প্রি টেস্ট পরীক্ষা প্রস্তুতি - HSC 2026

প্রোগ্রামিং ধারণা | তানভীর হোছাইন

---

১. মৌলিক সংজ্ঞা ও ধারণা

> ### প্রঃ ১. কী ওয়ার্ড (Key Word) কী?
>
> **সংক্ষিপ্ত:** প্রোগ্রামিং ভাষার বিশেষ শব্দ, যার নিজস্ব অর্থ আছে।
>
> **বিস্তারিত:** প্রোগ্রামিং ভাষার কিছু বিশেষ শব্দ রয়েছে, যা কম্পাইলারের (Compiler) কাছে সুনির্দিষ্ট অর্থ বহন করে। এই শব্দগুলি সাধারণত ভেরিয়েবল বা ফাংশনের নাম হিসেবে ব্যবহার করা যায় না।
>
> *উৎস: [২, ১৩]*

> ### প্রঃ ২. চলক বা ভেরিয়েবল (Variable) কী?
>
> **সংক্ষিপ্ত:** মেমরিতে ডেটা রাখার জায়গা, যার মান পরিবর্তন করা যায়।
>
> **বিস্তারিত:** চলক হলো মেমোরিতে (Memory) ডেটা (Data) বা তথ্য জমা রাখার একটি স্থান বা পাত্র। প্রোগ্রামের প্রয়োজনে এই স্থানের মান পরিবর্তন করা যায়।
>
> *উৎস: [২, ১৩]*

> ### প্রঃ ৩. ধ্রুবক (Constant) কী?
>
> **সংক্ষিপ্ত:** প্রোগ্রামে একবার নির্দিষ্ট করা মান, যা আর পাল্টানো যায় না।
>
> **বিস্তারিত:** ধ্রুবক হলো এমন এক ধরনের ভ্যালু, যা একবার সংজ্ঞায়িত করার পর প্রোগ্রামের নির্বাহের সময় আর পরিবর্তন করা যায় না।
>
> *উৎস: [২, ৩, ১৩]*

> ### প্রঃ ৪. ফাংশন (Function) কী?
>
> **সংক্ষিপ্ত:** সুনির্দিষ্ট কিছু কাজ করার জন্য নির্দেশাবলীর ব্লক।
>
> **বিস্তারিত:** ফাংশন হলো কিছু সুনির্দিষ্ট কাজ সম্পন্ন করার জন্য একসাথে রাখা নির্দেশাবলীর একটি ব্লক বা সমষ্টি।
>
> *উৎস: [৩, ১৩]*

> ### প্রঃ ৫. অ্যারে (Array) কী?
>
> **সংক্ষিপ্ত:** একই নামের অধীনে একই ডেটা টাইপের অনেক ডেটা রাখা।
>
> **বিস্তারিত:** অ্যারে হলো একই ডেটা টাইপের বেশ কিছু উপাদান বা চলককে একসাথে একটি সাধারণ নামে সংরক্ষণ করার পদ্ধতি।
>
> *উৎস: [৩, ১৩]*

> ### প্রঃ ৬. এক্সপ্রেশন (Expression) কী?
>
> **সংক্ষিপ্ত:** চলক, ধ্রুবক ও অপারেটর দিয়ে তৈরি একটি বিবৃতি।
>
> **বিস্তারিত:** এক্সপ্রেশন হলো কিছু চলক, ধ্রুবক এবং অপারেটরের সমন্বয়ে গঠিত একটি গাণিতিক বা যৌক্তিক বিবৃতি, যার একটি নির্দিষ্ট মান থাকে।
>
> *উৎস: [৩, ১৩]*

> ### প্রঃ ৭. ডেটা টাইপ (Data Type) কী?
>
> **সংক্ষিপ্ত:** চলক কী ধরনের তথ্য (সংখ্যা, অক্ষর) রাখবে, তা ঠিক করা।
>
> **বিস্তারিত:** ডেটা টাইপ হলো চলক কী ধরনের তথ্য বা ডেটা সংরক্ষণ করবে (যেমন: পূর্ণ সংখ্যা, ভগ্নাংশ বা অক্ষর), তা নির্ধারণ করার প্রক্রিয়া।
>
> *উৎস: [৩, ১৩]*

