



বেসিক সি প্রোগ্রামিং

শিহাব সাহারিয়ার সিজান

A Project Of Club Nebula

www.clubnebula.com
© 2013 CLUB NEBULA LTD

About This Project(Book)



Sihab Sahariar Sizan

Software Programmer And System Developer

CEO And Founder Of Team Error

Student Of Natore Govt. Boys' High School

Phone:-01782989898

Facebook:-www.facebook.com/sizan.first

Member Club Nebula(IT Department)

IT Executive Officer

BD-Soft Limited

Bangladesh Association Of Junior Programmer Foundation

Website:- <http://teamerror.wc.it>

Website:- <http://Sizansoft-vb.blogspot.com>

Website:- <http://clunnebula.blogspot.com>

বিশেষ দ্রষ্টব্যঃ-বেশ কিছু আর্টিকেল ইন্টারনেট থেকে নেওয়া হয়েছে। এবং সবার নাম আসলে দেওয়া সম্ভব হয় নাই।বইটি স্কুলের ছাত্র-ছাত্রীদের জন্য লিখেছি।ক্লাশ৬ থেকে ক্লাশ ৯-১০ এর স্টুডেন্টদের জন্যই লিখা।তাদের বেসিক ধারণা এবং মজায় মজায় প্রোগ্রামিং শেখানোই ছিল আমার উদ্দেশ্য।তাইতো এই বইটি লেখা।

সূচীপত্র

১ম অধ্যায়ঃ-কম্পিউটার প্রোগ্রামিং কি ???

অধ্যায় ২য়ঃ-প্রথম প্রোগ্রাম ও কোডব্লক্স বিস্তারিত

অধ্যায় ৩য়ঃ-Concept Of Flowchart & Algorithms

অধ্যায় ৪র্থঃ- ডাটা টাইপ, ইনপুট ও আউটপুট এবং ভেরিয়ারবলস

৫ম অধ্যায়ঃ-প্রোগ্রামিং ভাবনা

৬ষ্ঠ অধ্যায়ঃ-C/C++ Struct

৭ম অধ্যায়-Ifelse if..... else function.

৮ম অধ্যায়-Live Project Using If...Else

৯ম অধ্যায়- While & loop...

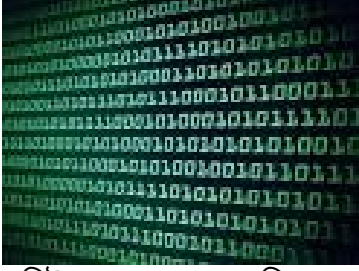
অধ্যায় ১০-পয়েন্টার বিস্তারিত

অধ্যায় ১১-টেস্ট এক্সাম ও সমস্যা সমাধান

অধ্যায় ১২-প্রয়োজনীয় বিষয়

চল শুরু করি □

১ম অধ্যায়ঃকম্পিউটার প্রোগ্রামিং কি ???



ছবিটা দেখে মনে হয় কিছু বুঝা না তাই না
আচ্ছা নিচের বিষয়গুলো মনোযোগ দিয়ে পড় ইনশাআল্লাহ বুঝবে।

কম্পিউটার তো আসলে গণনা করার যন্ত্র, তাই না? যদিও আমরা এটি দিয়ে গান শুনি, ভিডিও দেখি, গেমস খেলি, আরও নানা কাজ করি। আসলে শেষ পর্যন্ত কম্পিউটার বোঝে শূন্য (0) আর একের (1) হিসাব। তাই ব্যবহারকারী (user) যা-ই করুক না কেন, কম্পিউটার কিন্তু সব কাজ গণনার মাধ্যমেই করে। কম্পিউটারের ব্যবহার এত ব্যাপক হওয়ার পেছনে অন্যতম কারণ হচ্ছে নানা রকম সফটওয়্যার দিয়ে নানা ধরনের কাজ করা যায় কম্পিউটারে। এসব সফটওয়্যার তৈরি করতে হয় প্রোগ্রাম লিখে অর্থাৎ কী হলে কী করবে এটি প্রোগ্রামের সাহায্যে কম্পিউটারকে বোঝাতে হয়।

একসময় কিন্তু কেবল 0 আর 1 ব্যবহার করেই কম্পিউটারের প্রোগ্রাম লিখতে হতো। কারণ কম্পিউটার তো 0, 1 ছাড়া আর কিছু বোঝে না, আর কম্পিউটারকে দিয়ে কোনো কাজ করতে চাইলে তো তার ভাষাতেই কাজের নির্দেশ দিতে হবে। 0, 1 ব্যবহার করে যে প্রোগ্রামিং করা হতো, তার জন্য যে ভাষা ব্যবহৃত হতো, তাকে বলা হয় মেশিন ল্যাঙ্গুয়েজ। তারপর এল অ্যাসেম্বলি ল্যাঙ্গুয়েজ। এতে প্রোগ্রামাররা কিছু ইনস্ট্রাকশন যেমন ADD (যোগ), MUL (গুণ) ইত্যাদি ব্যবহারের সুযোগ পেল। আর এই ভাষাকে 0, 1-এর ভাষায় নিয়ে কাজ করার দায়িত্ব পড়ল অ্যাসেম্বলারের ওপর, প্রোগ্রামারদের সে বিষয়ে ভাবতে হতো না। কিন্তু মানুষের চাহিদার তো শেষ নেই। নতুন নতুন চাহিদার ফলে নতুন নতুন জিনিসের উদ্ভব হয়। একসময় দেখা গেল যে অ্যাসেম্বলি ল্যাঙ্গুয়েজ দিয়েও কাজ করা ঝামেলা হয়ে যাচ্ছে। তাই বড় বড় প্রোগ্রাম লিখার জন্য আরও সহজ ও উন্নত নানা রকম প্রোগ্রামিং ভাষা তৈরি হলো। যেমন - ফরট্রান (Fortran), বেসিক (Basic), প্যাসকেল (Pascal), সি (C)। তবে এখানেই শেষ নয়, এরপর এল আরও অনেক ল্যাঙ্গুয়েজ, যার মধ্যে অন্যতম হচ্ছে, সি প্লাস প্লাস (C++), ভিজুয়াল বেসিক (Visual Basic), জাভা (Java), সি শার্প (C#), পার্ল (Perl), পিএইচপি (PHP), পাইথন (Python), রুবি (Ruby)। এখনো কম্পিউটার বিজ্ঞানীরা নিত্যনতুন প্রোগ্রামিং ভাষা তৈরি করে যাচ্ছেন। প্রোগ্রামাররা এসব ভাষা ব্যবহার করে প্রোগ্রাম লেখেন আর প্রতিটি ভাষার রয়েছে আলাদা কম্পাইলার, যার কাজ হচ্ছে ওই প্রোগ্রামকে কম্পিউটারের বোধগম্য ভাষায় রূপান্তর করা, তাই এটি নিয়ে প্রোগ্রামারদের ভাবতে হয় না।

প্রোগ্রাম লিখার সময় প্রোগ্রামারকে তিনটি প্রধান কাজ করতে হয়। প্রথমে তার বুঝতে হয় যে সে আসলে কী করতে যাচ্ছে, মানে তার প্রোগ্রামটি আসলে কী কাজ করবে। তারপর চিন্তাভাবনা করে এবং যুক্তি (logic) ব্যবহার করে অ্যালগরিদম দাঁড় করাতে হয়। মানে, লজিকগুলো ধাপে

ধাপে সাজাতে হয়। এর পরের কাজটি হচ্ছে অ্যালগরিদমটাকে কোনো একটি প্রোগ্রামিং ভাষায় রূপান্তর করা, যাকে আমরা বলি কোডিং করা। একেক ধরনের কাজের জন্য একেক ল্যাঙ্গুয়েজ বেশি উপযোগী।

এই বইতে আমরা প্রোগ্রামিংয়ের মৌলিক কিছু জিনিস শেখার চেষ্টা করব এবং প্রোগ্রামগুলো আমরা লিখব সি ল্যাঙ্গুয়েজ ব্যবহার করে। আমি ধরে নিচ্ছি, তোমরা কম্পিউটার ব্যবহার করে অভ্যস্ত এবং প্রোগ্রামিং জিনিসটার সঙ্গে সম্পূর্ণ অপরিচিত। আর সি ল্যাঙ্গুয়েজ ব্যবহার করার পেছনে কারণ হচ্ছে, এটি বেশ পুরোনো হলেও অত্যন্ত শক্তিশালী ও জনপ্রিয় ল্যাঙ্গুয়েজ। প্রোগ্রামিংয়ের মৌলিক জিনিসগুলো বোঝার জন্য সি ভাষা অত্যন্ত সহায়ক। আর জনপ্রিয় সব প্রোগ্রামিং প্রতিযোগিতায় যে অল্প কয়েকটি ল্যাঙ্গুয়েজ ব্যবহার করা যায়, তার মধ্যে সি অন্যতম। আমরা অবশ্য সি ল্যাঙ্গুয়েজের পুরোটা এখানে শিখব না, কেবল মৌলিক বিষয়গুলো নিয়ে কাজ করতে যা দরকার সেটি দেখব। এই বইটি পড়ার পরে তোমরা কেবল সি-এর জন্য কোন বই পড়তে পারো অথবা অন্য কোনো ভাষা (যেমন- সি প্লাস প্লাস, জাভা কিংবা পাইথন) শেখা শুরু করে দিতে পারো। বইয়ের পরিশিষ্ট অংশে আমি কিছু বইয়ের নাম দিয়েছি, যা তোমাদের কাজে লাগবে।

বইটি পড়তে তোমাদের তিনটি জিনিস লাগবে, কম্পিউটার (ইন্টারনেট সংযোগ থাকলে ভালো হয়), সি ল্যাঙ্গুয়েজের কম্পাইলার এবং যথেষ্ট সময়। তাড়াহড়ো না করে দুই থেকে তিন মাস সময় নিয়ে বইটি পড়লে ভালো হয়। প্রোগ্রামিং শেখার জন্য কেবল পড়াই যথেষ্ট নয়, পাশাপাশি কোডিং করতে হবে। বইয়ের প্রতিটি উদাহরণ নিজে নিজে কোড করে কম্পিউটারে চালিয়ে দেখতে হবে। যখনই আমি কোনো প্রশ্ন করব, সেটি নিয়ে চিন্তা করতে হবে। তার জন্য যদি দু-তিন ঘণ্টা বা দু-তিন দিন সময় লাগে লাগুক, কোনো ক্ষতি নেই, বরং দীর্ঘ সময় কোনো সমস্যার সমাধান নিয়ে চিন্তা করার অভ্যাসটি খুব জরুরি। কোনো অধ্যায় পুরোপুরি বোঝার আগে পরের অধ্যায় পড়া শুরু করা যাবে না। আবার কোনো অংশ যদি তোমার কাছে খুব সহজ মনে হয়, সেই অংশ ঠিকভাবে না পড়ে এবং প্রোগ্রামগুলো না করে পরের অংশে চলে যেয়ো না কিছু। সাধারণ পড়ালিখার সঙ্গে প্রোগ্রামিং শেখার অনেক পার্থক্য। এখানে পড়ার সঙ্গে সঙ্গে কাজ করাও জরুরি। আর এই বই পড়েই কিছু তুমি প্রোগ্রামার হয়ে যাবে না, বইটি পড়ে তুমি প্রোগ্রামার হওয়া শুরু করবে।

এবার আসা যাক, কম্পাইলার পাবে কোথায়? সি-এর জন্য বেশ কিছু কম্পাইলার আছে। তুমি যদি লিনাক্স কিংবা ম্যাক ব্যবহারকারী হও, তবে সবচেয়ে ভালো হচ্ছে gcc। অধিকাংশ লিনাক্সেই এটি আগে থেকে ইনস্টল করা থাকে। তোমার কম্পিউটারে না থাকলে এটি ইনস্টল করে নিতে হবে। আর উইন্ডোজ ব্যবহার করলে তুমি Codeblocks (<http://www.codeblocks.org/>) ব্যবহার করতে পারো। এটি একটি ফ্রি ও ওপেন সোর্স IDE (Integrated Development Environment) এবং ম্যাক আর লিনাক্সেও চলে। এমনিতে সাধারণ কোনো টেক্সট এডিটর (যেমন: নোটপ্যাড, জিএডিট, কেরাইট) ব্যবহার করে কোড লিখে সেটি কম্পাইলার দিয়ে কম্পাইল করে রান করা যায়। তবে অধিকাংশ আইডিই (IDE) গুলোতেই নিজস্ব টেক্সট এডিটর ও কম্পাইলার থাকে। প্রোগ্রাম রান করার ব্যবস্থাও থাকে। এ ছাড়াও নানা ধরনের টুলস্ থাকে।

Codeblocks টা সরাসরি তুমি <http://www.codeblocks.org> সাইট থেকে ডাউনলোড ও ইনস্টল করতে পারো। Downloads পেইজে Binaries-এ গেলে উইন্ডোজের জন্য তুমি দুটি

অপশন দেখবে: codeblocks-10.05-setup.exe ও codeblocks-10.05mingw-setup.exe। তুমি দ্বিতীয়টি ডাউনলোড করবে (74.0 MB)। আর ইনস্টল করার কাজটি অন্য যেকোনো সফটওয়্যার বা গেমসের মতোই। যারা উবুন্টু ব্যবহার করো, তারা Ubuntu Software Center (Applications > Ubuntu Software Center) থেকে এটি ডাউনলোড করতে পারো।

প্রোগ্রামিং চর্চার বিষয়। ইন্টারনেটে বেশ কিছু ওয়েবসাইট আছে, যেখানে প্রচুর সমস্যা দেওয়া আছে যেগুলো প্রোগ্রামের সাহায্যে সমাধান করতে হয়। সব জায়গাতেই তুমি সি ল্যাঙ্গুয়েজে প্রোগ্রামিং করতে পারবে। এর মধ্যে কিছু কিছু ওয়েবসাইট আবার নিয়মিত প্রোগ্রামিং প্রতিযোগিতারও আয়োজন করে। এসব প্রতিযোগিতায় অংশগ্রহণ নিঃসন্দেহে তোমার প্রোগ্রামিং-দক্ষতা বৃদ্ধি করবে আর সেই সঙ্গে বিশ্বের নানা দেশের প্রোগ্রামারদের সঙ্গে মেশারও সুযোগ করে দেবে। অবশ্য প্রোগ্রামিং প্রতিযোগিতায় ভালো করতে হলে কেবল প্রোগ্রামিং জানলেই চলবে না, গাণিতিক দক্ষতাও যথেষ্ট গুরুত্বপূর্ণ। পরিশিষ্ট অংশে প্রোগ্রামিং প্রতিযোগিতা নিয়ে আলাপ করব।

বইয়ের প্রতিটি প্রোগ্রামের নিচে আমি একটি নম্বর দিয়েছি। প্রোগ্রামের নম্বর যদি ২.৫ হয়, তার মানে হচ্ছে এটি দ্বিতীয় অধ্যায়ের পাঁচ নম্বর প্রোগ্রাম।

এটি কিন্তু কোনো গল্পের বই নয়। তাই বিছানায় শুয়ে-বসে পড়া যাবে না। বইটি পড়ার সময় কম্পিউটার চালু রাখতে হবে এবং প্রতিটি উদাহরণ সঙ্গে সঙ্গে প্রোগ্রাম লিখে দেখতে হবে, কোনো সমস্যা সমাধান করতে দিলে তখনই সেটি সমাধানের চেষ্টা করতে হবে। মনে রাখবে, যত বেশি প্রোগ্রামিং তত বেশি আনন্দ।

আশা করছি, তুমি ধৈর্য নিয়ে বাকি অধ্যায়গুলো পড়বে এবং সবগুলো প্রোগ্রাম কম্পিউটারে চালিয়ে দেখবে। তোমার জন্য শুভ কামনা।

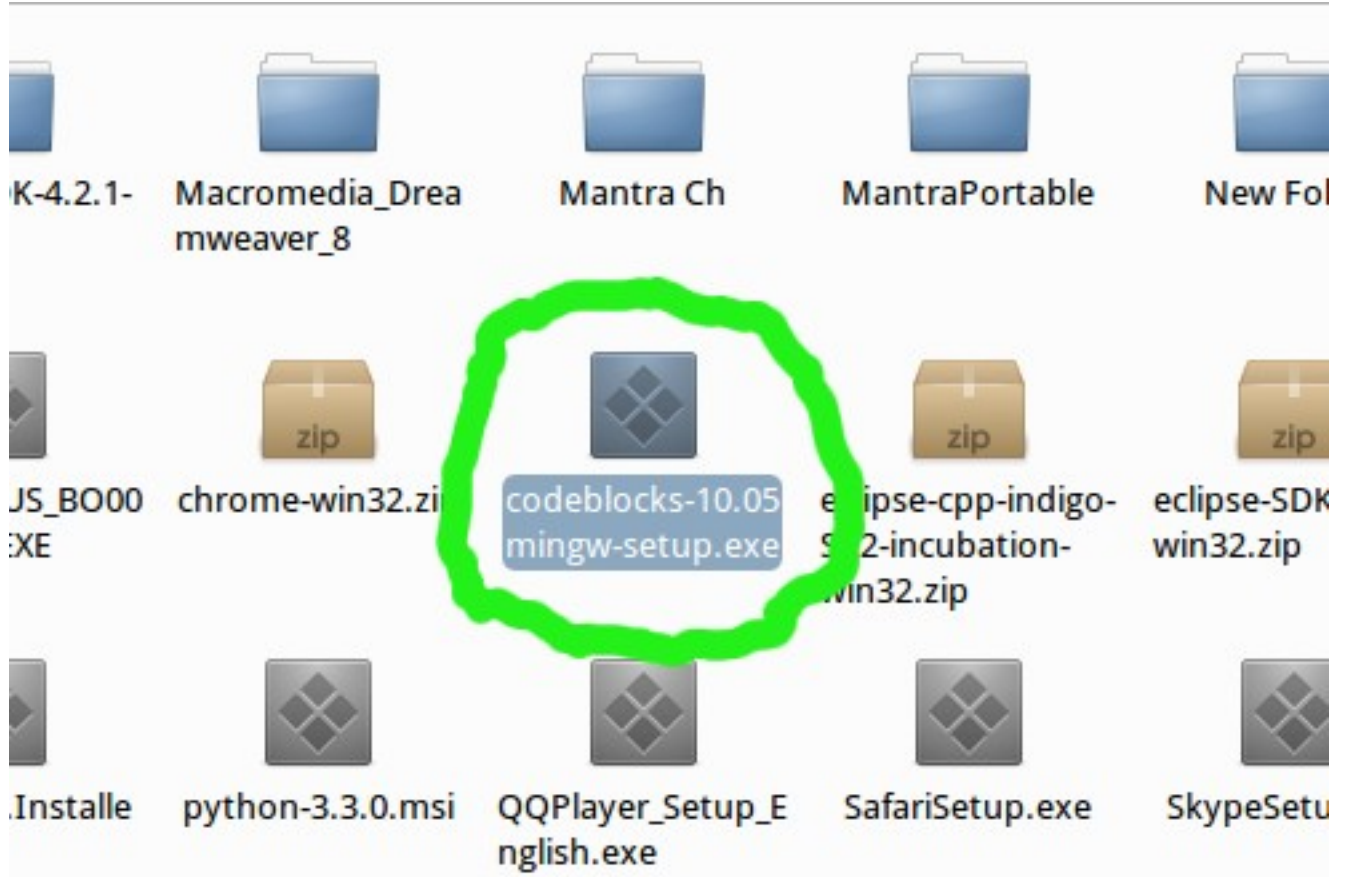
অধ্যায় ২য়ঃ-প্রথম প্রোগ্রাম ও কোডব্লক্স বিস্তারিত

প্রোগ্রামিংয়ের জগতে স্বাগতম!

