "))  
max_num = find_max(a, b, c)  
print(f"সবচেয়ে বড় সংখ্যা: {max_num}")
```

৫. ফাংশন ব্যবহার করে 1 হতে 100 পর্যন্ত বিজোড় সংখ্যাগুলোর যোগফল নির্ণয়ের একটি প্রোগ্রাম তৈরি কর। [বাকশিবো: '১৭]

উত্তর:

```
def sum_odd_numbers(start, end):  
  
    total = 0  
  
    for num in range(start, end + 1):  
  
        if num % 2 != 0:  
  
            total += num  
  
    return total  
  
result = sum_odd_numbers(1, 100)  
  
print(f"১ থেকে ১০০ পর্যন্ত বিজোড় সংখ্যাগুলোর যোগফল: {result}")
```

ব্যাখ্যা:

- sum_odd_numbers ফাংশন একটি রেঞ্জ নেয় এবং সেই রেঞ্জের বিজোড় সংখ্যা যোগ করে।
- তারপর রিটার্ন করে মোট যোগফল।

৬. ফাংশন ব্যবহার করে বৃত্তের ক্ষেত্রফল নির্ণয়ের একটি প্রোগ্রাম তৈরি কর। [বাকশিবো: '১৭]

উত্তর:

```
import math

def circle_area(radius):

    return math.pi * radius * radius

r = float(input("বৃত্তের ব্যাসার্ধ দিন: "))

area = circle_area(r)