> ### প্রঃ ৮. কোড (Code) কী?
>
> **সংক্ষিপ্ত:** প্রোগ্রামিং ভাষায় লেখা নির্দেশ।
>
> **বিস্তারিত:** প্রোগ্রামিং ভাষায় লেখা নির্দেশাবলীকেই সাধারণত কোড বলা হয়।
>
> *উৎস: [৪, ১৩]*

> ### প্রঃ ৯. লুপ (Loop) কী?
>
> **সংক্ষিপ্ত:** একটি শর্ত পূরণ না হওয়া পর্যন্ত কিছু নির্দেশ বারবার চালানো।
>
> **বিস্তারিত:** লুপ হলো প্রোগ্রামিংয়ের এমন একটি ব্যবস্থা, যার মাধ্যমে একটি নির্দিষ্ট শর্ত পূরণ না হওয়া পর্যন্ত কিছু নির্দেশাবলীকে বারবার চালনা করা হয়।
>
> *উৎস: [৪, ১৩]*

> ### প্রঃ ১০. ডিবাগ (Debug) কী?
>
> **সংক্ষিপ্ত:** প্রোগ্রামের ভুল বা ত্রুটি খুঁজে বের করা এবং ঠিক করা।
>
> **বিস্তারিত:** প্রোগ্রামের মধ্যেকার ত্রুটি (Error) বা ভুল খুঁজে বের করা এবং তা সংশোধন করার প্রক্রিয়াকে ডিবাগ (Debug) বলা হয়।
>
> *উৎস: [৪, ১৩]*

> ### প্রঃ ১১. সিনট্যাক্স ইরর (Syntax Error) কী?
>
> **সংক্ষিপ্ত:** প্রোগ্রামিং ভাষার ব্যাকরণ বা নিয়ম না মানলে যে ভুল হয়।
>
> **বিস্তারিত:** প্রোগ্রামিং ভাষার ব্যাকরণ (Grammar) বা নিয়মাবলী (Syntax) ভঙ্গের কারণে যে ত্রুটি দেখা দেয়, তাকে সিনট্যাক্স ইরর বা স্যানটেক্স ইরর বলা হয়।
>
> *উৎস: [৪, ১৩]*

> ### প্রঃ ১২. অপারেটর (Operator) কী?
>
> **সংক্ষিপ্ত:** গাণিতিক বা যৌক্তিক কাজ করার জন্য ব্যবহৃত চিহ্ন (+, -, %, ইত্যাদি)।
>
> **বিস্তারিত:** প্রোগ্রামে গাণিতিক যুক্তি এবং সম্পর্কযুক্ত কাজ করার জন্য যে সকল চিহ্ন (যেমন: +, -, *, /, %) ব্যবহার করা হয়, তাকে অপারেটর বলে।
>
> *উৎস: [৫, ১৩, ২২, ২৬]*

> ### প্রঃ ১৩. মডিউলাস অপারেটর (%) কী?
>
> **সংক্ষিপ্ত:** দুটি সংখ্যা ভাগ করার পর শুধু ভাগশেষ নির্ণয় করে।
>
> **বিস্তারিত:** মডিউলাস অপারেটর (%) হলো এমন একটি গাণিতিক অপারেটর, যা দুটি সংখ্যাকে ভাগ করার পর ভাগশেষ (Remainder) নির্ণয় করে। উদাহরণস্বরূপ: যদি ২৫ কে ৬ দ্বারা ভাগ করা হয়, তবে ভাগফল হবে ৪, কিন্তু ভাগশেষ হবে ১। এটি মডিউলাস অপারেটরের মাধ্যমে প্রকাশ করা হয়: ২৫ % ৬ = ১।
>
> *উৎস: [৫, ১৩, ২২, ২৬]*