আমরা এখন একটি প্রোগ্রাম লিখে ফেলব, যেটি তোমার কম্পিউটারের স্ক্রিনে Hello World দেখাবে বা প্রিন্ট করবে। এটি হচ্ছে প্রোগ্রামিংয়ের একটি ঐতিহ্য। পৃথিবীর অধিকাংশ প্রোগ্রামারই জীবনের প্রথম প্রোগ্রাম হিসেবে এটি লেখে। আমি এই বইয়ের প্রোগ্রামগুলো চালানোর জন্য Codeblocks ব্যবহার করব। তবে তোমরা অন্য কিছু ব্যবহার করলেও কোনো সমস্যা নেই, সবগুলোতে কাজের ধারা মোটামুটি একই রকম। কম্পিউটারে কোডব্লক্স ইনস্টল করে ফেলো।

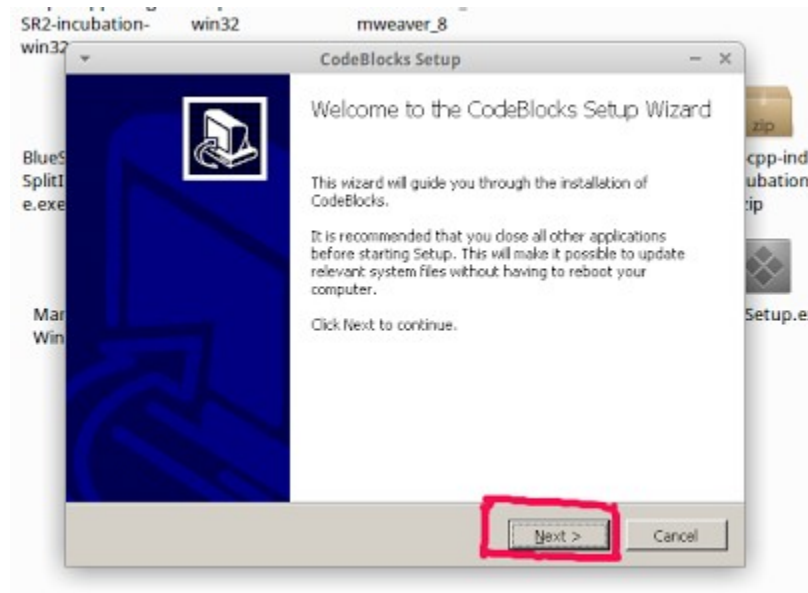
[codeblocks-10.05mingw-setup.exe](#) - [codeblocks-12.11mingw-setup.exe](#)

ডাউনলোডকৃত ফাইলটিতে ডাবল ক্লিক কর।

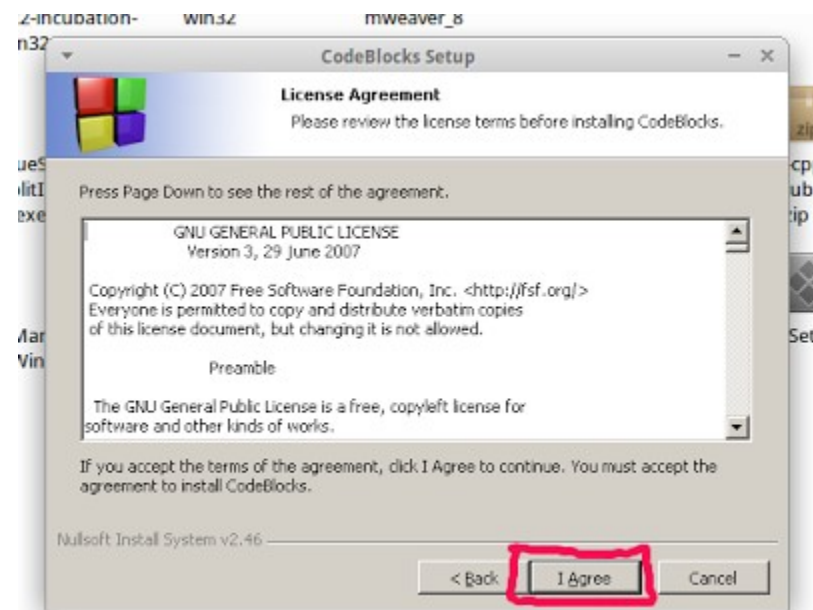


Setup window আসবে

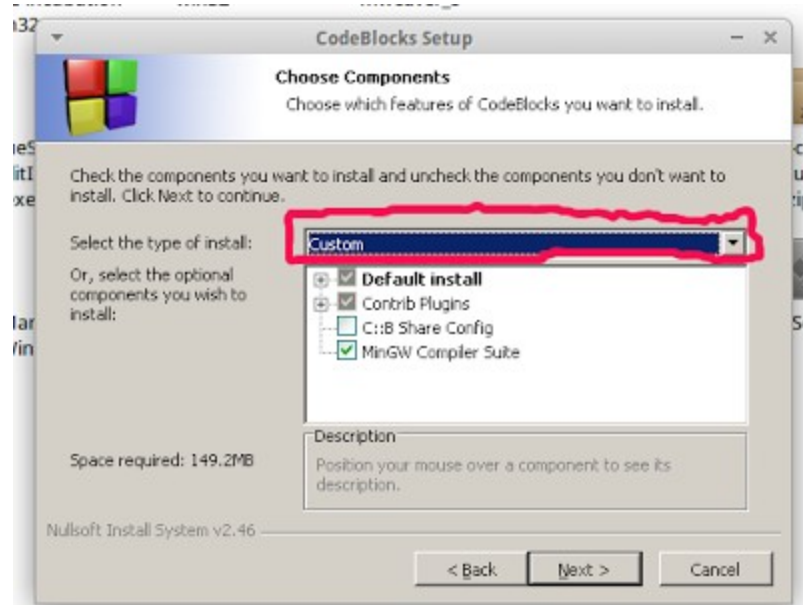
Next এ ক্লিক করুন



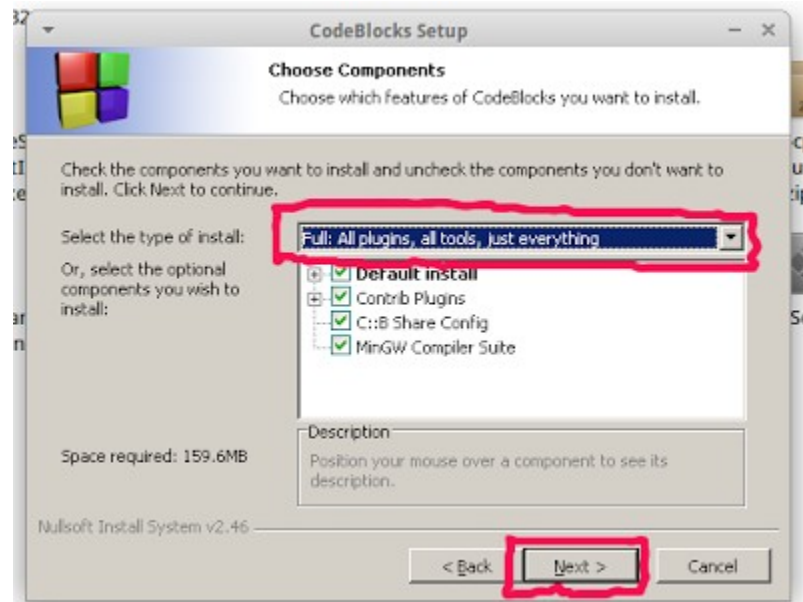
I Agree



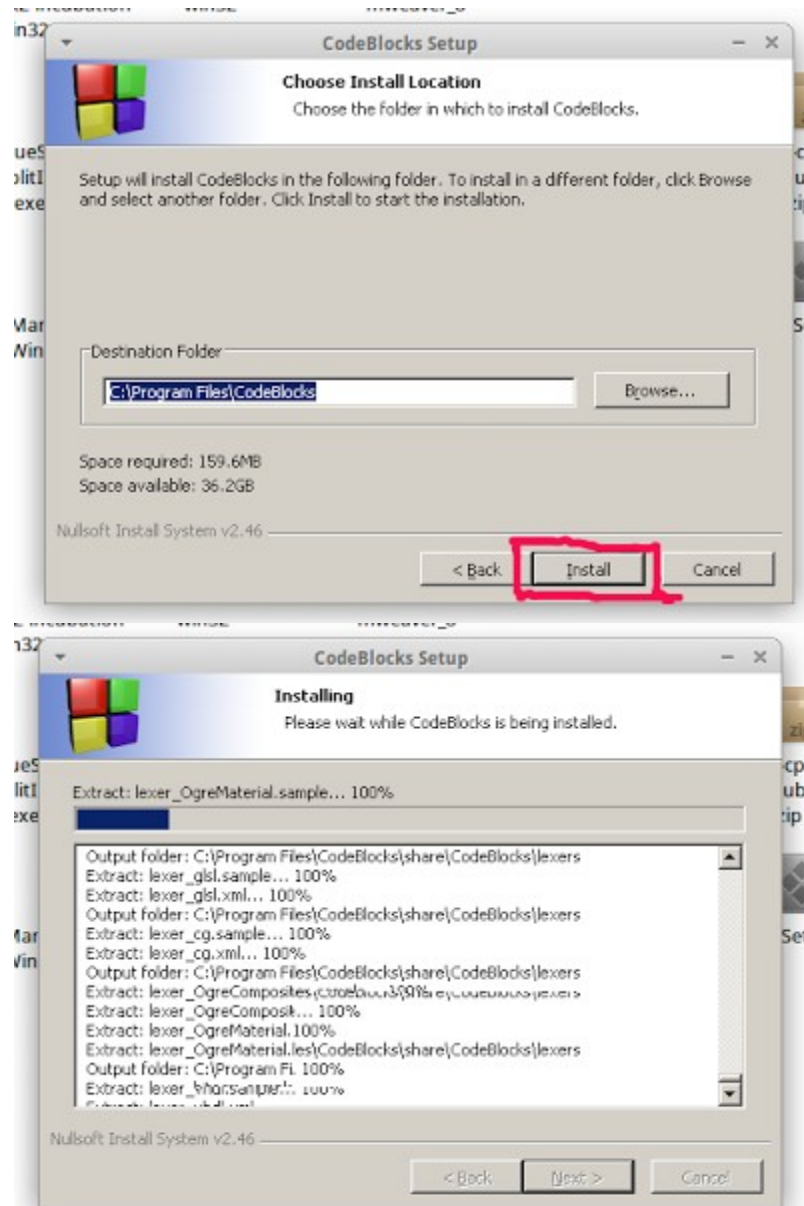
এবার যে উইন্ডোটি আসবে লক্ষ্য করুন , এখানে install type এ default ভাবে custom দেয়া আছে



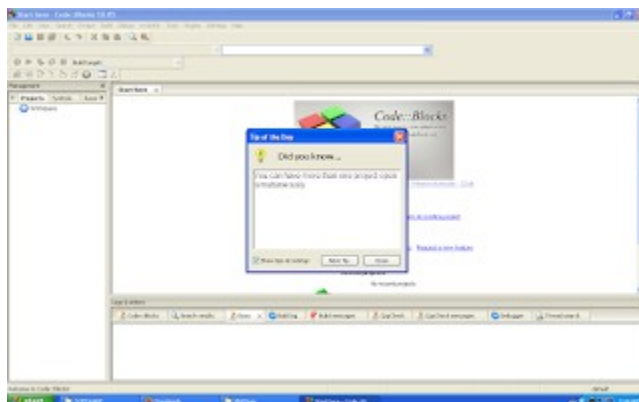
এটি পরিবর্তন করে Full: All plugins, all tools, just everything সিলেক্ট করে Next এ ক্লিক করুন



পরবর্তী উইন্ডোতে Install এ ক্লিক করুন , ইন্সটল হওয়া শুরু হবে

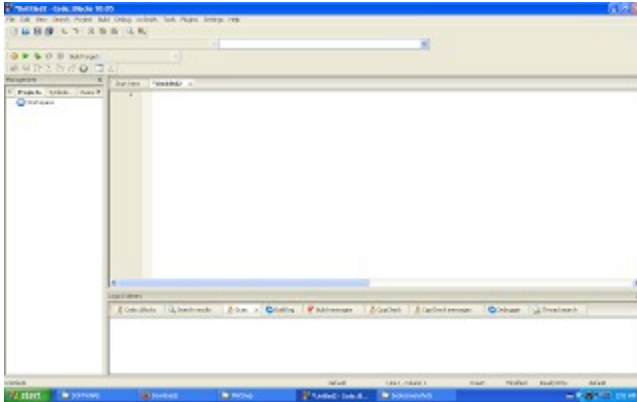


ইন্সটল হওয়া শেষ হলে কোডব্লকস রান করুন। নিচের ছবির মত আসবে



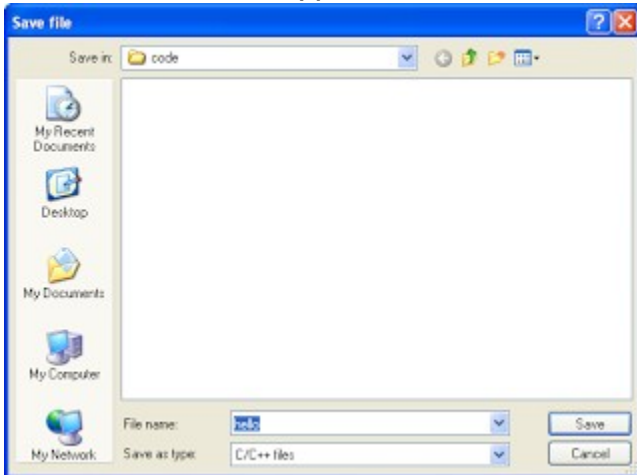
-->

এখানে তোমরা Show tips at startup চেকবক্সের টিক (tick) চিহ্নটি উঠিয়ে দিতে পারো।



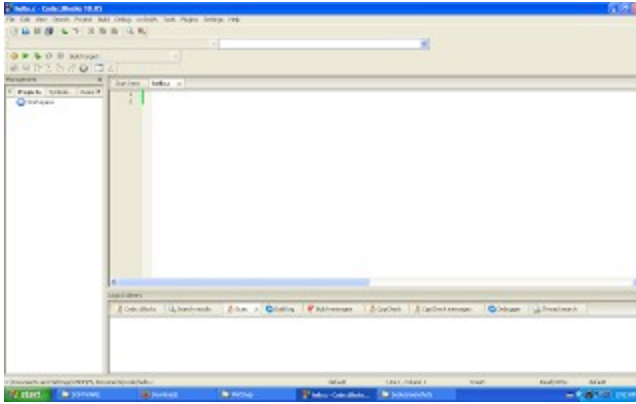
-->

এখন তোমরা প্রোগ্রামগুলো রাখার জন্য হার্ডডিস্কের ভেতর একটি ফোল্ডার তৈরি করে নাও। ওই ফোল্ডারে ফাইলগুলো সেভ (Save) করবে। ফাইলের যেকোনো একটি নাম দাও। আর Save as type হবে C/C++ files।



-->

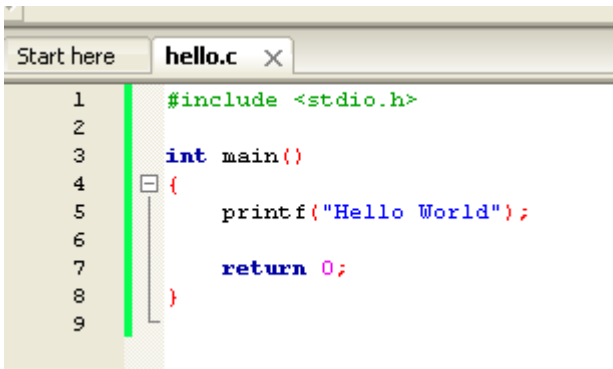
নিচের ছবিতে দেখো ফাইলের নাম হচ্ছে hello.c। সি প্রোগ্রামের সব ফাইলের এক্সটেনশন হবে .c।