print(f"বৃত্তের ক্ষেত্রফল: {area:.2f}")
```

ব্যাখ্যা:

- circle_area ফাংশন ব্যাসার্ধ নিয়ে ক্ষেত্রফল হিসাব করে:

$$\text{ক্ষেত্রফল} = \pi r^2$$

- math.pi ব্যবহার করে পাই মান নেওয়া হয়েছে।

অধ্যায়
১২

ফাইল আই/ও আপারেশন

এই অধ্যায়ের প্রধান সূচিপত্র (contents)

বিষয়	পৃষ্ঠা নং
❖ সূত্রাবলী ও শর্টকাট টেকনিক	
❖ অতি সংক্ষিপ্ত প্রশ্নাবলি	
❖ সংক্ষিপ্ত প্রশ্নাবলি	
❖ রচনামূলক প্রশ্নাবলি	

বোর্ড বোর্ড প্রশ্নাবলির বিশ্লেষণ (Board Questions Analysis) প্রশ্নাবলির বিশ্লেষণ (Board Questions Analysis)

এই অধ্যায় থেকে বিগত পাঁচ বছরের ডিপ্লোমা ইন ইঞ্জিনিয়ারিং, ডিপ্লোমা ইন টেক্সটাইল, ডিপ্লোমা ইন এগ্রিকালচার, ডিপ্লোমা ইন মেরিন পরীক্ষায় মোট প্রশ্ন। নিচে ‘Board Questions Analysis’ অংশে এই অধ্যায় থেকে কোন সালে কোন বোর্ডে কতটি প্রশ্ন হয়েছে তা দেখানো হলো।

সাল \ বোর্ড	অতি সংক্ষিপ্ত	সংক্ষিপ্ত প্রশ্ন	রচনামূলক প্রশ্ন
২০২৪	—	১টি	১টি
২০২৩	১টি	—	—
২০২২	—	১টি	১টি
২০২১	—	—	—



“প্রোগ্রামিংয়ে কঠিন কাজ সমস্যার সমাধান নয়, বরং কোন সমস্যার সমাধান করা উচিত তা নির্ধারণ করা।”

- জন কারম্যাক (John Carmack)

অতি সংক্ষিপ্ত প্রশ্নোত্তর :

১. File closing function-এর Syntax লেখ।

[বাকশিৰো: '২১]

অথবা, close() ফাংশনের সিনট্যাক্স লেখ।

উত্তর: close () ফাংশনের ফরম্যাট হলো:

FileNaem.close()

২. ফাইল কী?

[বাকশিৰো: '১৮]

উত্তর: ফাইল বা ডাটা ফাইল হচ্ছে স্টোরেজ ডিভাইসের (Hard disk, Floppy disk, Compact disk, Digital video disk ইত্যাদি) এমন একটি স্পেস (space), যেখানে স্থায়ীভাবে তথ্য সংরক্ষণ করা যায় এবং প্রয়োজনে তথ্যসমূহ রিড (read) করা যায়।

৩. ফাইল অপারেশন বলতে কী বোঝায়?

উত্তর: ফাইল অপারেশন বলতে কোনো ফাইলে ইনপুট-আউটপুট কার্যাদি (ডাটা পড়া, লেখা, পরিবর্তন করা, পরিমার্জন করা ইত্যাদি) সম্পাদনকে বুঝায়।

৪. পাইথনের বেসিক ইনপুট আউটপুট ফাংশনগুলো কী কী?

উত্তর: পাইথনের বেসিক ইনপুট আউটপুট ফাংশনগুলো হলো:

(i) input() ফাংশন ও

(ii) print () ফাংশন।

৫. print () ফাংশন এর ব্যবহার লেখ।

উত্তর: স্ক্রিনের মধ্যে ডাটা আউটপুট জন্য পাইথনে print() ফাংশন ব্যবহার করা হয়।

৬. পাইথনে প্রোগ্রামে কী কী উপায়ে মান ইনপুট নেয়া হয়?

উত্তর: পাইথনে প্রোগ্রামিং এর ক্ষেত্রে প্রোগ্রামে দুইভাবে মান ইনপুট নেয়া যায়, যথা-

(i) সরাসরি (Directly)

(ii) ইউজার হতে (From Keyboard) ।

৭. ফাইল কত প্রকার ও কী কী?

উত্তর: ফাইল প্রধানত দুই প্রকার, যথা-

- স্ট্রিম অরিয়েন্টেড বা স্ট্যান্ডার্ড ডাটা ফাইল (Stream oriented or standard data file) ।
- সিস্টেম অরিয়েন্টেড বা লো লেভেল ডাটা ফাইল (System oriented or low level data file) ।

৮. ফাইল অপারেশন বলতে কী বোঝায়?

উত্তর: কোনো ফাইলে ইনপুট-আউপুট কার্যাদি (ডাটা পড়া, লেখা, পরিবর্তন করা, পরিমার্জন করা ইত্যাদি) সম্পাদনের জন্য যে-সব অপারেশন সম্পন্ন করা হয়, তাদেরকে ফাইল অপারেশন (File operation) বলে। যেমন-

- ফাইলের নামকরণ করা
- ফাইল খোলা
- ফাইল থেকে ডাটা পড়া
- ফাইল ডাটা লেখা
- ফাইল বন্ধ করা ইত্যাদি।

৯. readline () ফাংশনের কাজ কী?

[বাকশির্বো: '২৩]

উত্তর: ফাইলের কন্টেন্ট লাইন বাই লাইন আকারে পড়ার জন্য readline () ফাংশন ব্যবহার করা হয়।

সংক্ষিপ্ত প্রশ্নোত্তর :

১. একটি ফাইলে ডাটা স্টোর করার প্রোগ্রাম তৈরি কর।

[বাকশির্বো : ১৭]

উত্তর: নিচে একটি ফাইলে ডাটা স্টোর করার প্রোগ্রাম তৈরি করা হলো-

EXAMPLE.PY

Beginning text.