> ### প্রঃ ১৪. টোকেন (Token) কী?
>
> **সংক্ষিপ্ত:** প্রোগ্রামের সবচেয়ে ছোট এবং অর্থপূর্ণ অংশ।
>
> **বিস্তারিত:** টোকেন হলো প্রোগ্রামিং ভাষার সবচেয়ে ছোট এবং অর্থপূর্ণ অংশ (যেমন: কীওয়ার্ড, অপারেটর, ভেরিয়েবলের নাম)।
>
> *উৎস: [৫, ১৩, ২৫]*

২. C ভাষা এবং পার্থক্যমূলক প্রশ্নাবলী

> ### প্রঃ ১৬. সি ভাষাকে কেন মধ্যম স্তরের ভাষা বলা হয়?
>
> **সংক্ষিপ্ত:** এটি উচ্চ স্তরের ভাষার মতো সহজ এবং নিম্ন স্তরের ভাষার মতো হার্ডওয়্যার নিয়ন্ত্রণ করতে পারে।
>
> **বিস্তারিত:** সি ভাষা একদিকে যেমন উচ্চস্তরের ভাষার (High-level Language) মতো মানুষের কাছাকাছি, তেমনি অন্যদিকে এটি নিম্নস্তরের ভাষার (Low-level Language) মতো হার্ডওয়্যারকে সরাসরি নিয়ন্ত্রণ করার ক্ষমতা রাখে।
>
> *উৎস: [৬, ১৪, ২৭]*

> ### প্রঃ ১৭. সি ভাষা কেন কেস সেনসিটিভ (Case Sensitive)?
>
> **সংক্ষিপ্ত:** ছোট হাতের অক্ষর (sum) এবং বড় হাতের অক্ষর (Sum) C ভাষায় ভিন্ন ধরা হয়।
>
> **বিস্তারিত:** এই ভাষা ছোট হাতের অক্ষর (lowercase) এবং বড় হাতের অক্ষরকে (uppercase) ভিন্ন ভিন্ন অক্ষর হিসেবে বিবেচনা করে। উদাহরণস্বরূপ, Sum এবং sum দুটি ভিন্ন ভেরিয়েবল বা কীওয়ার্ড হিসেবে গণ্য হবে।
>
> *উৎস: [৭, ১৪, ২৭]*

> ### প্রঃ ১৮. কম্পাইলারের চেয়ে ইন্টাপেটার বেশি বন্ধুভাবাপন্ন কেন?
>
> **সংক্ষিপ্ত:** ইন্টাপেটার লাইন ধরে ভুল দেখায়, তাই দ্রুত ভুল সংশোধন করা যায়।
>
> **বিস্তারিত:** ইন্টাপেটার প্রোগ্রামকে লাইন বাই লাইন নির্বাহ করে এবং কোনো লাইনে ত্রুটি পেলে তাৎক্ষণিকভাবে তা দেখায়। ফলে প্রোগ্রামার দ্রুত ভুল সংশোধন করতে পারে। কম্পাইলার পুরো প্রোগ্রাম একসাথে অনুবাদ করে, তাই ভুল শোধরানো কিছুটা কঠিন হয়।
>
> *উৎস: [৭, ১৪, ২৭]*

> ### প্রঃ ১৯. অ্যারে এবং চলক এক নয় ব্যাখ্যা কর।
>
> **সংক্ষিপ্ত:** চলক একটি ডেটা রাখে; অ্যারে একই নামের নিচে অনেক ডেটা রাখে।
>
> **বিস্তারিত:** চলক (Variable) একটি মাত্র ডেটা সংরক্ষণ করতে পারে, কিন্তু অ্যারে (Array) একই নামের অধীনে একই ডেটা টাইপের একাধিক ডেটা আইটেমকে পর পর সংরক্ষণ করতে পারে।
>
> *উৎস: [৮, ১৪, ২৭]*