-->

এখানে আমরা আমাদের কোড বা প্রোগ্রাম লিখব। নিচের কোডটি টাইপ করে ফেলো এবং ফাইলটি সেভ করো।

```
#include <stdio.h>
int main ()
{
    printf("Hello World");
    return 0;
}
প্রোগ্রাম: ১.১
```



-->

তোমরা হয়তো চিন্তা করছ, আমি এই হিজিবিজি কী লিখলাম? আস্তে ধীরে সব ব্যাখ্যা করব, চিন্তা নেই! আপাতত আমার কথামতো কাজ করে যাও। এবার Build মেনুতে গিয়ে Compile Current File-এ ক্লিক করো।



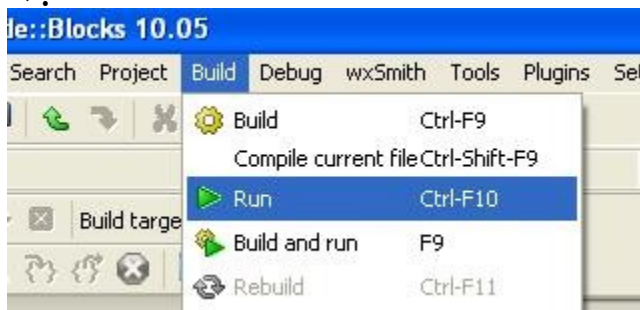
-->

তুমি যদি প্রোগ্রামটি ঠিকভাবে টাইপ করে থাকো তবে কম্পাইলার তোমাকে বলবে যে 0 errors, 0 warnings, মানে - প্রোগ্রামে syntax ঠিক আছে।

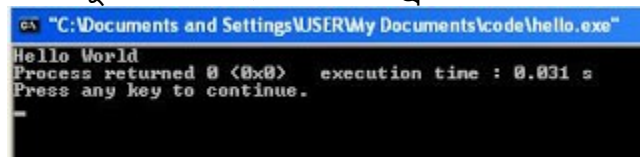


-->

এখন আবার Build মেনুতে গিয়ে Run-এ ক্লিক করো। তাহলে তোমার প্রোগ্রাম চালু হয়ে যাবে।



এবং তুমি নিচের ছবির মতো স্ক্রিন দেখতে পাবে।



-->

এখানে দেখো, তোমার প্রোগ্রামটি স্ক্রিনে Hello World প্রিন্ট করেছে। পরের লাইনে বলা আছে Process returned 0 (0x0) (এটির অর্থ নিয়ে আমাদের এখন মাথা না ঘামালেও চলবে) আর execution time : 0.031 s মানে প্রোগ্রামটি চলতে 0.031 সেকেন্ড সময় লেগেছে। তারপরের লাইন হচ্ছে, Press any key to continue. কি-বোর্ডে Any key খুঁজে না পেলে অন্য যেকোনো কি চাপলেই চলবে।

তুমি যদি প্রোগ্রামটি ঠিকঠাকভাবে রান করাতে পারো এবং Hello World লেখাটা দেখে থাকো তাহলে তোমাকে অভিনন্দন। তুমি বেশ গুরুত্বপূর্ণ একটি কাজ করে ফেলেছ।

এবারে দেখা যাক আমি কী লিখেছি কোডে।

প্রথম লাইন ছিল: `#include <stdio.h>`, এটি কেন লিখেছি একটু পরে বলছি।

দ্বিতীয় লাইন ফাঁকা। দেখতে সুন্দর লাগে তাই।

তৃতীয় লাইন: `int main()`। এটিকে বলে মেইন ফাংশন। সি প্রোগ্রামগুলো মেইন ফাংশন থেকে কাজ করা শুরু করে, তাই সব প্রোগ্রামে একটি (এবং কেবল একটি) মেইন ফাংশন থাকতে হয়। মেইন ফাংশনের শুরুতে দ্বিতীয় বন্ধনী দিয়ে শুরু করতে হয় আর শেষও করতে হয় একটি দ্বিতীয় বন্ধনী দিয়ে। শেষ করার আগে আমি `return 0;` লিখেছি, সেটি কেন এখন ব্যাখ্যা না করলেই ভালো হয়, ফাংশন নিয়ে যখন আলাপ করব তখন বলব। তাই আপাতত তোমরা যেকোনো প্রোগ্রামে নিচের অংশটুকু লিখে ফেলবে:

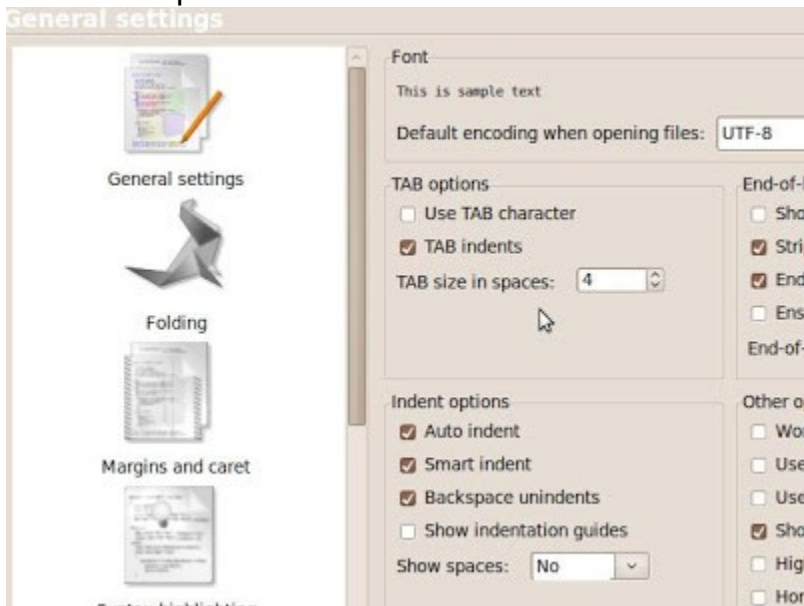
```
int main()
{
    এখানে কোড থাকবে।
    return 0;
}
```

প্রোগ্রামের পরের লাইন খেয়াল করো: `printf("Hello World");` এটি একটি স্টেটমেন্ট। এখানে `printf()` হচ্ছে একটি ফাংশন যার কাজ হচ্ছে স্ক্রিনে কিছু প্রিন্ট করা। ডবল কোটেশন চিহ্নের ভেতরে যা লিখবে তা-ই স্ক্রিনে সে প্রিন্ট করবে। এই ফাংশনটি স্ক্রিনে প্রিন্ট করে কীভাবে সেটি আসলে বলা আছে `stdio.h` নামে একটি ফাইলে। এই ফাইলগুলোকে বলে হেডার (header) ফাইল (.h হচ্ছে হেডার ফাইলের এক্সটেনশন)। `stdio.h` ফাইলে স্ট্যান্ডার্ড ইনপুট আর আউটপুট-সংক্রান্ত যাবতীয় ফাংশন লেখা আছে, আমরা কেবল সেগুলো ব্যবহার করব, ফাংশনগুলো কীভাবে কাজ করে সেটি এখন আমাদের জানার দরকার নেই। আর যেহেতু `printf()` ফাংশন ব্যবহার করেছি, তাই প্রোগ্রামের শুরুতে `#include <stdio.h>` লিখতে হয়েছে। এই রকম আরও অনেক প্রয়োজনীয় হেডার ফাইল আছে, যার কিছু আমরা পরবর্তী সময়ে কাজের প্রয়োজনে দেখব।

এখন একটি ব্যাপার খেয়াল করো। `printf("Hello World");`-এর শেষে একটি সেমিকোলন রয়েছে। সি ল্যাঙ্গুয়েজে প্রতিটি স্টেটমেন্টের পরেই একটি সেমিকোলন থাকে। একটি স্টেটমেন্টের কাজ শেষ হলে পরের স্টেটমেন্টের কাজ শুরু হয়। `return 0;`ও একটি স্টেটমেন্ট, তাই এটিও সেমিকোলন দিয়ে শেষ করতে হয়েছে। শুরুর দিকে

অনেকেই সেমিকোলন দিতে ভুলে যায়, তখন কম্পাইল এরর (compile error) হয়। তোমরা একটি সেমিকোলন মুছে দিয়ে কম্পাইল করার চেষ্টা করে দেখতে পারো।

এবারে একটি খুব গুরুত্বপূর্ণ কথা বলে রাখি। তোমরা কোডটি খেয়াল করলে দেখবে যে আমি `#include <stdio.h>`, `int main()`, `{` ও `}` যেই লাইনে আছে সেটি এডিটরের একেবারে বাঁ দিক থেকে শুরু করেছি। আর `printf` এবং `return 0`-এর আগে চারটি স্পেস (ফাঁকা জায়গা) দিয়ে নিয়েছি। এটিকে বলে ইন্ডেন্টেশন (Indentation)। এরকম না করলেও প্রোগ্রামটি চলত এবং তাই অনেকেই ইন্ডেন্টেশনের ব্যাপারটি গুরুত্ব দেয় না এবং ঠিকমতো ইন্ডেন্টেশন করে না। যেকোনো ভালো অভ্যাসের মতো ইন্ডেন্টেশনের অভ্যাস তৈরি করাটা একটু কঠিন, তবে বিষয়টা কিছু দাঁত মাজার মতোই গুরুত্বপূর্ণ। ইন্ডেন্টেশন করার অভ্যাস ঠিকমতো তৈরি না হলে প্রোগ্রামারদের সহকর্মী বা বসের বকা শুনতে হয়, অনেক জায়গায় তো ইন্টারভিউতেই বাদ পড়ে যেতে হয়। আশা করছি তোমরা ব্যাপারটি বেশ গুরুত্ব সহকারে নেবে। আমি বইয়ের সমস্ত উদাহরণেই যথাযথভাবে ইন্ডেন্টেশন করার চেষ্টা করব তবে ছাপার সময় একটু এদিক-ওদিক হতে পারে, সেটি তোমরা বুঝে নেবে। ইন্ডেন্টেশনের জন্য সাধারণত চারটি স্পেস দেওয়াটাই এখন স্ট্যান্ডার্ড। তোমরা এডিটরে অপশন সেট করতে পারো যাতে ট্যাব (Tab) চাপলে সেটি চারটি স্পেসের সমান হয়। Codeblocks-এ Settings মেনুতে Editor-এ ক্লিক করে TAB Options-এ TAB indents চেক করো এবং TAB size in spaces 4 দাও।

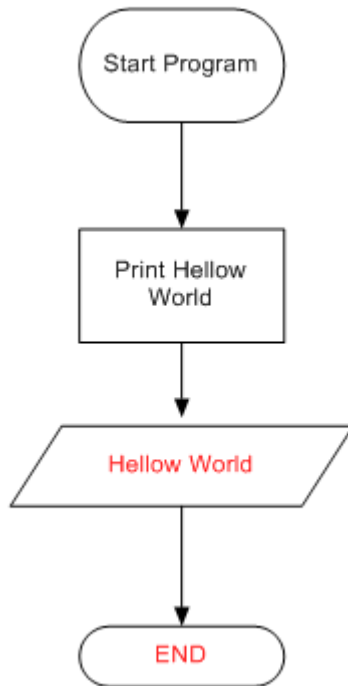


-->

এবারে তোমাদের জন্য একটি কাজ। একটি প্রোগ্রাম লেখো যেটি স্ক্রিনে প্রিন্ট করবে: I love my country, Bangladesh।

প্রোগ্রামটি টাইপ করার পরে অবশ্যই কম্পাইল ও রান করবে। কম্পাইল করার আগে সেভ করতে ভুলবে না।

অধ্যায় ৩য়:- Concept Of Flowchart & Algorithms



Algorithms হলো তোমার প্রোগ্রামের পূর্ব পরিকল্পনার সমষ্টি। অর্থাৎ তুমি যে সফটওয়্যার বানাতে চাচ্ছ তার প্রোসেসিং এর ধাপ সমূহ। মনে কর তুমি ২ টি সংখ্যার যোগফল নির্ণয় করতে চাচ্ছ তাহলে তোমাকে আগে ভেরিয়ারবল ডিক্লেয়ার করে এদের মান বসাতে হবে এবং যোগফল নির্ণয় করতে হবে। এবার এই ধারণা থেকে তুমি একটা চিত্র একে ফেল এটাই ফ্লোচার্ট। বাহ!! হয়ে গেলো। আচ্ছা এবার আরো একটু বিস্তারিত জানা যাক।

Flowchart হল একটা decision সাপোর্ট সিস্টেম। এই টা একটা ডায়াগ্রাম যা ব্যবহার করে আমরা আমাদের সমস্যা সমাধান টা ধাপে ধাপে শেষ করতে পারি।

বড় বড় প্রোগ্রাম লেখার সময় এই ফ্লো চার্ট এর মাধ্যমেই আপনাকে আপনার এলগরিদম ডিসাইন করতে হবে। ধাপে ধাপে সমস্যার সমাধান করার জন্য ফ্লো চার্ট খুব খুব গুরুত্বপূর্ণ।

উপরে দেখছ ফ্লোচার্ট এর একটি উদাহরণ।

ফ্লোচার্ট এর মধ্যে যেসব বক্স বা শেপ আপনারা দেখছেন এগুলার জন্য নির্দিষ্ট অর্থ আমাদের উল্লেখ করে দিতে হবে।

প্রথম সেপ টি আমরা আমাদের ফ্লো চার্ট এর শুরু এবং শেষ দেখানোর জন্য ব্যবহার করি।

আয়াতকার অথবা বর্ণাকার সেপ টি আমরা ব্যবহার করি প্রসেস দেখানোর জন্য। আপনি যা করছেন তা দেখানোর জন্য, আমাদের উদাহরণ দেখুন write advertise ,draw graphics , send advertise এগুলো প্রসেস।

ডায়ামন্ড এর মত সেপটি আমরা ব্যবহার করি ডিসিশন নেবার জন্য।
কন্ডিশন সত্য না মিথ্যা তা যাচাই করাই এর কাজ।

আর এই তীর চিহ্ন ব্যবহার করা হয় ইনফরমেশন এর দিক দেখানোর জন্য।

ফ্লো চার্ট এ ইনফরমেশন এর ফ্লো সাধারণত উপর থেকে নিচে অথবা বাম থেকে ডান দিকে যায়।

এখন আমাদের উদাহরণটি একটু ব্যাখ্যা করি। প্রথমেই শুরু করা হয়েছে বিজ্ঞাপনটির প্লান করুন - - > বিজ্ঞাপনটি লিখুন -- > এখন আপনি কি গ্রাফিক্স যোগ করবেন ? উত্তর হ্যা হলে গ্রাফিক্স যোগ করুন-- > তারপর বিজ্ঞাপনটি পাঠিয়ে দিন।

আর ঐ প্রশ্নের উত্তর না হলে সরাসরি গ্রাফিক্স ছারাই বিজ্ঞাপনটি পাঠিয়ে দিন।

বিজ্ঞাপন পাঠানোর পর বিজ্ঞাপন এর জন্য আমাদের কাজ সম্পন্ন হল।



এতে আমার মনে হয় ভালো ধারণা হয়ে গেছে।

অধ্যায় ৪র্থ ডাটা টাইপ, ইনপুট ও আউটপুট এবং ভেরিয়েবলস

এ অধ্যায়ে আমরা কিছু ছোট ছোট প্রোগ্রাম লিখব। সবগুলো প্রোগ্রাম অবশ্যই কম্পিউটারে চালিয়ে দেখবে এবং একটু পরিবর্তন করে কম্পাইল ও রান করার চেষ্টা করবে।

আমাদের প্রথম প্রোগ্রামটি হবে দুটি সংখ্যা যোগ করার প্রোগ্রাম। এখন কথা হচ্ছে, সংখ্যাগুলো তো কম্পিউটারের মেমোরিতে রাখতে হবে, সেই জটিল কাজটি কীভাবে করব? চিন্তা নেই! সব প্রোগ্রামিং ল্যাঙ্গুয়েজে ভেরিয়েবল বলে একটি জিনিস আছে যেটি কোন নির্দিষ্ট মান ধারণ করার জন্য ব্যবহার করা হয়। ভেরিয়েবলের একটি নাম দিতে হয়, তারপর ভেরিয়েবল = কোনো মান লিখে দিলে ভেরিয়েবলের ভেতর সেটি ঢুকে যায়। এটির সঙ্গে গাণিতিক সমীকরণের কিন্তু কোনো সম্পর্ক নেই। চলো, প্রোগ্রামটি লিখে রান করাই, তারপর ব্যাখ্যা করা যাবে।

```
#include <stdio.h>
int main()
{
    int a;
    int b;
    int sum;
    a = 50;
    b = 60;
    sum = a + b;
    printf("Sum is %d", sum);
    return 0;
}
```

প্রোগ্রাম: ২.১

প্রোগ্রামটি রান করাও, তুমি স্ক্রিনে দেখবে: Sum is 110।

এখানে a, b, sum তিনটি ভেরিয়েবল (variable) আলাদা সংখ্যা ধারণ করে। প্রথমে আমাদের বলে দিতে হবে যে a, b, sum নামে তিনটি ভেরিয়েবল আছে। এবং এগুলোতে কী ধরনের ডাটা থাকবে সেটিও বলে দিতে হবে। int a; দিয়ে আমরা কম্পাইলারকে বলছি a নামে একটি ভেরিয়েবল এই প্রোগ্রামে আছে যেটি একটি পূর্ণসংখ্যা (integer)-এর মান ধারণ করার জন্য ব্যবহার করা হবে। এই কাজটিকে বলে ভেরিয়েবল ডিক্লারেশন। আর int হচ্ছে ডাটা টাইপ, যেটি দেখে সি-এর কম্পাইলার বুঝবে যে এতে ইন্টিজার টাইপ ডাটা থাকবে। আরও বেশ কিছু ডাটা টাইপ আছে, সেগুলো আমরা আশ্বে আশ্বে দেখব। আমরা চাইলে একই টাইপের ভেরিয়েবলগুলো ডিক্লেয়ার করার সময় আলাদা লাইনে না লিখে একসঙ্গে কমা দিয়েও লিখতে পারতাম, যেমন: int a, b,

sum;। আর লক্ষ করো যে ভেরিয়েবল ডিক্লারেশনের শেষে সেমিকোলন ব্যবহার করতে হয়।

এরপর আমি দুটি স্টেটমেন্ট লিখেছি:

a = 50;

b = 60;