```
python_file, open("example.py", "w")
```

```
python_file.write('print("Hello World!")')
```

```
python_file.close()
```

EXAMPLE.PY

```
print("Hello World!")
```

২. Python file এ ব্যবহৃত যেকোনো চারটি এক্সেস মোডের নাম ও ব্যবহার উল্লেখ করো।

[বাকশির্বো : '২২, '২৩]

উত্তর:

অ্যাক্সেস মোড	নাম ও প্রতীক	ফাইল পয়েন্টারের অবস্থান	ব্যবহারের উদ্দেশ্য
1	পড়ার মোড (<code>'r'</code>)	ফাইলের শুরুতে।	ফাইল থেকে ডেটা শুধুমাত্র পড়ার (Read) জন্য ব্যবহৃত হয়। এই মোডে ফাইলে লেখা যায় না। যদি ফাইলটি বিদ্যমান না থাকে, তবে এরর (<code>FileNotFoundException</code>) দেখায়।
2	লেখার মোড (<code>'w'</code>)	ফাইলের শুরুতে।	ফাইলটিতে লেখার (Write) জন্য ব্যবহৃত হয়। যদি ফাইলটি আগে থেকে থাকে, তবে এর ভেতরের পুরো কন্টেন্ট মুছে ফেলে নতুন করে লেখা শুরু করে। ফাইল না থাকলে, নতুন ফাইল তৈরি করে।
3	অ্যাপেন্ড মোড (<code>'a'</code>)	ফাইলের শেষে।	ফাইলটিতে নতুন ডেটা যোগ (Append) করার জন্য ব্যবহৃত হয়। এটি বিদ্যমান ডেটা মুছে ফেলে না, বরং ডেটা ফাইলের শেষে যোগ করে। ফাইল না থাকলে, নতুন ফাইল তৈরি করে।
4	পড়া এবং লেখা (<code>'r+'</code>)	ফাইলের শুরুতে।	একই সাথে ফাইল থেকে পড়ার এবং লেখার উভয় কাজের জন্য ব্যবহৃত হয়। এটি ফাইলটিকে মুছে ফেলে না, বরং ফাইল পয়েন্টার শুরুতেই থাকে, ফলে পড়া বা লেখা শুরু থেকে করা যায়।

রচনামূলক প্রশ্নোত্তর :

১. ওপেনিং ও ক্লোসিং ফাইল বর্ণনা কর।

[বাকশিবো: '২২,২৩]

অথবা, পাইথন ভাষায় ফাইল খোলা ও বন্ধ করার পদ্ধতি বর্ণনা কর।

[বাকশিবো: '১৯]

উত্তর: **open() ফাংশন:** পাইথনে ফাইল ওপেন করার জন্য open() ফাংশন ব্যবহার করা হয়।

open() ফাংশনের গঠন বা সিনট্যাক্স বা ফরম্যাট নিম্নরূপ-

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

open() ফাংশনে তিনটি প্যারামিটার আছে; যথা-

- ফাইলের নাম
- অ্যাক্সেস মোড এবং
- বাফারিং।

ফাইলের নাম: প্রথম প্যারামিটার হলো ফাইলের নাম। যদি পাইথন স্ক্রিপ্ট ও ফাইলটি যদি কম্পিউটারের একই ডিরেক্টরিতে হয়, তবে শুধু ফাইলের নামটা স্ট্রিং হিসাবে দিতে হয়। যেমন-

file object = open ("file_name.txt")

আর ফাইলটি যদি আলাদা ডিরেক্টরিতে থাকে তবে পুরো পথটি স্ট্রিং হিসাবে দিতে হবে। যেমন-

file object = open ("C:\Users\Mahbub CPI\Desktop\66631\Python Link..txt")

বাফারিং: পাইথনে open () ফাংশনের অন্যতম একটি প্যারামিটার হলো বাফারিং। ফাইল অ্যাক্সেসিং-এর সময় যদি বাফারিং ভ্যালু ০ হয় তবে পাইথনে কোনো বাফারিং সংঘটিত হয় না। তবে যদি বাফারিং ভ্যালু ১ হয় তবে পাইথনে বাফারিং সংঘটিত হয়। আবার বাফারিং ভ্যালু যদি ১-এর চেয়ে বেশি হয় তবে নির্ধারিত বাফার সাইজের মধ্যে বাফারিং সংঘটিত হয়। আর যদি বাফারিং ভ্যালু নেগেটিভ হয় তাহলে বাফার সাইজ হবে ডিফল্ট সাইজ।

অ্যাক্সেস মোড: দ্বিতীয় প্যারামিটার হলো অ্যাক্সেস মোড। পাইথন প্রোগ্রামে ফাইল নতুন করে তৈরি করা হবে, নাকি পূর্বে তৈরি করা কোনো ফাইল পরিবর্তন বা পরিবর্তনের জন্য খোলা হবে, নাকি শুধু পড়ার জন্য ফাইল খোলা হবে তা ফাইল মোডের উপর নির্ভর করে।