> ### প্রঃ ২০. লোকাল এবং গ্লোবাল ভেরিয়েবলের পার্থক্য কী?
>
> **সংক্ষিপ্ত:** গ্লোবাল প্রোগ্রামের সব জায়গায় ব্যবহার করা যায়। লোকাল শুধু নির্দিষ্ট ফাংশনের মধ্যে কাজ করে।
>
> **বিস্তারিত:** গ্লোবাল ভেরিয়েবল প্রোগ্রামের যেকোনো অংশ থেকে ব্যবহার করা যায়। পক্ষান্তরে, লোকাল ভেরিয়েবল কেবলমাত্র নির্দিষ্ট ফাংশন বা ব্লকের মধ্যেই ব্যবহার করা যায়।
>
> *উৎস: [৮, ১৪, ২৭]*

> ### প্রঃ ২১. K++ এবং ++K ব্যাখ্যা কর।
>
> **সংক্ষিপ্ত:** K++ (Post): আগে মান ব্যবহার, পরে বৃদ্ধি। ++K (Pre): আগে বৃদ্ধি, পরে ব্যবহার।
>
> **বিস্তারিত:** K++ (Post-increment) প্রথমে K-এর বর্তমান মানটি ব্যবহার করে, তারপর K-এর মান ১ বাড়ায়। ++K (Pre-increment) প্রথমে K-এর মান ১ বাড়ায়, তারপর সেই বর্ধিত মানটি ব্যবহার করে।
>
> *উৎস: [৯, ১৪, ২৬]*

> ### প্রঃ ২২. for এবং do while লুপের মধ্যে কোনটি ব্যবহার করা সহজ?
>
> **সংক্ষিপ্ত:** for লুপ সহজ, কারণ এর প্রাথমিকীকরণ, শর্ত ও পরিবর্তন এক জায়গায় লেখা যায়।
>
> **বিস্তারিত:** সাধারণত for লুপ ব্যবহার করা সহজ কারণ এর মধ্যে লুপের তিনটি প্রধান অংশ—প্রাথমিকীকরণ (initialization), শর্ত (condition) এবং মান পরিবর্তন (increment/decrement)—একসাথে লেখা যায়, যা কোডকে আরও সুসংগঠিত করে তোলে।
>
> *উৎস: [৯, ১৪, ২৬]*

৩. নিয়মাবলী ও ফাংশন

> ### প্রঃ ২৩. ভেরিয়েবল নাম লেখার নিয়মগুলো ব্যাখ্যা কর।
>
> **সংক্ষিপ্ত:** বর্ণ দিয়ে শুরু করতে হবে। স্পেস (ফাঁকা জায়গা) বা অন্যান্য বিশেষ চিহ্ন (_ ছাড়া) ব্যবহার করা যাবে না।
>
> **বিস্তারিত:** ভেরিয়েবলের নাম লেখার সময় নিম্নলিখিত নিয়মগুলো মানতে হবে: ১. ভেরিয়েবল নাম অবশ্যই বর্ণ হতে হবে (a-z, A-Z) ২. ভেরিয়েবল নামের সাথে সংখ্যা থাকতে পারে (যেমন: a1) ৩. ভেরিয়েবল নামের পূর্বে সংখ্যা হতে পারবে না ৪. শুধু সংখ্যা ভেরিয়েবল হতে পারবে না ৫. ভেরিয়েবল নামে পাশে হোয়াইট স্পেস হতে পারবে না ৬. নামের মাঝে শুধুমাত্র আন্ডার স্কোর (_) ব্যবহার করা যাবে ৭. আন্ডার স্কোর ছাড়া অন্য কোনো স্পেশাল ক্যারেক্টার ব্যবহার করা যাবে না
>
> *উৎস: [১০, ১৫, ২৩, ২৪, ২৮]*