এখানে a-এর মান 50 আর b-এর মান 60 বলে দিলাম (assign করলাম), যতক্ষণ না এটি আমরা পরিবর্তন করছি, কম্পাইলার a-এর মান 50 আর b-এর মান 60 ধরবে।

পরের স্টেটমেন্ট হচ্ছে: sum = a + b;। এতে বোঝায়, sum-এর মান হবে a + b-এর সমান, অর্থাৎ a ও b-এর যোগফল যে সংখ্যাটি হবে সেটি আমরা sum নামের ভেরিয়েবলে রেখে দিলাম (বা assign করলাম)।

এবারে যোগফলটি মনিটরে দেখাতে হবে, তাই আমরা printf ফাংশন ব্যবহার করব।
printf("Sum is %d", sum);

এখানে দেখো printf ফাংশনের ভেতরে দুটি অংশ। প্রথম অংশে "Sum is %d" দিয়ে বোঝানো হয়েছে স্ক্রিনে প্রিন্ট করতে হবে Sum is এবং তার পরে একটি ইন্টিজার ভেরিয়েবলের মান যেটি %d-এর জায়গায় বসবে। তারপর কমা দিয়ে আমরা sum লিখে বুঝিয়ে দিয়েছি যে %d-তে sum-এর মান প্রিন্ট করতে হবে। ইন্টিজারের জন্য যেমন %d ব্যবহার করলাম, অন্য ধরনের ভেরিয়েবলের জন্য অন্য কিছু লিখতে হবে, যেটি আমরা ব্যবহার করতে করতে শিখব। খুব ভালো হতো যদি আমি এখন একটি চার্ট লিখে দিতাম যে সি ল্যাঙ্গুয়েজে কী কী ডাটা টাইপ আছে, সেগুলো কী দিয়ে লেখে এবং প্রিন্ট করার জন্য কী করতে হবে আর তোমরা সেই চার্ট মুখস্থ করে ফেলতে। কিন্তু শুধু শুধু মুখস্থ করার কোনো দরকার নেই, মুখস্থ করার প্রবণতা চিন্তাশক্তি কমায় আর প্রোগ্রামারদের জন্য চিন্তা করার ক্ষমতা খুবই গুরুত্বপূর্ণ।

আমরা ওপরের প্রোগ্রামটি চাইলে এভাবেও লিখতে পারতাম:

```
#include <stdio.h>
int main()
{
    int a, b, sum;
    a = 50;
    b = 60;
    sum = a + b;
    printf("Sum is %d", sum);
    return 0;
```



```
}
```

প্রোগ্রাম: ২.২

আবার ভেরিয়েবল ডিক্লেয়ার করার সময় একই সঙ্গে অ্যাসাইনও করা যায়:

```
#include <stdio.h>
int main()
{
    int a = 50, b = 60, sum;
    sum = a + b;
    printf("Sum is %d", sum);
    return 0;
}
```

প্রোগ্রাম: ২.৩

এখন তোমাদের জন্য একটি প্রশ্ন। নিচের প্রোগ্রামটির আউটপুট কী হবে?

```
#include <stdio.h>
int main()
{
    int x, y;
    x = 1;
    y = x;
    x = 2;
    printf("%d", y);
    return 0;
}
```

প্রোগ্রাম: ২.৪

কী মনে হয়? আউটপুট 1 নাকি 2? আউটপুট হবে 1, কারণ প্রথমে কম্পাইলার দেখবে, x-এর মান 1 অ্যাসাইন করা হয়েছে (x = 1;)। তারপর x-এর মানটি আবার y-এ অ্যাসাইন করা হয়েছে (y = x;)। এখন y-এর মান 1। তারপর আবার x-এর মান 2 অ্যাসাইন করা হয়েছে। কিন্তু তাতে y-এর মানের কোনো পরিবর্তন হবে না। প্রোগ্রামিংয়ে y = x; আসলে কোনো সমীকরণ না যে এটি সবসময় সত্য হবে। '=' চিহ্ন দিয়ে একটি ভেরিয়েবলে নির্দিষ্ট কোনো মান রাখা হয়।

এবারে নিচের প্রোগ্রামটি দেখো:

```
#include <stdio.h>
int main()
{
    int a = 50, b = 60, sum;
    sum = a + b;
    printf("%d + %d = %d", a, b, sum);
    return 0;
}
```

প্রোগ্রাম: ২.৫

প্রোগ্রামটি মনিটরে কী প্রিন্ট করে? চালিয়ে দেখো। `printf("%d + %d = %d", a, b, sum);` না লিখে `printf("%d + %d = %d", b, a, sum);` লিখে প্রোগ্রামটি আবার চালাও। এখন জিনিসটি চিন্তা করে বুঝে নাও।

লেখাপড়া করার সময় আমাদের মনে নানা বিষয়ে নানা প্রশ্ন আসে, যার উত্তর আমরা বইতে খুঁজি, শিক্ষককে জিজ্ঞাসা করি, ইন্টারনেটে খুঁজি বা চিন্তা করে যুক্তি দাঁড় করিয়ে উত্তরটি বের করি। আমাদের দুর্ভাগ্য যে বেশিরভাগ ছেলেমেয়েই শেষ কাজটি করে না, কারণ নিজে নিজে চিন্তা করতে একটু সময় লাগে ও পরিশ্রম হয়, সেই সময় আর শ্রম তারা দিতে চায় না। আর আমাদের অভিভাবক, শিক্ষক ও শিক্ষাব্যবস্থা চিন্তা করার জন্য কোনো পুরস্কার দেয় না, বরং মুখস্থ করার জন্যই পুরস্কৃত করে।

যা-হোক, প্রোগ্রামিংয়ের ব্যাপারে যখনই মনে কোনো প্রশ্ন আসবে, সঙ্গে সঙ্গে একটি প্রোগ্রাম লিখে ফেলবে। দেখো তোমার কম্পাইলার কী বলে। ধরা যাক, আমরা যদি `int` টাইপের ভেরিয়েবলে দশমিক যুক্ত সংখ্যা (বাস্তব সংখ্যা বা `real number`) ব্যবহার করতাম, তাহলে কী হতো?

```
#include <stdio.h>
int main()
{
    int a = 50.45, b = 60, sum;
    sum = a + b;
    printf("%d + %d = %d", a, b, sum);
    return 0;
}
```

প্রোগ্রাম: ২.৬

এখানে `a`-এর মান `50.45` ব্যবহার করলাম। এবারে প্রোগ্রাম চালাও, দেখো কী হয়। আবার মনে যদি প্রশ্ন আসে যে, `main` ফাংশনের শেষ লাইনে `return 0;` না লিখলে কী হয়? তাহলে `return 0;` ছাড়া প্রোগ্রাম চালিয়ে দেখো কী হয়।

আউটপুট হবে: `50 + 60 = 110।`

সি কম্পাইলার `a`-এর মান `50` ধরেছে, যদিও আমরা `50.45` অ্যাসাইন করেছি। এই ব্যাপারটিকে বলে টাইপ কাস্ট (type cast)। বাস্তব সংখ্যা রাখার জন্য সি ভাষায় `double` নামের ডাটা টাইপ রয়েছে। টাইপ কাস্ট করে `double` সংখ্যাটিকে `int`-এ নেওয়া হয়েছে, এটি অটোমেটিক হয়। আবার কম্পাইলারকে বলেও দেওয়া যায়: `int a = (int) 50.45।`

int a = 50.99; এখানে a-এর মান হবে 50। int a = -50.9; লিখলে a-এর মান হয় -50। এক কথায় বললে double থেকে int-এ টাইপ কাস্ট করলে দশমিকের পরের অংশটি বাদ পড়ে যাবে।

আরেকটি কথা। যেই ভেরিয়েবলকে টাইপ কাস্ট করা হচ্ছে, তার মান কিন্তু পরিবর্তন হয় না। টাইপ কাস্ট করা মানটি একটি ভেরিয়েবলে রাখা যায়। নিচের প্রোগ্রামটি কম্পিউটারে চালালেই বুঝতে পারবে।

```
#include <stdio.h>
int main()
{
    int n;
    double x;
    x = 10.5;
    n = (int)x;
    printf("Value of n is %d\n", n);
    printf("Value of x is %lf\n", x);
    return 0;
}
```

প্রোগ্রাম: ২.৭

প্রোগ্রামের আউটপুট দেখো। x-এর মান কিন্তু পরিবর্তন হয়নি। আর বুঝতেই পারছ যে বাস্তব সংখ্যা রাখার জন্য সি-তে যে double টাইপের ভেরিয়েবল ব্যবহার করা হয়, তা প্রিন্ট করার সময় %lf (I এখানে ইংরেজি ছোট হাতের L) ব্যবহার করতে হয়।

int ডাটা টাইপে তো কেবল পূর্ণ সংখ্যা রাখা যায়। কিন্তু সেটি কী যেকোনো পূর্ণসংখ্যা? উত্তরের জন্য একটি প্রোগ্রাম লিখি:

```
#include <stdio.h>
int main()
{
    int a;
    a = 1000;
    printf("Value of a is %d", a);
    a = -21000;
    printf("Value of a is %d", a);
    a = 10000000;
    printf("Value of a is %d", a);
    a = -10000000;
    printf("Value of a is %d", a);
    a = 100020004000503;
    printf("Value of a is %d", a);
    a = -4325987632;
    printf("Value of a is %d", a);
}
```

```
return 0;  
}
```

প্রোগ্রাম: ২.৮

এখানে আমরা a-তে বিভিন্ন সংখ্যা অ্যাসাইন করলাম। সব মান কি ঠিকঠাক আসছে? আসেনি। কেন আসেনি সেটি ব্যাখ্যা করার আগে একটি কথা বলে নিই। পরপর এতগুলো printf-এর কারণে তোমার কম্পিউটারের স্ক্রিনে নিশ্চয়ই দেখতে একটু অস্বস্তিকর লাগছে। প্রতিটি printf তোমরা এভাবে লিখতে পারো: printf("Value of a is %d\n", a);। এখন printf ফাংশনে ""-এর ভেতর \n লিখলে কী হয় সেটি আমি বলব না। প্রোগ্রামটি চালালেই বুঝতে পারবে।

a-এর সবগুলো মান কিন্তু ঠিকভাবে দেখা যায়নি, যেসব মান -2147483648 থেকে 2147483647 পর্যন্ত কেবল সেগুলোই ঠিকঠাক প্রিন্ট হবে, কারণ এই রেঞ্জের বাইরের সংখ্যা int টাইপের ভেরিয়েবলে রাখা যায় না। এটি হলো int টাইপের সংখ্যার সীমা। সিনে int টাইপের ডাটার জন্য মেমোরিতে চার বাইট (byte) জায়গা ব্যবহৃত হয়। চার বাইট মানে বত্রিশ বিট (1 byte = 8 bit)। প্রতি বিটে দুটি জিনিস রাখা যায়, 0 আর 1। দুই বিটে রাখা যায় চারটি সংখ্যা (00, 01, 10, 11)। তাহলে 32 বিটে রাখা যাবে: 2^{32} টা সংখ্যা অর্থাৎ 4294967296 টি সংখ্যা। এখন অর্ধেক ধনাত্মক আর অর্ধেক ঋণাত্মক যদি রাখি, তাহলে -2147483648 থেকে -1 পর্যন্ত মোট 2147483648 সংখ্যা আবার 0 থেকে 2147483647 পর্যন্ত মোট 2147483648 টি সংখ্যা, সর্বমোট 4294967296 টি সংখ্যা। আশা করি, হিসাবটা বুঝতে পেরেছ।

এখন আমরা যোগ করার প্রোগ্রামটি লিখব যেটি সব বাস্তব সংখ্যা (real number) যোগ করতে পারবে। তোমাদের মনে করিয়ে দিই, পূর্ণসংখ্যা হচ্ছে, ... -3, -2, -1, 0, 1, 2, 3 ... ইত্যাদি। আর বাস্তব সংখ্যা হচ্ছে -5, -3, -2.43, 0, 0.49, 2.92 ইত্যাদি (সংখ্যারেখার ওপর সব সংখ্যাই কিন্তু বাস্তব সংখ্যা)।

```
#include <stdio.h>  
int main()  
{  
    double a, b, sum;  
    a = 9.5;  
    b = 8.743;  
    sum = a + b;  
    printf("Sum is: %lf\n", sum);  
    printf("Sum is: %0.2lf\n", sum);  
    return 0;  
}
```

প্রোগ্রাম: ২.৯

প্রোগ্রামটি কম্পাইল এবং রান করো। আউটপুট হবে নিচের দুই লাইন:

Sum is: 18.243000

Sum is: 18.24

%lf ব্যবহার করায় প্রথম লাইনে দশমিকের পরে ছয় ঘর পর্যন্ত প্রিন্ট হয়েছে। আবার দ্বিতীয় লাইনে দশমিকের পরে দুই ঘর পর্যন্ত প্রিন্ট হয়েছে, কারণ %0.2lf লিখেছি (তিন ঘর পর্যন্ত প্রিন্ট করতে চাইলে %0.3lf লিখতাম, আবার দশমিক অংশ প্রিন্ট করতে না চাইলে %0.0lf)। double টাইপের ডাটার জন্য 64 বিট ব্যবহৃত হয় এবং $1.7E-308$ (1.7×10^{-308}) থেকে $1.7E+308$ (1.7×10^{308}) পর্যন্ত ডাটা রাখা যায়। বিস্তারিত হিসাব বুঝতে হলে কম্পিউটার বিজ্ঞানসংক্রান্ত আরও কিছু জ্ঞানবুদ্ধির দরকার, তাই আমি আর এখন সেদিকে যাচ্ছি না।

এখন আমরা আমাদের প্রোগ্রামে এমন ব্যবস্থা রাখতে চাই, যাতে কোন দুটি সংখ্যা যোগ করতে হবে সেটি আমরা কোডের ভেতর লিখব না, ব্যবহারকারীর কাছ থেকে ইনপুট আকারে জেনে নেব। ব্যবহারকারীর (মানে যে প্রোগ্রামটি ব্যবহার করছে) কাছ থেকে ইনপুট নেওয়ার জন্য আমরা scanf ফাংশন ব্যবহার করব (সি-তে আরও ফাংশন আছে এই কাজের জন্য)। তাহলে দেরি না করে প্রোগ্রাম লিখে ফেলি:

```
#include <stdio.h>
int main()
{
    int a, b, sum;
    scanf("%d", &a);
    scanf("%d", &b);
    sum = a + b;
    printf("Sum is: %d\n", sum);
    return 0;
}
```

প্রোগ্রাম: ২.১০

প্রোগ্রামটি রান করলে দেখবে ফাঁকা স্ক্রিন (blank screen) আসে। তখন তুমি একটি সংখ্যা লিখবে, তারপর স্পেস (space) বা এন্টার (enter) দিয়ে আরেকটি সংখ্যা লিখবে। তারপর আবার এন্টার চাপলে যোগফল দেখতে পাবে।

তোমরা নিশ্চয়ই scanf ফাংশনের ব্যবহার শিখে ফেলেছ। scanf("%d", &a); এখানে ডবল কোটেশনের ভেতরে %d দিয়ে scanf-কে বলে দেওয়া হচ্ছে যে একটি ইন্টিজার বা int টাইপের ভেরিয়েবলের মান পড়তে হবে (ব্যবহারকারী কিবোর্ড থেকে ইনপুট দেবে)। আর দেখো a-এর আগে এমপারসেন্ড (&) চিহ্ন ব্যবহার করা হয়েছে, &a দিয়ে বোঝানো হয়েছে যে সংখ্যাটি ইনপুট দেওয়া হবে সেটি a ভেরিয়েবলের মাঝে অ্যাসাইন হবে। তোমরা যখন সি আরেকটু ভালোভাবে শিখবে, তখন &a-এর প্রকৃত অর্থ বুঝতে পারবে, আপাতত আমরা ব্যবহারের দিকেই মনোযোগ দিই। a এবং b-এর মান একটি scanf ফাংশন দিয়েও নেওয়া যেত এভাবে: scanf("%d %d", &a, &b);। ভেরিয়েবলের আগে & চিহ্ন না দিলে কী সমস্যা? নিচের প্রোগ্রামটি রান করার চেষ্টা করো, কিছু একটি এরর পাবে। এই মুহূর্তে এররটা ব্যাখ্যা করছি না, কারণ ব্যাখ্যাটা

একটু জটিল আর এখন বোঝাতে গেলে তোমরা ভুল বুঝতে পারো এবং পরে আমাদের গালমন্দ করবে।

```
#include <stdio.h>
int main()
{
    int a, b, sum;
    scanf("%d", &a);
    scanf("%d", &b);
    sum = a + b;
    printf("Sum is: %d\n", sum);
    return 0;
}
```

প্রোগ্রাম: ২.১১

এখন আমরা যদি ইনপুট হিসেবে ইন্টিজার না নিয়ে ডবল টাইপের ডাটা নিতে চাইতাম তাহলে কী করতে হতো? scanf-এ %d-এর বদলে %lf ব্যবহার করলেই চলত। তোমরা প্রোগ্রামটি লিখে ফেলো এবং দেখো ঠিকঠাক রান হয় কি না। তারপরে বাকি অংশ পড়া শুরু করো।