```
1 #Example Learn With Hafiz Vaiya
2 file = open ("my_file.txt","w")
3 file.write("Welcome To Python Programming")
4 file.close()
```

উপরোক্ত প্রোগ্রামে 'file.txt' ফাইলটি ডিফল্টভাবে রিডিং ওপেন হয়েছে। কারণ ওপেন মোড ব্যবহার করলে ফাইল ডিফল্টভাবে। রিডিং মোডে ওপেন হয়। আবার উক্ত ফাইলে রাইটিং মোড ওপেন করার জন্য 'w' ব্যবহার করা হয়েছে। পরবর্তীতে write() ফাংশন ব্যবহার করে 'Welcome to Python Programming Language' স্ট্রিংটি ফাইলে রাইট করা হয়েছে। এবার নিচের প্রোগ্রামটি লিখে রান করলে ফাইলে পূর্বে রাইটকৃত 'Welcome to Python Programming Language' স্ট্রিংটি রিড মোডের কারণে আউটপুটে প্রদর্শিত হবে।

```
1 #Example Learn With Hafiz Vaiya
2 file = open ("my_file.txt","r")
3 x = file.read()
4 print(x)
5 file.close()
```

ফাইল বন্ধ করা (Closing a file): ফাইল open করে ফাইলে read/write করার পর ফাইল অবশ্যই close (বন্ধ) করতে হবে। না হলে অকারণেই পাইথনের কাছে ফাইলটি ওপেন অবস্থাতে থাকবে, যা বস্তুত মেমরি দখল করে রাখবে এবং প্রোগ্রামের পারফরমেন্সে খারাপ ভূমিকা রাখবে। ফাইল close করার জন্য পাইথনে close () ফাংশন ব্যবহৃত হয়। close () ফাংশনের ফরম্যাট হলো-

FileName.close()

close() ফাংশন ব্যবহার করার বেশ কয়েকটি সুবিধা রয়েছে, যথা-

File-এ কিছু লিখলে তা মূলত disk-এ লেখা হয়। তবে ফাইলে কিছু লিখলেই তা সাথে সাথে disk-এ লেখা হয় না, বরং তা buffer-এ সংরক্ষিত থাকে এবং buffer পূর্ণ হলে বা close() ফাংশন ব্যবহার করলে ঐ data disk-এ লেখা হয়, ফলে ফাইলের যথাযথ ব্যবহার নিশ্চিত হয়।

> close() ব্যবহার করলে প্রোগ্রামের readability বৃদ্ধি পায়।

উদাহরণ:

```
my_file = open('test.txt', 'w')
my_file.write('Welcome to Python Programming Language')
my_file.close()
```

২. open() ফাংশনের প্যারামিটারসমূহ বর্ণনা কর।

উত্তর:

open() ফাংশন: পাইথনে ফাইল ওপেন করার জন্য open() ফাংশন ব্যবহার করা হয়। open() ফাংশনের গঠন বা সিনট্যাক্স বা ফরম্যাট নিম্নরূপ-

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

open() ফাংশনে তিনটি প্যারামিটার আছে; যথা-

- ফাইলের নাম
- অ্যাক্সেস মোড এবং
- বাফারিং।

ফাইলের নাম: প্রথম প্যারামিটার হলো ফাইলের নাম। যদি পাইথন স্ক্রিপ্ট ও ফাইলটি যদি কম্পিউটারের একই ডিরেক্টরিতে হয়, তবে শুধু ফাইলের নামটা স্ট্রিং হিসাবে দিতে হয়। যেমন-

```
file object = open ("file_name.txt")
```

আর ফাইলটি যদি আলাদা ডিরেক্টরিতে থাকে তবে পুরো পথটি স্ট্রিং হিসাবে দিতে হবে। যেমন-