> **প্রঃ ২৪. ভেরিয়েবল ঘোষণার ক্ষেত্রে অনুসরণীয় পদক্ষেপ ব্যাখ্যা কর।**
>
> **সংক্ষিপ্ত:** ব্যবহারের আগে অবশ্যই ভেরিয়েবলের ডেটা টাইপ উল্লেখ করে ঘোষণা করতে হয়।
>
> **বিস্তারিত:** ভেরিয়েবল ব্যবহারের আগে অবশ্যই তার ডেটা টাইপ উল্লেখ করে ঘোষণা করতে হয়। এর ফলে কম্পাইলার বুঝতে পারে মেমরিতে কতটুকু জায়গা এবং কী ধরনের ডেটা সংরক্ষণ করতে হবে।
>
> *উৎস: [১১, ১৫, ২৮]*

> **প্রঃ ২৫. মেইন ফাংশন (main()) ব্যাখ্যা কর।**
>
> **সংক্ষিপ্ত:** এটি C প্রোগ্রামের প্রবেশদ্বার। প্রোগ্রাম নির্বাহ (Execution) এখান থেকেই শুরু হয়।
>
> **বিস্তারিত:** main() ফাংশন হলো C প্রোগ্রামের প্রবেশদ্বার। যেকোনো C প্রোগ্রাম নির্বাহ (Execute) শুরু হয় এই ফাংশন থেকেই।
>
> *উৎস: [১১, ১৫, ২৯]*

> **প্রঃ ২৬. সি ভাষায় কেন হেডারফাইল ব্যবহার করা হয়?**
>
> **সংক্ষিপ্ত:** প্রোগ্রামে প্রয়োজনীয় ফাংশন (যেমন: ইনপুট-আউটপুট) ব্যবহারের জন্য তাদের সংজ্ঞা যুক্ত করতে।
>
> **বিস্তারিত:** হেডারফাইল (Header File) হলো এমন ফাইল, যাতে বিভিন্ন প্রয়োজনীয় ফাংশনের (যেমন: ইনপুট-আউটপুট ফাংশন) পূর্বনির্ধারিত সংজ্ঞা থাকে। প্রোগ্রামিংয়ের কাজ সহজ করার জন্য এবং এই স্ট্যান্ডার্ড ফাংশনগুলি ব্যবহার করার জন্য হেডারফাইলগুলি #include নির্দেশ ব্যবহার করে প্রোগ্রামে যুক্ত করা হয়।
>
> *উৎস: [১২, ১৫, ২৯]*

> **প্রঃ ২৭. scanf ("%d", &a); ব্যাখ্যা কর।**
>
> **সংক্ষিপ্ত:** কী-বোর্ড থেকে একটি পূর্ণ সংখ্যা ইনপুট নেয় এবং তা a ভেরিয়েবলে জমা করে।
>
> **বিস্তারিত:** এই স্টেটমেন্টটি ব্যবহারকারী থেকে একটি পূর্ণ সংখ্যা (%d) ইনপুট হিসেবে গ্রহণ করে। সেই মানটি a নামক ভেরিয়েবলের মেমরি অ্যাড্রেসে (&a) সংরক্ষণ করা হয়।
>
> *উৎস: [১৩, ১৫, ২১]*

> **প্রঃ ২৮. #include <stdio.h> ব্যাখ্যা কর।**
>
> **সংক্ষিপ্ত:** স্ট্যান্ডার্ড ইনপুট/আউটপুট লাইব্রেরি যুক্ত করার নির্দেশ।
>
> **বিস্তারিত:** এই নির্দেশটি কম্পাইলারকে জানায় যে স্ট্যান্ডার্ড ইনপুট/আউটপুট লাইব্রেরি ফাইল (stdio.h) যেন বর্তমান প্রোগ্রামে যুক্ত করা হয়। এই লাইব্রেরির মধ্যে printf() এবং scanf() এর মতো ফাংশনগুলির সংজ্ঞা থাকে।
>
> *উৎস: [১২, ১৫, ২৯]*