আসলে ঠিকঠাক রান হবে না, কারণ ডাটা টাইপও পরিবর্তন করতে হবে। মানে int না লিখে double লিখতে হবে। প্রোগ্রামটি ঠিক করে আবার চালাও।

বইতে যখনই আমি কোনো প্রোগ্রাম লেখতে বলব সেটি যত সহজ কিংবা কঠিনই মনে হোক না কেন, সেটি কম্পিউটারে লিখে কম্পাইল ও রান করতে হবে। এ কাজ না করে সামনে আগানো যাবে না। মনে রাখবে, গাড়ি চালানো শেখার জন্য যেমন গাড়ি চালানোর কোনো বিকল্প নেই, সাঁতার শেখার জন্য যেমন সাঁতার কাটার বিকল্প নেই, তেমনই প্রোগ্রামিং শেখার জন্য প্রোগ্রামিং করার কোনো বিকল্প নেই, শুধু বই পড়ে প্রোগ্রামার হওয়া যায় না।

এবারে আমরা আরেক ধরনের ডাটা টাইপ দেখব, সেটি হচ্ছে char (character) টাইপ। তো এই character টাইপের চরিত্র হচ্ছে একে মেমোরিতে রাখার জন্য মাত্র এক বাইট জায়গার দরকার হয়। সাধারণত যেকোনো অক্ষর বা চিহ্ন রাখার জন্য এই টাইপের ডাটা ব্যবহার করা হয়। তবে সেই অক্ষরটা ইংরেজি বর্ণমালার অক্ষর হতে হবে, অন্য ভাষার অক্ষর char টাইপের ভেরিয়েবলে রাখা যাবে না। নিচের প্রোগ্রামটি কম্পিউটারে লিখে রান করাও:

```
#include <stdio.h>
int main()
{
    char ch;
    printf("Enter the first letter of your name: ");
    scanf("%c", &ch);
}
```



```
printf("The first letter of your name is: %c\n", ch);
return 0;
}
```

প্রোগ্রাম: ২.১২

কোড দেখে বুঝতেই পারছ, char টাইপের জন্য printf এবং scanf ফাংশনের ভেতরে %c ব্যবহার করতে হয়। আরেকটি ফাংশন আছে getchar, এটি দিয়েও char টাইপের ডাটা রিড করা যায়। নিচের প্রোগ্রামটি দেখো:

```
#include <stdio.h>
int main()
{
    char ch;
    printf("Enter the first letter of your name: ");
    ch = getchar();
    printf("The first letter of your name is: %c\n", ch);
    return 0;
}
```

প্রোগ্রাম: ২.১৩

এটি রান করাও। আগের প্রোগ্রামটির মতো একই কাজ করবে। getchar ফাংশন একটি অক্ষর পড়ে সেটি ch ভেরিয়েবলের ভেতরে অ্যাসাইন করে দিল। আর সরাসরি কোনো কিছু char টাইপ ভেরিয়েবলে রাখতে চাইলে যেই অক্ষর বা চিহ্ন রাখবে তার দুই পাশে সিঙ্গেল কোটেশন চিহ্ন দেবে। যেমন: char c = 'A';

এখন নিচের প্রোগ্রামটি দেখো:

```
#include <stdio.h>
int main()
{
    int num1, num2;
    printf("Please enter a number: ");
    scanf("%d", &num1);
    printf("Please enter another number: ");
    scanf("%d", &num2);
    printf("%d + %d = %d\n", num1, num2, num1+num2);
    printf("%d - %d = %d\n", num1, num2, num1-num2);
    printf("%d * %d = %d\n", num1, num2, num1*num2);
    printf("%d / %d = %d\n", num1, num2, num1/num2);
    return 0;
}
```

প্রোগ্রাম: ২.১৪

এটি কম্পাইল ও রান করাও। এটি দেখে নিশ্চয়ই বুঝতে পারছ বিয়োগ, গুণ ও ভাগের কাজ কীভাবে করতে হয়। এবারে তোমাদের কাজ হচ্ছে চারটি। এক, num1 ও num2-এর মধ্যকার যোগ, বিয়োগ, গুণ, ভাগের কাজটি printf ফাংশনের ভেতরে না করে

আগে করা এবং মানটি অন্য একটি ভেরিয়েবলে রেখে দেওয়া। এর জন্য একটি প্রোগ্রাম লিখে ফেলো। দ্বিতীয় কাজ হচ্ছে প্রোগ্রামটি ডবল টাইপের ভেরিয়েবল ব্যবহার করে। তৃতীয় কাজ হচ্ছে, num2-এর মান 0 দিয়ে দেখো কী হয়। চতুর্থ কাজটি হচ্ছে printf ফাংশনের ভেতরে ডবল কোটেশনের ভেতরে যেই +, -, *, / চিহ্ন আছে সেগুলো সরাসরি ব্যবহার না করে একটি char টাইপ ভেরিয়েবলে রেখে তারপর ব্যবহার করা। চারটি কাজ ঠিকমতো করার পরে নিচের প্রোগ্রামটি দেখো:

```
#include <stdio.h>
int main()
{
    int num1, num2, value;
    char sign;
    printf("Please enter a number: ");
    scanf("%d", &num1);
    printf("Please enter another number: ");
    scanf("%d", &num2);
    value = num1 + num2;
    sign = '+';
    printf("%d %c %d = %d\n", num1, sign, num2, value);
    value = num1 - num2;
    sign = '-';
    printf("%d %c %d = %d\n", num1, sign, num2, value);
    value = num1 * num2;
    sign = '*';
    printf("%d %c %d = %d\n", num1, sign, num2, value);
    value = num1 / num2;
    sign = '/';
    printf("%d %c %d = %d\n", num1, sign, num2, value);
    return 0;
}
```

প্রোগ্রাম: ২.১৫

প্রোগ্রামটি দেখলেই বুঝতে পারবে কী কাজ করে। তবে শুধু দেখে বুঝলেই হবে না। কোড করে কম্পাইল ও রান করো।

সি ল্যাঙ্গুয়েজে আরও বেশ কিছু ডাটা টাইপ রয়েছে। ইনপুট ও আউটপুটের জন্যও রয়েছে নানা পদ্ধতি, যা তোমরা আস্তে আস্তে শিখবে (সব হয়তো এ বইয়ে থাকবে না, সি-এর অন্য বই পড়লে জানতে পারবে)। আপাতত যা শিখেছ, তা দিয়েই তোমরা অনেক প্রোগ্রাম লিখে ফেলতে পারবে।

এখন একটি মজার এবং দরকারি জিনিস বলে রাখি। প্রোগ্রামের কোডের ভেতরে তুমি নিজের ভাষাও ব্যবহার করতে পারো। এটিকে বলে কমেন্ট (comment) করা। কম্পাইলার কমেন্টগুলোকে প্রোগ্রামের অংশ ধরবে না। এক লাইনের কমেন্ট হলে // চিহ্ন

দিয়ে কমেট শুরু করতে পারো। আর একাধিক লাইন থাকলে /* দিয়ে শুরু এবং */ দিয়ে শেষ করতে হবে। নিচের প্রোগ্রামটি কিন্তু ঠিকঠাক কম্পাইল ও রান হবে।

```
#include <stdio.h>
int main()
{
    // test program - comment 1
    printf("Hello ");
    /* We have printed Hello,
    now we shall print World.
    Note that this is a multi-line comment */
    printf("World"); // printed world
    return 0;
}
```

প্রোগ্রাম: ২.১৬

এবারে একটি প্রশ্ন, (যেটি সি-এর টেক্সট বইয়ে এই চ্যাপটারের শুরুতেই বলে দিত), ভেরিয়েবলগুলোর নামকরণের নিয়মকানুন কী? ভেরিয়েবলের নাম এক বা একাধিক অক্ষরের হতে পারে, অক্ষরগুলো হতে পারে a থেকে z, A থেকে Z, 0 থেকে 9 এবং _ (আন্ডারস্কোর বা আন্ডারবার)। তবে একাধিক অক্ষরের ক্ষেত্রে প্রথম অক্ষরটা অঙ্ক (ডিজিট) হতে পারবে না। তুমি একটি প্রোগ্রামে int 7d; লিখে দেখো কম্পাইলার কী বলে। আর ভেরিয়েবলের নামগুলো অর্থপূর্ণ হলে ভালো হয়। যেমন, যোগফল রাখার জন্য ভেরিয়েবলের নাম sum হলেই ভালো, যদিও y নাম দিলেও প্রোগ্রাম চলে। অর্থপূর্ণ নাম দিলে বুঝতে সুবিধা হয়, ভেরিয়েবলটা কী উদ্দেশ্যে ব্যবহার করা হয়েছে।

৫ম অধ্যায়ঃ - প্রোগ্রামিং ভাবনা

The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it. -Mark Weiser [১]

আমরা কি অনুভব করি কী অভাবনীয় একটা সময়ে আমরা বাস করছি? এইতো একটু আগে আমি ডানে ঘুরে তাকালাম, আর পাঁচ ফুট দূরেই জানালার কাঁচে দেখতে পেলাম নিজের প্রতিবিম্ব। আমার ছবিটা কাঁচে প্রতিফলিত হয়ে আবার আমার কাছে ফিরে আসতে অতিক্রম করলো প্রায় দশ ফুট। এটুকু যেতে আসতে আলোর সময় লাগলো দশ ন্যানো সেকেন্ড [২]। আমার ল্যাপটপের প্রসেসরের ক্লক স্পিড যদি ১ গিগাহার্টজ হতো তাহলে এই অতি ক্ষুদ্র সময়ে সে প্রায় দশটা ছোট ছোট গণনার কাজ করে ফেলতে পারতো! কিন্তু আমার কম্পিউটারের স্পিড প্রায় ২ গিগাহার্টজ। এবং তার প্রসেসরের কোর আছে ৪ টা অর্থাৎ এই সময়েই সে, ৮০ টা গণনা কাজ শেষ করে ফেলেছে [৩]। এ প্রসঙ্গে ভাবা যেতে পারে অ্যাপোলো ১১ নভোযানের কথা যেটাতে করে প্রথম মানুষ চাঁদে পৌঁছায়। এই নভোজানের নিয়ন্ত্রণকারী কম্পিউটার পুরো ৮ দিনের মিশনে যতগুলো গণনা কাজ করেছে, তা আমার এই ল্যাপটপ করতে পারবে মাত্র ৩ মিনিটে [৪]! আমি নিশ্চিত তুমি যে কম্পিউটারের সামনে বসে এই লেখাটা পড়ছো সেটার ক্ষমতাও এর কাছাকাছি। কিন্তু কই, আমরা তো চাঁদে চলে যাচ্ছি না। এমনকি চাঁদে চন্দ্রবিন্দুর সাথে তুলনীয় কোনো “অ-সাম” কাজও করতে পারছি না আমরা কেউই। কী অভাবনীয় এক শক্তির আধার হাতে নিয়ে বসে আছি তা আমরা উপলব্ধি করছি না। আর যারা একটু আধটু উপলব্ধি করি তারাও সেটা ব্যবহারে ব্যর্থ হই কারণ আমরা প্রোগ্রামিং জানি না। আর জানলেও, কখনো নিজের কম্পিউটারের ক্ষমতার আলটিমেট ব্যবহার কী করতে পারি তা নিয়ে ভাবি না আমরা কেউ। কম্পিউটারের সাহায্যে কী অভাবনীয় সব কাজ কর্ম করা হচ্ছে এবং নিজেদের অজান্তেই তার কতকিছু আমরা ব্যবহার করছি সেসব নিয়ে লিখবো অন্য কখনো। আজকের এই লেখাটি যারা প্রোগ্রামিং শিখতে চায় তাদের জন্য। অনেকটা প্রশ্ন-উত্তর ফরম্যাটে লেখা [৫]। পাঠকরা যদি নিজেদের প্রশ্নের উত্তর মূল লেখায় না পান, তাহলে মন্তব্যের ঘরে জিজ্ঞেস করলে পরবর্তী ভাষনে আপডেট করে নেওয়া হবে। তাহলে শুরু করা যাক,

১) প্রথম কোন প্রোগ্রামিং ভাষাটি শিখবো? সি, সি++, পাইথন, জাভা, হ্যাসকেল, প্যাসকেল, লিসপ, অ্যাসেম্বলি, ম্যাটল্যাব, আর, ফরট্রান ... ব্লা ব্লা ব্লা?

উত্তর:- সি

কারণ:- আমি ধরে নিচ্ছি, তুমি তোমার বাবা-মা বা বন্ধু-বান্ধবকে ইম্প্রেস করার জন্য প্রোগ্রামিং শিখতে চাইছো না। অথবা শুধু ঘরে বসে আউটসোর্সিং করে বড়লোক হওয়াই

তোমার উদ্দেশ্য নয়। আমি ধরে নিচ্ছি, তুমি প্রোগ্রামিং শিখতে চাচ্ছে কারণ বাংলাদেশ যখন চন্দ্রযান পাঠাবে তার নিয়ন্ত্রণকারী কম্পিউটারের অপারেটিং সিস্টেম লিখবে তুমি, অথবা তুমি হয়তো ছোট্ট একটা রোবট বানাতে চাও যেটা তোমার টুকটাকি কাজ করে দেবে। তুমি হয়তো চাও লক্ষকোটি টাকা দামে বিদেশ থেকে আনা জীবন রক্ষাকারী মেডিক্যাল ইকুইপমেন্ট কম খরচে দেশে বসেই তৈরি করতে[৭]। তুমি হয়তো এমন একজন এক্সপার্ট হতে চাও যে বিভিন্ন কল কারখানা অটোমেটেড করে ফেলতে পারে। অথবা তুমি হয়তো চাও কৃত্রিম বুদ্ধিমত্তা নিয়ে গবেষণা করতে... (এই তালিকা অনেক বড় হতে পারে)। অর্থাৎ তুমি যদি চাও, কম্পিউটারের সাহায্যে তোমার আসেপাশের প্রকৃতির সাথে মিথোস্ক্রিয়া করতে তাহলে সি-ই তোমার প্রথম প্রোগ্রামিং ভাষা। সি, এর মত এত পরিষ্কারভাবে আর কোনো ভাষাই কম্পিউটারকে আমাদের সামনে তুলে ধরতে দেখিনি। অ্যাসেম্বলি বা মেশিন ল্যাঙ্গুয়েজে হয়তো কম্পিউটারের নাড়ি নক্ষত্রের উপর আরো বেশি নিয়ন্ত্রণ পাওয়া যায়। কিন্তু ওর জন্য প্রয়োজনীয় সুদীর্ঘ ডিটেইল বেশিরভাগ সময়ই মূল চিত্রটা ঢেকে ফেলে। প্রথম শেখা কম্পিউটার ভাষাও অনেকটা মাতৃভাষার মত। পরবর্তী জীবনে অন্য অনেক ভাষা নিয়ে কাজ করলেও, মূল অ্যালগরিদমটা তুমি হয়তো সি-তেই ভাববে। সি, প্রতিটি ভেরিয়েবলকে কোনো রাখঢাক না করেই তুলে দেয় আমাদের হাতে। শিখিয়ে দেয়, এত শক্তিশালী কম্পিউটারের দুর্বলতার স্থান কোনগুলো। ফলে তুমি যখন কম্পিউটারের সাহায্যে তোমার হাতে বানানো কোনো যন্ত্রকে নিয়ন্ত্রণ করতে চাইবে, তখন সি এর জ্ঞান তোমাকে এগিয়ে নেবে অনেকখানি। সি জানলে, তুমি অন্যান্য হাই লেভেল ভাষাকে অ্যাপ্রিশিয়েট করতেও শিখবে। এবং তাদের শক্তি বা দুর্বলতাটাও বুঝতে পারবে।

২) আমি একবার একটা প্রোজেক্টের জন্য সি(বা ম্যাটল্যাব বা আর বা ম্যাথমেটিকা বা জাভা) ব্যবহার করেছি। আমি কি এখন প্রোগ্রামিং শিখতে গিয়ে বাড়তি সুবিধা পাবো?