```
file object = open ("C:\Users\Mahbub CPI\Desktop\66631\Python Link..txt")
```

অ্যাক্সেস মোড: দ্বিতীয় প্যারামিটার হলো অ্যাক্সেস মোড। পাইথন প্রোগ্রামে ফাইল নতুন করে তৈরি করা হবে, নাকি পূর্বে তৈরি করা কোনো ফাইল পরিবর্তন বা পরিবর্ধনের জন্য খোলা হবে, নাকি শুধু পড়ার জন্য ফাইল খোলা হবে তা ফাইল মোডের উপর নির্ভর করে।

বাফারিং: পাইথনে open () ফাংশনের অন্যতম একটি প্যারামিটার হলো বাফারিং। ফাইল অ্যাক্সেসিং-এর সময় যদি বাফারিং ভ্যালু ০ হয় তবে পাইথনে কোনো বাফারিং সংঘটিত হয় না। তবে যদি বাফারিং ভ্যালু 1 হয় তবে পাইথনে বাফারিং সংঘটিত হয়। আবার বাফারিং ভ্যালু যদি 1-এর চেয়ে বেশি হয় তবে নির্ধারিত বাফার সাইজের মধ্যে বাফারিং সংঘটিত হয়। আর যদি বাফারিং ভ্যালু নেগেটিভ হয় তাহলে বাফার সাইজ হবে ডিফল্ট সাইজ।

৩. File Input/Output Operation এর একটি programme লিখ। [বাকাশিবো: '২৪]

উত্তর: ফাইল ইনপুট/আউটপুট (File I/O) অপারেশন দেখানোর জন্য নিচে একটি সরল পাইথন প্রোগ্রাম লেখা হলো। এই প্রোগ্রামটি মূলত তিনটি কাজ করবে:

১. **ফাইলে লেখা (Write):** একটি নতুন ফাইল তৈরি করে তাতে কিছু ডেটা লিখবে। ২. **ফাইল বন্ধ করা (Close):** নিরাপত্তার জন্য ফাইলটি বন্ধ করবে। ৩. **ফাইল থেকে পড়া (Read):** ফাইলটি আবার খুলে তার ভেতরের সমস্ত ডেটা পড়ে কনসোলে প্রিন্ট করবে।

পাইথন প্রোগ্রাম: File I/O অপারেশন

Python

```
file_name = "sample_data.txt"
```

```
# --- ১. ফাইলে লেখা (Writing to a file) ---
```

```
# 'w' মোডে ফাইল খোলা: ফাইলটি তৈরি হবে বা পূর্বের কন্টেন্ট মুছে যাবে
```

```
with open(file_name, 'w') as file:
```

```
    file.write("এটি প্রথম লাইন।\n")
```

```
    file.write("পাইথন ফাইল I/O অপারেশন সফল।\n")
```

```
    file.write("এটি শেষ লাইন।")
```

```
print(f"ফাইলে লেখা সম্পন্ন হয়েছে: {file_name}")
```

```
# --- ২. ফাইল থেকে পড়া (Reading from the file) ---
```

```
# 'r' মোডে ফাইল খোলা: ফাইল থেকে পড়ার জন্য
```

```
with open(file_name, 'r') as file:
```

```
    content = file.read()
```

```
print("-" * 30)
```

```
print(f"'{file_name}' ফাইলের কন্টেন্ট:")
```

```
print(content)
```


প্রোগ্রামটির আউটপুট:

ফাইলে লেখা সম্পন্ন হয়েছে: sample_data.txt

'sample_data.txt' ফাইলের কন্টেন্ট:

এটি প্রথম লাইন।

পাইথন ফাইল I/O অপারেশন সফল।

এটি শেষ লাইন।

ব্যাখ্যা:

- `with open(file_name, 'w') as file::`
 - `open()` ফাংশনটি ফাইলটিকে 'w' (Write) মোডে খোলে।
 - `with` স্টেটমেন্টটি ব্যবহার করা হয়েছে, যা নিশ্চিত করে যে ফাইল অপারেশন শেষ হওয়ার সাথে সাথে ফাইলটি স্বয়ংক্রিয়ভাবে বন্ধ হয়ে যাবে, ম্যানুয়ালি `file.close()` কল করার প্রয়োজন হবে না।
- `file.write(...)`: এই মেথড ব্যবহার করে টেক্সট ফাইলের মধ্যে লেখা হয়।
- `content = file.read()`: ফাইল পয়েন্টার যেখানে আছে, সেখান থেকে ফাইলের সমস্ত কন্টেন্ট পড়ে `content` ভেরিয়েবলে সংরক্ষণ করে।

-সমাপ্তি