৪. গাণিতিক ও প্রোগ্রামিংয়ের প্রশ্ন

> **প্রঃ ২৯. প্রত্যেকটি প্রোগ্রামের ৩টি অংশ থাকে—ব্যাখ্যা কর।**
>
> **সংক্ষিপ্ত:** প্রতিটি প্রোগ্রামের ৩টি প্রধান অংশ থাকে (ইনপুট, প্রসেসিং ও আউটপুট)।
>
> **বিস্তারিত:** প্রত্যেকটি প্রোগ্রামের ৩টি অংশ থাকে—এই বিষয়ে ব্যাখ্যা করতে বলা হয়েছে। সাধারণত একটি প্রোগ্রামের প্রধান অংশগুলি হলো—ইনপুট (Input), প্রসেসিং (Processing) এবং আউটপুট (Output)।
>
> *উৎস: [১৬, ১৮]*

> **প্রঃ ৩০. কন্ট্রোল স্টেটমেন্ট (Control Statement) কী?**
>
> **সংক্ষিপ্ত:** প্রোগ্রামের প্রবাহ নিয়ন্ত্রণকারী স্টেটমেন্টগুলো হলো: কন্ডিশনাল ও লুপস।
>
> **বিস্তারিত:** কন্ট্রোল স্টেটমেন্ট হলো সেই নির্দেশাবলী যা প্রোগ্রামের নির্বাহের প্রবাহ নিয়ন্ত্রণ করে। এগুলি মূলত দুই প্রকার: - কন্ডিশনাল স্টেটমেন্ট: শর্তের ভিত্তিতে কাজ করে (যেমন: if, if-else, else if, Switch) - লুপস: পুনরাবৃত্তিমূলক কাজ করে (যেমন: for, while, do-while, goto, Continue)
>
> *উৎস: [১৬, ২২, ২৬]*

> **প্রঃ ৩১. নিম্নলিখিত গাণিতিক সিরিজগুলি উল্লেখ কর।**
>
> **সংক্ষিপ্ত:** $1+2+3+..+n$, $1^2+2^2+3^2+..+n^2$, এবং $1^3+2^3+3^3+..+n^3$
>
> **বিস্তারিত:** এই তিনটি গাণিতিক সিরিজের যোগফল নির্ণয় করতে বলা হয়েছে, যা সাধারণত প্রোগ্রামিংয়ে লুপ ব্যবহার করে সমাধান করা হয়।
>
> *উৎস: [১৬, ১৯, ২২]*

> **প্রঃ ৩২. C কোড a = b%25; এর আউটপুট কত হবে? (যেখানে b = 125)**
>
> **সংক্ষিপ্ত:** আউটপুট হবে ০ (শূন্য)।
>
> **বিস্তারিত:** কোডে b = 125 সেট করা আছে এবং a = b % 25; এই লাইনের মাধ্যমে ১২৫ কে ২৫ দিয়ে ভাগ করার পর ভাগশেষ নির্ণয় করা হয়েছে। যেহেতু ১২৫ সংখ্যাটি ২৫ দ্বারা সম্পূর্ণভাবে বিভাজ্য, তাই ভাগশেষ হবে ০ (শূন্য)। এই উদ্দীপকে ব্যবহৃত প্রোগ্রামিং ভাষাটি General purpose Language।
>
> *উৎস: [১৬, ১৮]*

> **প্রঃ ৩৩. প্রেক্ষাপট ১ অনুসারী কার বয়স সবচেয়ে বেশি, এটা নির্ণয়ের জন্য অ্যালগরিদম লেখ।**
>
> **সংক্ষিপ্ত:** A, B এবং C (বয়স 50, 35, 13) এর মধ্যে কার বয়স বেশি, তা নির্ণয়ের জন্য অ্যালগরিদম লিখতে বলা হয়েছে।
>
> **বিস্তারিত:** প্রেক্ষাপট: A, B এবং C এর বয়স যথাক্রমে 50, 35, 13। এটি তুলনামূলক কন্ডিশনাল স্টেটমেন্ট ব্যবহার করে সমাধান করা যায়। অ্যালগরিদম: ১. শুরু (Start) ২. A, B, C এর মান ইনপুট নাও (50, 35, 13) ৩. যদি (if) A > B এবং A > C হয়, তাহলে A বৃহত্তম। ৪. তা না হলে যদি (else if) B > C হয়, তাহলে B বৃহত্তম। ৫. অন্যথায় (else), C বৃহত্তম। ৬. বৃহত্তম মান প্রিন্ট কর ৭. শেষ (End)
>
> *উৎস: [১৬, ১৯, ২১, ২২]*