উত্তর:- এই তো কামটা সেরেছ। তোমার জন্য প্রোগ্রামিং শেখা একেবারে নতুন কারো চেয়ে অনেকগুণ কঠিন হয়ে গেছে।

কারণ:- কোনো একটা প্রোগ্রামিং ভাষা শেখা। আর কোনো একটা প্রোগ্রামিং ভাষা ব্যবহার করে কিছু-মিছু করে ফেলা এক কথা নয়। যেমন আমি হয়তো, ইন্টারনেট ঘেটে কিছু ফ্রেঞ্চ কথা শিখে প্যারিসের কোনো মুদি দোকান থেকে ঠিক ঠিক দুই কেজি আলু কিনে আনলাম। কিন্তু এর মানেই আমি ফ্রেঞ্চ শিখে ফেলেছি তা নয়। প্রোগ্রামিং ভাষার ক্ষেত্রে ব্যাপারটা আরো করুণ। কারণ প্রোগ্রামিং ভাষার শুরুতে, নানান টাইপের ভেরিয়েবল কোনটার কাজ কী, কিভাবে দুইটি সংখ্যা যোগ করা যায়। একটা স্ট্রিং-কে কিভাবে উলটো করে লেখা যায়। কত ভিন্ন ভিন্ন উপায়ে লুপ ঘোরাণো সম্ভব। একটা তালিকা থেকে নির্দিষ্ট কোনো উপাদান কিভাবে সব চেয়ে দ্রুত খুঁজে বের করা যায়। পয়েন্টার কিভাবে কাজ করে। নেগেটিভ সংখ্যার বাইনারি প্রকাশ কীভাবে করে। ফাংশন। রিকার্সন। এইসব খুটিনাটি খুব নিখুত ভাবে শিখতে হয়। এবং এগুলোতে পারফেকশন

আনতে মাসের পর মাস সাধণা করতে হয়। ওদিকে, অতীতে হয়তো, এর ওর কোড থেকে কিছু অংশ নিয়ে, একটু এদিক ওদিক করে, তুমি বড় কোনো সিমুলেশন করে ফেলেছ। সেইসব বড় বড় কাজের স্মৃতি, তোমাকে এসব খুঁটিনাটি শেখার সময় অধৈর্য্য করে তুলবে। আমি আমার অনেক বন্ধুকেই, শুধুমাত্র এই কারণে সঠিকভাবে প্রোগ্রামিং শিখতে ব্যর্থ হতে দেখেছি। এই ফাঁদ কাটিয়ে ওঠার উপায় তোমাকে নিজের মত করে বের করতে হবে। তবে এ বিষয়ে আমার টিপ্স হলো, নিজের জন্য কিছু বেঞ্চমার্ক ঠিক করা। ধরো, নিজেকে প্রশ্ন করলে, যে ভাষাটি তুমি শিখছো, সেই ভাষায় কি তুমি একটা সিম্পল “নোট প্যাড” (টেক্সট এডিটর) প্রোগ্রাম বানাতে পারবে; যেটার সাহায্যে টেক্সট ফাইল খোলা যায়, এডিট করা যায়, কোনো শব্দ সার্চ করা যায়, রিপ্লেস করা যায়, নতুন কিছু টাইপ করা যায় এবং পরিবর্তিত লেখাটা সেইভও করা যায়? এটা যেদিন বানাতে পারবে, বা সত্য-সত্যই বানাতে পারার মত কনফিডেন্ট হতে পারবে। সেইদিন তুমি ঐ ভাষাটা মোটামুটি আয়ত্ত্ব করে ফেলেছ।

৩) শুনেছি প্রোগ্রামার হতে হলে খুব ক্রিয়েটিভ হতে হয়। আমার তো অতো ক্রিয়েটিভিটি নেই, আমি কি পারবো?

উত্তর:- একদম বাজে কথা।

কারণ:- প্রোগ্রামিং শিখতে গেলে, যে দুইটা জিনিস লাগে তা হলো নিষ্ঠা, আর সততা। জিদ ধরে লেগে থাকতে হয়। বেশিরভাগ শিক্ষার্থী যাদের দেখে মনে হয়, অনেক চেষ্টা করেও প্রোগ্রামিং ঠিক মত আয়ত্ত্ব করতে পারলো না, তারা স্রেফ নিজের কাছে অসৎ। তাই পারেনি। হয়তো কিছু একটা ধারণা পুরোপুরি পরিষ্কার হয়নি কিন্তু সে এগিয়ে গেল পরের টপিকে। এবং নিজেকে মিথ্যা প্রবোধ দিলো যে সে আগের জিনিসটা বুঝেছে। ফলে পরে উচ্চতর কোনো সমস্যা সমাধান করতে গিয়ে সে আটকে যাবে। হয়তো সে বুঝবেও না যে, অতীতে যে মামুলি জিনিসটা সে উপেক্ষা করেছিলো, সেটার কারণেই সে আজ আটকে আছে। সে ভাববে, সে হয়তো যথেষ্ট ক্রিয়েটিভ নয়। এ ছাড়া আমরা আরেকটা কথাও ভাবতে পারি। কেউ লিখতে পারলেই তো রবীন্দ্রনাথ ঠাকুর হয়ে ওঠে না। কিন্তু চেষ্টা সাধণা করলে, যে কোনো ভাষাতেই যে কেউ দক্ষতা অর্জন করতে পারে। কিন্তু মানুষের ভাষা নিয়ে আসলে খুব বেশি কিছু করার নেই। কবিতা লেখা, বা গান, বা গল্প, বা রচনা- প্রবন্ধ-ভাষণ দেওয়া, আড্ডাবাজি এসবের বাইরে তেমন কিছু না। কিন্তু কম্পিউটার ভাষায় দক্ষতা থাকলে এত হাজার কোটি ভিন্ন ভিন্ন জিনিস করা সম্ভব যে, তার কোনো না কোনোটা তোমার মন মত হবে। আর তখন নিজের সৃষ্টিশীলতা কোথা থেকে যে এসে হাজির হবে সেটা বুঝতেও পারবে না।

৪) প্রোগ্রামিং ভাষা শেখার সবচেয়ে উপযুক্ত বয়স কোনটা?

উত্তর:- যে বয়সে তোমার মনে হবে, “আরেহ! আমি তো এটা বুদ্ধিমান প্রাণী। বুদ্ধি

খাটিয়ে অসাধারণ সব কাজকর্ম তো চাইলে আমি করতেই পারি!” সেই বয়সই প্রোগ্রামিং শেখার সব চেয়ে উপযুক্ত বয়স। কেউ হয়তো ক্লাস ফোর ফাইভেই নিজের বুদ্ধিমত্তা সম্পর্কে সচেতন হয়ে উঠবে আবার কেউ হয়তো বুড়ো হয়ে মরে যাবে কিন্তু খেয়াল করবে না যে সে একটা বুদ্ধিমান প্রাণী ছিলো।

কারণ:- গণিত, কম্পিউটার প্রোগ্রামিং, মিউজিক, জিমন্যাস্টিক্স এসব জিনিস যতো আগে থেকে শিখতে শুরু করা যায় ততোই ভালো। আন্তর্জাতিক প্রোগ্রামিং প্রতিযোগীতা, গণিত অলিম্পিয়াড বা অলিম্পিক এসবে যারা সেরাদের সেরা হয় তাদের বেশিরভাগই একেবারে ছোট থেকেই ওসব শিখতে শুরু করেছে। তবে প্রতিযোগীতামূলক ভাবে কিছু করা আর সৃষ্টিশীল কিছু করা এক কথা নয়। তাই অনেক পরে যারা শেখে, তারাও অভাবনীয় সব সৃষ্টিশীল কাজ করতে পারে এবং অহরহ করছে তাদের প্রোগ্রামিং স্কিল এর সাহায্যে। এমনকি ষাটোর্ধ বা পঞ্চাশোর্ধ কেউও চাইলে প্রোগ্রামিং শিখতে পারে। এ ধরনের কার্যক্রম তাদের ব্রেনকে আরো অনেকদিন সক্রিয় রাখবে। এবং একটা কিশোর কম্পিউটার প্রোগ্রামিং শিখে যে আনন্দ পায়, সেই আনন্দটা সেও পেতে পারবে ঐ বয়সেও।

৫) আমি ভাষাতত্ত্ব, বায়োলজি, মেডিক্যাল, অর্থনীতি, সিভিল, ইঞ্জিনিয়ারিং, চারুকলা, সামাজিক বিজ্ঞান, বিবিএ ... ইত্যাদি কোনো একটি বিষয়ে পড়েছি। কম্পিউটার প্রোগ্রামিং শিখে আমার কী লাভ?

উত্তর:- ওরে কে কোথায় আছিস ওকে দুটো টাকা দে, কারণ ও সব কিছুতে সবার আগে লাভ খোঁজে।

কারণ:- তুমি কি নোয়াম চমস্কির নাম শুনেছ? সে একজন ভাষাতাত্ত্বিক ও দার্শনিক। কম্পিউটার বিজ্ঞানেও তার মৌলিক অবদান আছে। আজকাল ভাষাতত্ত্ব সামাজিক বিজ্ঞান, এসবের অত্যাধুনিক গবেষণা হচ্ছে লক্ষ লক্ষ মানুষের তথ্য উপাত্ত বিশ্লেষণ করে। যা কম্পিউটার ছাড়া সম্ভব নয়। জেনেটিক্স, অনুজীব বিজ্ঞান, ফার্মাসিউটিক্যালস, অত্যাধুনিক অস্ত্রপচার পদ্ধতি, কৃষি, ইত্যাদি সকল জীব বিজ্ঞান সংশ্লিষ্ট বিষয়ের গবেষণা এখন কম্পিউটার ছাড়া কল্পনাও করা সম্ভব নয়। নগর পরিকল্পনাবিদ, ভূতাত্ত্বিক, খনি অনুসন্ধান কারী, নতুন ধরনের সেতু নির্মাণ কারী, গ্লোবাল ওয়ার্মিং এর কারণে সমুদ্রের উচ্চতা বেড়ে গেলে কিভাবে সেটা সামলাবো সেসব নিয়ে গবেষণাকারী কম্পিউটার ছাড়া একটুও এগোতে পারবে না। লিওনার্দো দ্য ভিঞ্চি যে সময়ে জন্মেছিলেন, সেই সময়ে ছবি আঁকার সবচেয়ে সেরা/লেইটেস্ট পদ্ধতি ছিলো তেল রঙ এ আঁকা। তিনি সেটাই ব্যবহার করেছেন। এমনকি সেই রঙের উন্নয়ন কিভাবে করা যায় সেটা নিয়েও গবেষণা করেছেন। এ যুগের লিওনার্দ দ্য ভিঞ্চির কাজ করবে কম্পিউটার গ্রাফিক্সে। এবং এ ক্ষেত্রে কম্পিউটার প্রোগ্রামিং জানলে কতটা বাড়তি সুবিধা পাওয়া সম্ভব সেটা বাইরে থেকে বোঝা সম্ভব নয়। সাম্প্রতিক অর্থনৈতিক মান্দার পরে, বড় বড় কোম্পানিগুলো বুঝে গেছে, কোট টাই পরে সুন্দর সুন্দর প্রেজেন্টেশন দেওয়ার চেয়েও গুরুত্বপূর্ণ স্কিল হলো

মার্কেট প্রেডিকশন করতে পারা। এখন মার্কেট এত হাজারটা জিনিশের সাথে এমন ভাবে জড়িত, যে কম্পিউটারের সাহায্য ছাড়া সেগুলোর উপর কোনো রকম নিয়ন্ত্রণই সম্ভব নয়। আমরা বাংলাদেশে থেকে এসব হয়তো উপলব্ধি করি না। যেদিন উপলব্ধি করবো, সেইদিন আমাদের দেশটা আর এমন থাকবে না।

৬) আমি ফিজিক্স, কেমিস্ট্রি, ম্যাথ বা বায়োলজি পড়ছি, কম্পিউটার প্রোগ্রামিং শিখে আমার কী হবে?

উত্তর:- তুমি হয়তো এসব বিষয়ে পড়েছ কিন্তু গবেষণা করছো না, বা এসব বিষয়ের সাম্প্রতিক গবেষণা নিয়ে খোঁজ খবর রাখো না। তাই এই প্রশ্ন করলে।

কারণ:- তুমি ফিজিক্স বা কেমিস্ট্রিতে পাশ করে ফেলেছ কিন্তু কম্পিউটার প্রোগ্রামিং জানো না। এর মানে হলো বিজ্ঞানী হিসাবে তুমি নিশ্চিত ভাবেই প্রথম শ্রেণীর কিছু হতে পারবে না। লার্জ হ্যাড্রন কলাইডার, গ্রাফেন, কোয়ান্টাম অপটিক্স, কোল্ড অ্যাটম, যে কোনো বিষয়ে গুগল করলেই দেখবে, কম্পিউটার ছাড়া ওসব গবেষণা স্বেচ্ছা অচিন্তনীয়। কেমিস্ট্রির ক্ষেত্রেও তাই। ইন ফ্যাক্ট এ যুগে কেমিস্ট্রি গবেষণা টেস্ট টিউবের চেয়ে বেশি হয় কম্পিউটারে। যেখানে নানান রকম নিউমেরিক্যাল মেথডের জয়-জয়কার। আর গণিতের ক্ষেত্রেও কাটিং এজ রিসার্চ করতে প্রোগ্রামিং না জানার বিকল্প নেই। কেন বা কীভাবে সেটা লিখবো অন্য কোনো সময়। বায়োলজি প্রসঙ্গটা এসেছে আগের প্রশ্নে, এর কারণ সাধারণভাবে দেখলে বায়োলজিকে গণিত, কম্পিউটার এসবের থেকে অনেক দূরের বিষয় মনে হয়। কিন্তু এই ধারণা যে কত বড় ভুল!

৭) প্রোগ্রামিং শিখে চাকরী পাবো কোথায়?

উত্তর:- যে প্রোগ্রামিং জানে, সে চাকরী খোঁজে না। চাকরীই তাকে খোঁজে।

কারণ:- তুমি যে একাডেমিক ব্যাকগ্রাউন্ডেরই হও না কেন সৎ ভাবে চেষ্টা করলে প্রোগ্রামিং শিখতে ফেলতে পারবেই। এবং আজকাল, নিজেকে একজন সুযোগ্য প্রোগ্রামার প্রমাণ করার জন্য বড় কোনো বিশ্ববিদ্যালয়ের ডিগ্রি থাকার দরকার নেই। অনলাইনেই সেটা প্রমাণ করার হাজারটা উপায় আছে। এত এত বিশ্ববিদ্যালয় থেকে হাজার হাজার কম্পিউটার বিজ্ঞানে পাশ করা ছেলেমেয়েকে হয়তো বের হতে দেখা। তাদের অনেকের মধ্যে চাকরী নিয়ে হতাশাও দেখা যায় প্রায়ই। তারাও যদি ঠিক ভাবে প্রোগ্রামিং শিখতো তাহলে এমন হতো না। কম্পিউটার বিজ্ঞান, কম্পিউটার ইঞ্জিনিয়ারিং এসব যারা পড়ে, তারা নানান রকম বিষয়ে ভালো গ্রেড পাওয়ার চাপে, প্রোগ্রামিং ঠিক মত শেখে না। প্রোগ্রামিংকেও অন্য আর দশটা বিষয়ের মত মনে করে। স্বেচ্ছা এই ভুলটা কাটিয়ে উঠতে পারলেই আমাদের দেশের আইটি ইন্ডাস্ট্রি অন্য যে কোনো দেশের সাথে পাল্লা দিতে পারতো। আর তারা প্রোগ্রামিং না বুঝে মুখস্থ করে যেটাই এই সমস্যার মুখ্য কারণ।

৮) টিচ ইউওরসেল্ফ সি ইন ২১ ডেইজ নামক একটা বই কিনেছি। এটা পড়ে কি প্রোগ্রামিং শিখতে পারবো?

উত্তর:- না, পারবে না। ঠিক যেমন টিচ ইউওরসেল্ফ ফ্রেন্ড ইন ২১ ডেইজ পড়ে একুশ দিনে ফরাসি ভাষা শিখতে পারবে না তেমন।

কারণ:- মাস ছয়েক নিয়মিত খাটাখাটনি না করে প্রোগ্রামিং শিখতে চাওয়াটা স্রেফ হঠকারিতা। আর ছয় মাস, অনেক সময় হলেও খুব বেশি সময় নয়। ছয়মাস আগের নিজেকে কল্পনা করো। তখন প্রোগ্রামিং শিখতে শুরু করলে এখন তুমি সেটা শিখে ফেলতে। অবশ্য নানাবিধ অ্যালগরিদম, ডাটা স্ট্রাকচার এসবে পারংগমতা আসতে সময় লাগতো আরো বেশি। তবে এই জার্নি কষ্টকর হলেও, পুরো সময়টা এত বেশি আনন্দময় হবে, যার সাথে খুব অল্প কিছুই তুলনা চলে। প্রোগ্রামিং শেখার পিছনে সময় ব্যয় করে কেউ কখনো আফসোস করেনি।

৯) সি শেখার জন্য কোন বই পড়বো?

উত্তর:- ইংরেজীতে হার্বার্ট শিল্ডের টিচ ইয়োরসেল্ফ সি [১০] বইটা দিয়ে শুরু করা যেতে পারে। এ ছাড়া বাংলায় সুবিন ভাই এর বই[৮] , আমার এই বইয়ে বেসিক পাবে এবং অন্য আরো কিছু বই আছে। কয়েকটা ঘাটাঘাটি করে দেখতে হবে কোনটা ভালো লাগে।

কারণ:- একেক বই একেক জনের কাছে আবেদন সৃষ্টি করে। কোনো বিষয়ে পড়ে তেমন ভালো না লাগলে, শুরুতেই ভাবা ঠিক না যে বিষয়টা বোরিং। বরং খুব সম্ভবত তুমি যে বইটা পড়ছো, সেটা তোমার মন মত লেখা হয়নি। হাল না ছেড়ে একই বিষয়ে অন্য কোনো লেখকের অন্য কোনো বই চেষ্টা করে দেখ। এ ছাড়া ইন্টারনেটে অনেক রিসোর্স পাবে [৯]।

১০) প্রোগ্রামিং বই কীভাবে পড়ে?

উত্তর:- আগাপাশতলা।

কারণ:- বেশির ভাগ বইয়ের ভূমিকায়, বা প্রিফেইসে, বইটি কীভাবে পড়বে তা বলা থাকবে। তবে নিজের প্রথম প্রোগ্রামিং ভাষা পুরোপুরি শেখার ক্ষেত্রে যেটা সব সময় সত্যি, সেটা হলো কোনো পাতা না পড়ে পরের পাতায় যাওয়া যাবে না। প্রতিটি অধ্যায়ের সবগুলো এক্সারসাইজ নিজে করতে হবে। কোনো পাতায় কোড দেওয়া থাকলে সেটা নিজ হাতে টাইপ করে চালিয়ে দেখতে হবে। (কোনো অবস্থাতেই কপি পেস্ট করা যাবে না)। একবার উদাহরণের কোড রান হয়ে যাবার পরে তার নানান অংশ নিজের ইচ্ছে মত

বদলে বদলে এক্সপেরিমেন্ট করতে হবে। প্রতিটি পরিবর্তনের ফলে আউটপুটে কী বদল হচ্ছে, সেটা থেকে বুঝতে হবে কোডের কোন অংশের কাজ কী। অনেক সময় কোনো অধ্যায় পড়ে মনে হতে পারে সব বুঝে গেছি। কিন্তু সাঁতারের নিয়ম বুঝে ফেলা আর সাঁতার কাটতে পারা যেমন এক কথা নয়, প্রোগ্রামিংও তেমন।

১১) আমি আউটসোর্সিং করতে চাই। কিন্তু আপনি শুরুতে যেভাবে বললেন মনে হলো আউটসোর্সিং কে বাতিল করে দিলেন।

উত্তর:- আমি মোটেই তা বলিনি। আউট সোর্সিং করা খুবই ভালো। কিন্তু “ঘরে বসে আউট সোর্সিং করে বড় লোক হবো”, শুধু এরকম চিন্তা করে প্রোগ্রামিং শিখতে আসলে তুমি প্রোগ্রামিং এর অনেক মজা মিস করবে। এবং সম্ভবত ভালো প্রোগ্রামার হয়ে উঠতে পারবে না।

কারণ:- আমাদের দেশটা উন্নত বিশ্বের মত হয়ে যাবার আগ পর্যন্ত দেশে বসে প্রোগ্রামিং করে কর্মসংস্থানের জন্য বেশির ভাগ প্রোগ্রামারকেই আউটসোর্সিং করতে বা আউটসোর্সিংকারী প্রতিষ্ঠানে চাকরী করতে হবে। এবং ঠিক কী ধরনের কাজ তুমি পাচ্ছ তার উপর নির্ভর করে তোমাকে ব্যবহার করতে হবে নানান রকম প্রোগ্রামিং ভাষা। কিন্তু কম্পিউটারের প্রকৃত শক্তি সম্পর্কে জানতে। সেই শক্তি নিজের ইচ্ছেমত ব্যবহার করতে সি জানার বিকল্প নেই।

১২) এইসব ল্যাপটপ ডেস্কটপ কম্পিউটার আমার ভালো লাগে না। আমার লাগবে সুপার কম্পিউটার। কিন্তু সেটা আমি কোথায় পাবো?

উত্তর:- একেবারে শুরুতেই সুপার কম্পিউটারের অ্যাক্সেস তুমি পাবে না। কিন্তু সুপারকম্পিউটারের মত কম্পিউটার আসলে তোমার হাতের নাগালেই আছে।

কারণ:- সুপার কম্পিউটারে লক্ষ লক্ষ প্রসেসর থাকে, যারা একই সঙ্গে কাজ করে। ফলে এক ক্লক সাইকেলেই লাখ খানেক কাজ হয়ে যায়। নরমাল কম্পিউটারের প্রোগ্রামিং আর সুপার কম্পিউটারের প্রোগ্রামিং একটু আলাদা হয়। কিন্তু জেনে অবাক হবে, গেইম খেলার জন্য উচ্চক্ষমতার যে গ্রাফিক্স কার্ডগুলো তুমি ব্যবহার করো সেগুলো ব্যবহার করে তুমি প্যারালাল প্রোগ্রামিং করতে পারো। সি বা সিপ্লাসপ্লাসেই! [৬] যেটা তোমাকে প্রস্তুত করতে পারে, সত্যিকারের সুপার কম্পিউটার ব্যবহার করে বড় কোনো সমস্যা সমাধানের জন্য!

১৩) প্রোগ্রামিং করলে কি স্কুলের পড়ার ক্ষতি হয়?

উত্তর:- না।

কারণ:- কম্পিউটার প্রোগ্রামিং করতে শিখলে মস্তিষ্কের ক্ষমতা বাড়ে। মস্তিষ্কের ক্ষমতা বেড়ে যাওয়ার সাথে পড়াশুনার ক্ষতি হবার কোনো সম্পর্ক নেই। ক্ষতি হয় তখন, যখন প্রোগ্রামিং করাটাকে তুমি পড়ায় ফাকি দেবার অজুহাত হিসাব ব্যবহার করো। □

১৪) আমার বাবা (অথবা মা, বড় ভাই, আপু) মনে করে প্রোগ্রামিং করে করে আমি সময় নষ্ট করছি। তারা দিনে আমাকে একঘণ্টার বেশি কম্পিউটার চালাতে দেয় না। আমি কীভাবে কম্পিউটার প্রোগ্রামিং শিখবো?

উত্তর:- সম্ভবত তুমি সময় নষ্ট করছো ঠিকই। কিন্তু প্রোগ্রামিং করে নয়। অন্যকোনো ভাবে। ফলে তোমার রেজাল্ট খারাপ হচ্ছে।

কারণ:- বাবা-মা রা যেহেতু আশে পাশে অন্য কোনো ছেলে-মেয়েকে প্রোগ্রামিং করতে দেখে না, সেহেতু তোমাকে কম্পিউটার নিয়ে বসে থাকতে দেখে সেটাকেই সব নষ্টের গোড়া মনে করছে। দুঃখজনক ব্যাপার হলো, টিভির মত কম্পিউটারের সামনে বসেও কাজের কাজ না করে ঘণ্টার পর ঘণ্টা সময় নষ্ট করে ফেলা সম্ভব। তাই জেনে বা না জেনে বাবা-মাদের এমন সন্দেহ পোষণ করা মোটেই অমূলক নয়। তাহলে উপায়? উপায় আসলে আছে। প্রোগ্রামিং শিখতে গেলে নিয়মিত ইন্টারেস্টিং কিন্তু বেশ জটিল সব বিষয় নিয়ে গভীরভাবে চিন্তাভাবনা করতে হয়। এভাবে মাথা খাটানোর অভ্যাস যার, একটু খানি বাড়তি সময় দিয়ে তার পক্ষে স্কুলের পড়াশুনায় ভালো করা খুব একটা কঠিন নয়। আর এছাড়া যদি দিনে খুব কম সময় কম্পিউটার ব্যবহার করার সুযোগ হয় তাহলেও প্রোগ্রামিং শিখতে কোনো বাধা নেই। কারণ কম্পিউটার ছাড়াই, কোনো সমস্যা কাগজ কলমে সমাধান করা যায়। সেটা কীভাবে প্রোগ্রাম করবে তা চিন্তা করে রাখা যায়। এমনকি খাতায় লিখেও ফেলা যায়। এভাবে প্রোগ্রামিং নিয়ে অনেকগুলো চিন্তাভাবনা আগে থেকেই করে রাখলে অভিভাবকদের হাত থেকে বা কোনো বন্ধুর বাসায় গিয়ে কম্পিউটার হাতে পাওয়া মাত্র সব চালিয়ে দেখে নেওয়া সম্ভব ঠিকঠাক কাজ করছে কি না। বা কোথায় ভুল হচ্ছে। জেনে অবাক হবে, এসিএম আইসিপিপি নামক যে আন্তর্জাতিক প্রোগ্রামিং প্রতিযোগিতা হয় সেখানে প্রতি দলে থাকে ৩ জন। কিন্তু কম্পিউটার থাকে মাত্র একটা। ফলে কেউ একজন যখন কম্পিউটারে প্রোগ্রাম করে বাকিরা খাতা কলমেই তাদের সমস্যাটা সমাধান করে। এভাবেই খুব সীমিত সময়ে মিলে মিশে সমস্যাগুলো সমাধান করতে হয়। তাই দেখা যাচ্ছে, কম সময় কম্পিউটার ব্যবহার করে (কিন্তু বেশি বেশি চিন্তা করে) কঠিন সব প্রোগ্রামিং সমস্যা সমাধান এর অভ্যাস হয়ে গেলে হয়তো তুমি পরে ওয়ার্ল্ড চ্যাম্পিয়নও হয়ে যেতে পারো!

পরবর্তীতে এখানে আরো কিছু প্রশ্ন-উত্তর যুক্ত করা যেতে পারে।

সূত্রাবলী-

[১] Weiser, M. The Computer for the 21st Century. Scientific American , September, 1991.

[২] এক ন্যানো সেকেন্ডে আলো যায় $3 \times 10^8 \times 10^{(-9)} \text{m} = 30 \text{ cm} = 11.81 \text{ inch} \approx 1 \text{ foot}$

[৩] আসলে একটা ইন্সট্রাকশন সম্পন্ন হতে, অনেকগুলো ক্লক সাইকেল লাগে। কিন্তু প্রতি ক্লক সাইকেলে, প্রসেসরের একটা মাইক্রো ইন্সট্রাকশন সম্পন্ন হয় ঠিকই। আমরা হিসাবে সুবিধার্থে সেটাকেই ধরেছি। এক গিগা হার্ড মানে প্রতি ন্যানো সেকেন্ডে একটা করে ক্লক পাল্স।

[৪] অ্যাপোলোর পুরো মিশনের সময় ছিলো, ৪ দিন ০৩ ঘন্টা ১৪ মিনিট ৩৫ সেকেন্ড। তার কম্পিউটারের ক্লক স্পিড ছিলো ১ মেগাহার্ড। এক ক্লকে একটার বেশি মাইক্রোইনস্ট্রাকশন সম্পন্ন করা সম্ভব নয়। ফলে এই পুরো অভিযানের সময় সে যতগুলো মাইক্রো ইনস্ট্রাকশন চালাতে পারলো হিসাব করলে দেখা যায়, আমার ল্যাপটপ সেটা পারবে ৩ মিনিটের একটু কম সময়ে। এছাড়াও আমার কম্পিউটারের রেজিস্টার সংখ্যা, পাইপ লাইনিং, বাস উইড, হিসাব করলে, ঐ পরিমান কাজ আসলে কয়েক সেকেন্ডে করতে পারার কথা এটার!

[৫] কিছু গুরুত্বপূর্ণ রেফারেন্স

<http://en.wikipedia.org/wiki/CUDA>

<http://www.sachalayatan.com/sporsho/43507>

<http://cpbook.subeen.com/>

<http://www.shikkhok.com/কোর্স-তালিকা/সি-প্রোগ্রামিং/>

<http://www.amazon.com/Teach-Yourself-C-Herbert-Schildt/dp/0078823110>

৬ষ্ঠ অধ্যায়:-C/C++ Struct

C++ Struct একটি অত্যন্ত জরুরি ফিচার C/C++ programming language এর। এই আর্টিকেল এ আমরা জানবো C structure বা C++ struct এর ব্যবহার।

প্রথমেই জানি কখন আমাদের এই ফিচার ব্যবহার করা লাগতে পারে। ধরলাম আমরা একটি ফোন বুক প্রোগ্রাম বানাবো, যেখানে একাধিক ব্যক্তির নাম, ঠিকানা, মোবাইল নাম্বার save করে রাখব। এখন সাধারনত আমরা কি করতাম ?

```
1char name[1000][1001]; // each name can be upto 1000 character
2int mobile [1000]; // we can store 1000 person's info
3int home [1000];
```

এখন একটা ব্যাপার খেয়াল হতে পারে, আমি একি লোক এর জন্য, ৩ টি array ব্যবহার করছি, এখন ধরলাম, আমি এই প্রোগ্রাম এ আসল ডাটা গুলোর ব্যাকআপ হিশেবে আরো ৩ টি array বানাবো। তাহলে কোড টা দাঁড়াচ্ছে এরকম-

```
1char name[1000][1001]; // each name can be upto 1000 character
2int mobile [1000]; // we can store 1000 person's info
3int home [1000];
4
5char backup_name[1000][1001]; // backup array
6int backup_mobile [1000];
7int backup_address [1000];
```

একি ব্যক্তি এর জন্য আমরা ৬ টি array declare করলাম, এখন কখনো যদি কোন array এর নাম পরিবর্তন করা লাগে, বা এরকম আরও ব্যাকআপ বানানো লাগে, তাহলে আমরা কিছু ঝামেলা তে পরব। তখন আমাদের মনে হতে পারে, যদি এই ৩ টি array কে একটা গ্রুপ এ রাখা যেতো, coding এ ঝামেলা কমতো। এই সমস্যা সমাধানে আমরা struct ব্যবহার করতে পারি।

এখন দেখা যাক, অই ৩ টি array কে কিভাবে একটি struct এ প্রকাশ করা যায়।

```
1struct telephone{
2char name[1001];
3int mobile;
4int home;
5};
```

খেয়াল করো, struct লেখার পর একটা নাম দিতে হয়, [যেমন এখানে telephone] এই নাম দিয়েই আমরা পরে নতুন নতুন struct declare করতে পারবো। আর খেয়াল করতে হবে } এর শেষে ; [সেমিকোলন] দিতেই হবে, নাহলে compilation error হবে। এবার দেখা যাক, struct declare করা যায় কিভাবে-

```
1 struct telephone entry [1000];
```

এবার আমরা struct এর ভিতরের element গুলো কে access করতে চাই, কিভাবে করব, কোড দেখাই-

```
1  
2 for(i=0;i<100;i++){  
3     scanf(" %s",entry[i].name);  
4     scanf("%d",&entry[i].mobile);  
5     scanf("%d",&entry[i].home);  
6 }  
7  
8 for(i=0;i<100;i++){  
9     printf("%s",entry[i].name);  
10    printf("%d",&entry[i].mobile);  
11    printf("%d",&entry[i].home);  
12 }
```

কাজ সহজ হয়ে গেলো অনেক , তাই না ?

এই struct feature ব্যবহার করে অনেক ডাটা স্ট্রাকচার , গ্রাফ প্রব্লেম এর অ্যালগরিদম সহজেই ইমপ্লিমেন্ট করা যায়।

৭ম অধ্যায় :-If ,else if....else function.

```
#include<stdio.h>

int main(void)
{
    int i;
    int a;
    printf("Apne ki Sihab Sizan ke pochondo koren? if yes press 0 if no
    press 1: ");
    scanf("%d",&i);
    if(i==0)
    {
        printf("Apne ekta khati bolda");
    }
    else if(i==1)
    {
        printf("Apni bibekban manus.apnake salute!ekhon apner age bolen");
        scanf("%d",&a);
        if(a<=30)
        {
            printf("apnar moto torun kei desher dorkar");
        }
        else
        {
            printf("chacha cholen Sihab ke pitaya vochkay dei!");
        }
    }
    else
    {
        printf("sudhu 0 ar 1 tipate bola hoise ulta palta number tipan ken?");
    }
    getch();
}
```

উপরের কোডটায় মোট ৩০ লাইন আছে। এই কোড এর মাধ্যমে আমরা যেই আউটপুট আশা করেছি তা নিচে দিলাম।

Computer: Apni ki Sihab Sizan ke pochondo koren? If yes press 0 if no press 1

এখন আপনি যদি ০ টিপেন তাহলে

computer বলবে : apne ekta khati bolda

আর যদি (Else if) ১ টিপেন তাহলে বলবে apni bibek ban manus.ekhon apnar boyos bolun.

এখন যদি আপনার বয়স ৩০ বা তার কম হয় তাহলে কম্পিউটার বলবে apnar moto torun kei desher dorkar.

আর যদি আপনার বয়স ৩০ এর বেশী হয়। তাহলে বলবে chacha cholen Sihab ke pitaya vochkay dei

আর যদি আপনি ০ বা ১ না টিপে অন্য সংখ্যা টিপেন

তাহলে বলবে sudhu 0 ar 1 tipate bola hoise ulta palta tipan ken?

এখন দেখি কোড তার প্রতি লাইন এ কি আছে

Int main(void) এর পর { এর মাধ্যমে ফাংশন এর স্টেটমেন্ট লেখা শুরু করা হল।

প্রথমেই ২ টি Variable নিলাম i আর a

এরপর কম্পিউটার এর পর্দায় প্রশ্ন টা আউটপুট পাবার জন্য printf("Apni bibekban manus.apnake salute!ekhon apner age bolen"); লিখা হইছে।

তারপরের লাইনে আপনাকে লিখতে হবে একটা কিছু ০-৯ এর মধ্যে। যদিও বলা হয়েছে শুধু ০ আর ১ লিখতে। scanf("%d",&i); আপনি যে সংখ্যা লিখবেন i এর ভিতর ঢুকে যাবে।

এর পরের লাইন

If(i==0)

{ printf("apne ekta khati bolda"); }

উপরের ইফ স্টেটমেন্ট এর মানে যদি i এর মান ০ হয় তাহলে কম্পিউটার কে লিখতে হবে 'apni ekta bolda'

এর পরের লাইন থেকে else if বা অথবা যদি স্টেটমেন্ট শুরু হল।

else if(i==1)

{ printf("Apni bibekban manus.apnake salute!ekhon apner age bolen"); মানে অথবা যদি i এর মান ১ হয় তাহলে কম্পিউটার লিখবে apni bibekban manus.apnake salute!ekhon apner age bolen।

scanf("%d",&a); আপনাকে যেহেতু বয়স জিজ্ঞেস করছে তাহলে আপনার বয়স বলতে হবে। এই লাইন তার মাধ্যমে আপনি আপনার বয়স ইনপুট দিবেন। যা a variable টায় ঢুকে যাবে।

if(a<=30) যদি আপনার বয়স বা a এর মান ৩০ অথবা ৩০ এর কম হয়

{printf("apnar moto torun kei desher dorkar"); কম্পিউটার লিখবে apnar moto torun kei desher dorkar.

} else অথবা

{ printf("chacha cholen Sihab ke pitaya vochkay dei!"); বলবে chacha cholen Sihab ke pitaya vochkay dei}

} - এই ব্রাকেট এর মাধ্যমে else if function er kaj sesh holo.