৫. কোড বিশ্লেষণ ও অন্যান্য ধারণা

> ### প্রঃ ৩৪. প্রদত্ত C কোড a = b%25; এর আউটপুট কত হবে?
>
> **সংক্ষিপ্ত:** আউটপুট হবে ০ (শূন্য)।
>
> **বিস্তারিত:** প্রোগ্রাম: #include <stdio.h> main(){ int a,b; b = 125; a = b%25; printf ("%d",a); } বিশ্লেষণ: এই প্রোগ্রামে b এর মান 125 এবং a তে b % 25 (125 কে 25 দিয়ে ভাগ করার পর ভাগশেষ) সংরক্ষণ করা হয়েছে। যেহেতু 125, 25 দ্বারা সম্পূর্ণভাবে বিভাজ্য, তাই ভাগশেষ হবে 0 (শূন্য)। প্রোগ্রামটির আউটপুট হবে ০ (শূন্য)। এই উদ্দীপকে ব্যবহৃত প্রোগ্রামিং ভাষাটি General purpose Language।
>
> *উৎস: [১৬, ১৮]*

> ### প্রঃ ৩৫. প্রোগ্রাম ২: a=b/0.25 এর ফলাফল কী হবে?
>
> **সংক্ষিপ্ত:** কোড a = b/0.25 এর ফলাফল a=0 হতে পারে, কারণ a পূর্ণসংখ্যা (int) হিসেবে ঘোষিত। (গাণিতিকভাবে ফল 500 হলেও ডেটা টাইপ সমস্যার জন্য 0 হতে পারে)।
>
> **বিস্তারিত:** প্রোগ্রাম: #include <stdio.h> main () { int a,b; b=125 a=b/0.25 printf("%d",a); } বিশ্লেষণ: গাণিতিকভাবে 125 / 0.25 = 500 হওয়ার কথা। কিন্তু যেহেতু a কে পূর্ণসংখ্যা (int) হিসেবে ঘোষণা করা হয়েছে এবং ভাগ করা হচ্ছে ভগ্নাংশ (0.25) দিয়ে, ফলে ডেটা টাইপ মিসম্যাচ ঘটতে পারে। উৎসটিতে লেখা আছে: এখানে a=0।
>
> *উৎস: [২১]*

> ### প্রঃ ৩৬. প্রেক্ষাপট ২ এর জন্য প্রোগ্রাম লিখা এবং বিশ্লেষণ করো।
>
> **সংক্ষিপ্ত:** শেষ নাম্বার 80 এবং সিরিজের বেত্তাফল 11, এর জন্য পাওয়ার ভাষা প্রোগ্রাম লিখতে বলা হয়েছে।
>
> **বিস্তারিত:** প্রেক্ষাপট: শেষ নাম্বার 80 এবং সিরিজের বেত্তাফল 11। ব্যাখ্যা: প্রেক্ষাপট ২ এর পাওয়ার ভাষা প্রোগ্রাম লিখতে এবং বিশ্লেষণ করতে বলা হয়েছে। এটি সাধারণত লুপ ব্যবহার করে একটি গাণিতিক সিরিজের যোগফল নির্ণয়ের প্রোগ্রাম হবে, যেখানে সিরিজের শেষ পদ 80 এবং সিরিজের মোট যোগফল বা 'বেত্তাফল' 11 হতে হবে।
>
> *উৎস: [১৬, ১৯, ২২]*

> ### প্রঃ ৩৭. #include <stdio.h> ব্যাখ্যা কর।
>
> **সংক্ষিপ্ত:** স্ট্যান্ডার্ড ইনপুট/আউটপুট লাইব্রেরি যুক্ত করার নির্দেশ।
>
> **বিস্তারিত:** এই নির্দেশটি কম্পাইলারকে জানায় যে স্ট্যান্ডার্ড ইনপুট/আউটপুট লাইব্রেরি ফাইল (stdio.h) যেন বর্তমান প্রোগ্রামে যুক্ত করা হয়। এই লাইব্রেরির মধ্যে printf() এবং scanf() এর মতো ফাংশনগুলির সংজ্ঞা থাকে।
>
> *উৎস: [১২, ১৫, ২৯]*

> ### প্রঃ ৩৮. প্রোগ্রামের ভিত্তি (Base of Program) কী?
>
> **সংক্ষিপ্ত:** প্রোগ্রামের ভিত্তি কী—এই সংজ্ঞাটি জানতে চাওয়া হয়েছে।
>
> **বিস্তারিত:** প্রোগ্রামের ভিত্তি কী—এই বিষয়ে সংজ্ঞা জানতে চাওয়া হয়েছে। প্রোগ্রামের ভিত্তি বলতে সাধারণত প্রোগ্রামিংয়ের মৌলিক ধারণা, যেমন: ডেটা টাইপ, ভেরিয়েবল, অপারেটর, এবং কন্ট্রোল স্ট্রাকচারকে বোঝানো হয়।
>
> *উৎস: [৬, ১৯, ২৫]*

> ### প্রঃ ৩৯. অবজেক্ট প্রোগ্রাম (Object Program) কী?
>
> **সংক্ষিপ্ত:** অবজেক্ট প্রোগ্রাম কী, তা জানতে চাওয়া হয়েছে।
>
> **বিস্তারিত:** অবজেক্ট প্রোগ্রাম হলো সোর্স কোড কম্পাইল করার পর যে মেশিন ল্যাঙ্গুয়েজ কোড তৈরি হয়, সেটি। এটি কম্পিউটার সরাসরি বুঝতে এবং execute করতে পারে। সোর্স প্রোগ্রাম থেকে কম্পাইলার বা অ্যাসেম্বলার দিয়ে অবজেক্ট প্রোগ্রাম তৈরি করা হয়।
>
> *উৎস: [১৯, ২৫]*

> ### প্রঃ ৪০. ন্যাচারাল ল্যাঙ্গুয়েজ, প্রচলক, অ্যাসেম্বলার/অ্যাসেম্বলি ভাষা এবং মুখ—এই ধারণাগুলোর সংজ্ঞা দাও।
>
> **সংক্ষিপ্ত:** ন্যাচারাল ল্যাঙ্গুয়েজ, প্রচলক, অ্যাসেম্বলার/অ্যাসেম্বলি ভাষা এবং মুখ—এই সংজ্ঞাগুলোও উৎসগুলোতে জানতে চাওয়া হয়েছে।
>
> **বিস্তারিত:** উৎসগুলিতে নিম্নলিখিত সংজ্ঞাগুলি জানতে চাওয়া হয়েছে: • ন্যাচারাল ল্যাঙ্গুয়েজ (Natural Language): মানুষের স্বাভাবিক ভাষা যা দৈনন্দিন যোগাযোগে ব্যবহৃত হয় (যেমন: বাংলা, ইংরেজি)। • প্রচলক (Parameter): ফাংশনে পাঠানো মান বা আর্গুমেন্ট যা ফাংশন কাজ করার জন্য ব্যবহার করে। • অ্যাসেম্বলার/অ্যাসেম্বলি ভাষা: নিম্ন স্তরের প্রোগ্রামিং ভাষা যা মেশিন ল্যাঙ্গুয়েজের কাছাকাছি এবং অ্যাসেম্বলার দিয়ে মেশিন কোডে রূপান্তরিত হয়। • মুখ (Mukh): এই প্রসঙ্গে সম্ভবত প্রোগ্রামিং ইন্টারফেস বা entry point বোঝানো হয়েছে।
>
> *উৎস: [৬, ২৫]*