এরপর আবার else দেবার কারন। যদি কেউ ভুল করে ০ বা ১ না বসিয়ে অন্য কোন

সংখ্যা বসায় তার জন্য নতুন একটা কিছু দেখানো।

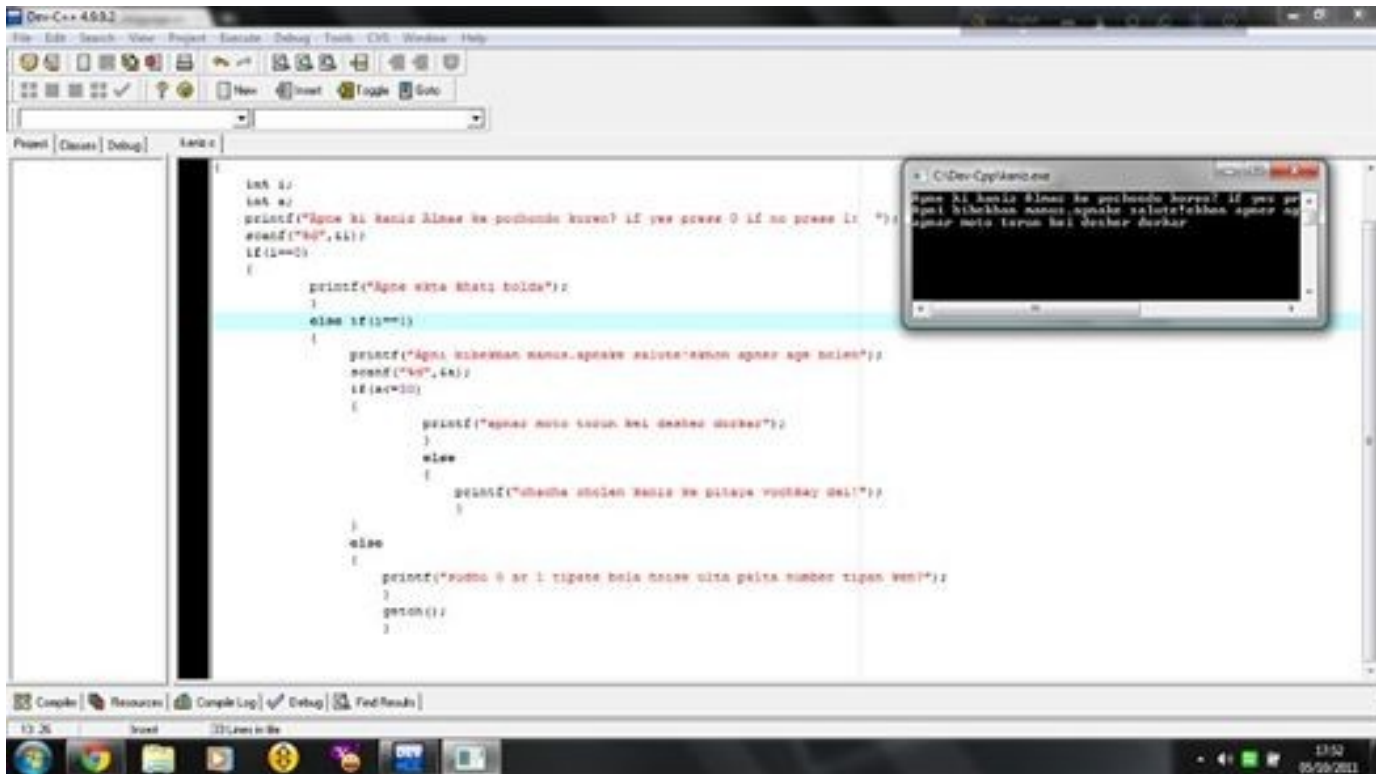
else {printf ("sudhu 0 ar 1 tipate bola hoise ulta palta number tipan ken?"); এর মানে হল যদি i এর মান ০ বা ১ ছাড়া অন্য কিছু বসানো হয় তাহলে কম্পিউটার বলবে sudhu 0 ar 1 tipate bola hoise ulta palta tipan ken?

} এই ব্রাকেট এর মাধ্যমে else function সমাপ্ত হল।

এর পর getch();

} পুরা প্রোগ্রাম শেষ হল।

আশা করি বুঝতে পেরেছেন। না বুঝলে জানাবেন।



The screenshot shows the Dev-C++ 4.9.9.2 IDE. The main editor window displays a C program with the following code:

```
int i;  
int a;  
printf("Agne ki kani alase ke posthondo kure? if yes press 0 if no press 1: ");  
scanf("%d",&i);  
if(i==0)  
{  
    printf("Agne kiti kati bolda");  
}  
else if(i==1)  
{  
    printf("Agni kibekhan manna.apake salute'mann agne age bolon");  
    scanf("%d",&a);  
    if(a==10)  
    {  
        printf("agane moto tarun hai dekho dekho");  
    }  
    else  
    {  
        printf("dekho unglien kani ke pitape vutray del");  
    }  
}  
else  
{  
    printf("sudhu 0 ar 1 tipate bola hoise ulta palta number tipan ken?");  
}  
getch();  
}
```

A small terminal window titled "C:\Dev-Cpp\kani.exe" is open in the top right corner, showing the program's output:

```
Agne ki kani alase ke posthondo kure? if yes pr  
Agni kibekhan manna.apake salute'khan agne ag  
agane moto tarun hai dekho dekho
```

The bottom status bar of the IDE shows the file name "13_26", the word "void", and "33 Lines in file". The system tray at the bottom right indicates the time is 13:32 on 05/09/2011.

৮ম অধ্যায়-Live Project1

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

```
int main(void)
```

```
{
```

```
    int cokh;
```

```
    int angul;
```

```
    printf("apnar cokh koyta? \n");
```

```
    scanf("%d",&cokh);
```

```
    printf("apnar hater angul koyta? \n");
```

```
    scanf("%d",&angul);
```

```
    if(cokh==2 && angul==10)
```

```
    {
```

```
        printf("apni normal ");
```

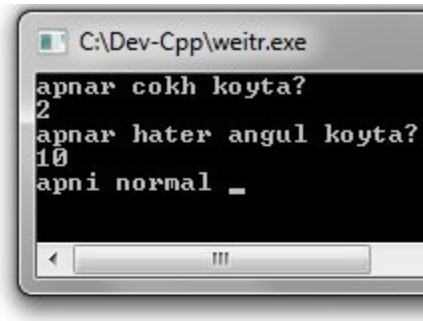
```
    }
```

```
    else
```

```
    {printf("hmmmm interesting "); }
```

```
    getch();
```

```
}
```



কিছু বুঝলেন ??? আমরা printf And Scanf ফাংশন ব্যবহার করে অনেক মজার একটা প্রোগ্রাম বানিয়ে ফেললাম তাইনা ??? আচ্ছা এতক্ষনে আমরা যত বিষয় সম্পর্কে জানলাম তার একটি রিয়েল লাইফ প্রোজেক্ট। আর এখানে প্রথমত আমরা ভেরিয়ারবলস ডিক্লেয়ার করছি এর পর সেগুলোর মান দিয়ে রাখছি সফটওয়্যারটা আপনার থেকে ২ টি নাম্বার ইনপুট নিচ্ছে এরপরে যদি ম্যাচ হয় তাহলে বলছে আপনি নরমাল আর অনথ্যায় হুম ইন্টারেস্টিং। খুব মজার প্রোগ্রাম তাই না ????

৯ম অধ্যায়- While loop

আজ আমরা শিখব While loop . মনে করেন আপনি প্রতিদিন চটপটি খান। আপনি ৫ দিনে গড়ে কত প্লেট খান তা জানতে চান। তার জন্য প্রয়োজনীয় প্রোগ্রাম আমরা আজ লিখব। কোড আমি লিখে ফেলেছি। নিচে দিলাম

```
/* akta chotpoti khabar contest */

#include<stdio.h>

int main(void)
{
    int chotpoti;
    int loopcount;
    int total;
    int average;
    total=0;
    loopcount=0;

    while(loopcount<5)
    {
        printf("Koy plate chotpoti khailen vai ajke? ");
        scanf("%d",&chotpoti);
        total=total+chotpoti;
        loopcount=loopcount+1;
    }
    average=total/5;
    printf("Apni protidin gore %d plate chotpoti khaisen ",average);
    getch();
}
```

কোডটার মোট লাইন ২১ টি।

প্রথম লাইন এ কমেন্ট আছে, আগেই বলেছি /* */ <- এটার ভিতর যা লিখবেন তা আপনার মতব্যা। এটা কম্পিউটার পড়বে না।

এর পর int main(void) এর মাধ্যমে মেইন ফাংশন শুরু করলাম { ব্র্যাকেট এর ভিতর

প্রথমেই ৪ টা variable declare করলাম। chotpoti,loopcount,total,average.
chotpoti ভেরিয়েবলটা কয় প্লেট চটপটি খেলেন তার মান স্টোর করবে।

loopcount হল যতবার কম্পিউটার একই কাজ করবে তার সংখ্যা।

total সর্বমোট যত প্লেট চটপটি খেলেন তার হিসাব রাখবে।

average ৫ দিন শেষে গড় বের করবে।

মনে রাখবেন variable এর নাম আপনারা যা খুশি দিতে পারবেন। average না দিয়ে যদি X দেন তাও হবে।

এখন আসি পরের লাইনে।

total=0 প্রথম দিন খাওয়া শুরুর আগে আপনার বন্ধু যদি জিজ্ঞেস করে কয় প্লেট খাইলি? আপনি বলবেন ০ প্লেট। SO total =0

loopcount=0 প্রথম দিন খাওয়া শুরুর আগে যদি আপনার বন্ধু জিজ্ঞেস করে কয়েকদিন ধরে খাচ্ছি? আপনি বলবেন আরে আজকেই তো শুরু করতেন। এখনো তো শুরু করিনি। তারমানে ০ দিন ধরে। loopcount=0 তাই।

এখন আসি আমরা while loop এ।

while(loopcount<5) while এর পর (টেস্ট কন্ডিশন এ আমরা বলে দিলাম loopcount 5 বার করতে হবে)

এর পর { এর মাধ্যমে while loop এর স্টেটমেন্ট লিখা শুরু করলাম।

printf("koy plate chotpoti khailen vai ajke? "); এই একটা লাইন লিখে দিলাম যাতে কম্পিউটার লিখে দেয়া কথাটা প্রোগ্রাম চলার সময় আমাদের জিজ্ঞেস করে।

এর পর এই প্রশ্নের উত্তর দেবার জন্য আপনার একটা ইনপুট ফাংশন scanf ব্যবহার করতে হবে।

scanf("%d",&chotpoti); আপনি এখন যেই সংখ্যা লিখবেন মানে যত প্লেট খাইছেন লিখবেন তা chotpoti variable তার মান হবে। প্রথম দিন যদি ৩ প্লেট খান তাহলে chotpoti র মান হবে ৩ ।

এখন total=total+chotpoti আগে বলেছি শুরুতে total=0 ছিল। প্রথম দিন খাবার পর total হবে 0+chotpoti র প্লেট এর সংখ্যা। প্রথম দিন 3 প্লেট খান যদি তাহলে মোট হবে total=0+3 =3 .total এর মান ০ থেকে ৩ হয়ে যাবে।

পরের দিন যদি ৫ প্লেট খান তাহলে পরদিন total =3+5=8 তাহলে ২য় দিন এর মান হয়ে গেল ৮।

এর পর খুব জরুরি একটা লাইন,

loopcount=loopcount+1; এই লাইন না দিলে আপনার কম্পিউটার while loop বন্ধ করত না। তারমানে আজীবন মেমরি স্পেস না শেষ হওয়া পর্যন্ত রান করত। শুরুতে loopcount=0 ছিল। প্রথম দিন শেষে আপনাকে আপনার বন্ধু যদি জিজ্ঞেস করে কয়েকদিন ধরে খাচ্ছি? আপনি বলবেন 0+1 দিন = 1 দিন। পরের দিন আবার জিজ্ঞেস করে কয়েকদিন ধরে খাচ্ছি? আপনি বলবেন 1+1 =2 দিন। এভাবে যখন ৫ম দিন শেষ হবে তখন আপনাকে জিজ্ঞেস করলে বলবেন 5+1=6 দিন তখন আপনার মনে পড়বে আরে!! আমি আগে বলছিলাম loopcount<5 তারমানে আমার আর খাওয়া যাবে না। তখন while loop close হবে।

} এর মাধ্যমে while loop স্টেটমেন্ট লিখা শেষ করলাম।

তারপর গড় বাহির করার জন্য লিখলাম average=total/5 আপনারা সবাই গড় বাহির

করার নিয়ম জানেন। কম্পিউটার এর ভাগ করার চিনহ-> / .total যত প্লেট খাইছেন তাকে দিনের সংখ্যা দিয়ে ভাগ করা হলেই গড় বের হবে।

এরপর গড় যাতে কম্পিউটার দেখা তার জন্য একটি printf ফাংশন লেখা দরকার।

Printf("apni protidin gore %d plate chotpoti kheyechen",average); %d এর জায়গায় average এর মান বসবে।

Expected result:

computer : koy plate chotpoti khailen aj vai ? : 4

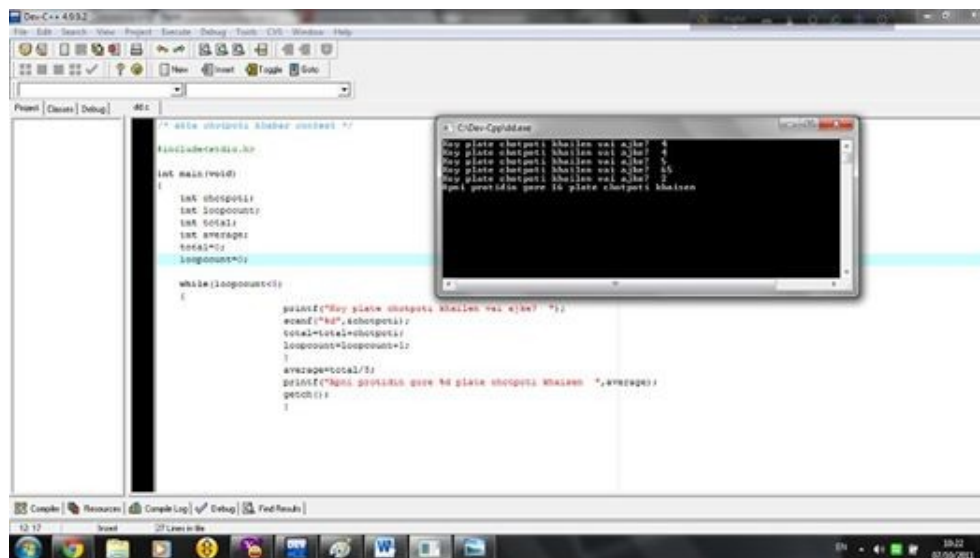
computer : koy plate chotpoti khailen aj vai ? : 3

computer : koy plate chotpoti khailen aj vai ? :2

Computer : koy plate chotpoti khailen aj vai?

Computer : koy plate chotpoti khailen aj vai ?

computer : apni protidin gore 3 plate chotpoti khaisen.



```
/* apni protidin gore 3 plate chotpoti khaisen */  
#include<iostream.h>  
  
int main(void)  
{  
    int chotpoti;  
    int loopcount;  
    int total;  
    int average;  
    total=0;  
    loopcount=0;  
  
    while(loopcount<5)  
    {  
        printf("koy plate chotpoti khailen aj vai? ");  
        scanf("%d",&chotpoti);  
        total=total+chotpoti;  
        loopcount=loopcount+1;  
    }  
    average=total/5;  
    printf("apni protidin gore %d plate chotpoti khaisen ",average);  
    getch();  
}
```

Output:

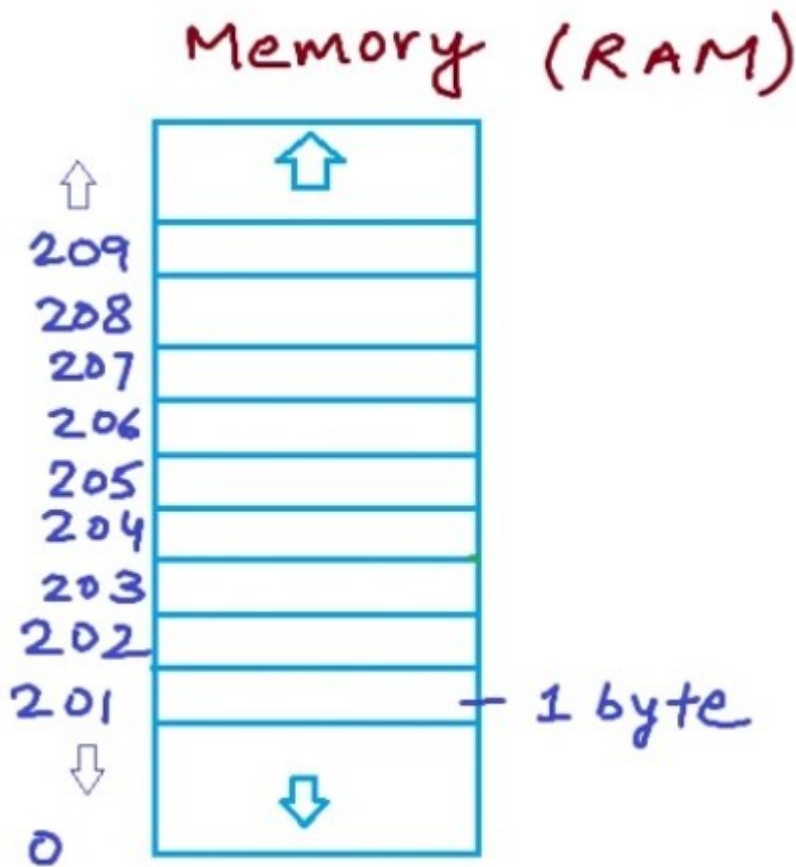
```
koy plate chotpoti khailen aj vai? 4  
koy plate chotpoti khailen aj vai? 3  
koy plate chotpoti khailen aj vai? 2  
koy plate chotpoti khailen aj vai? 2  
apni protidin gore 3 plate chotpoti khaisen
```

অধ্যায়:-১০ পয়েন্টার বিস্তারিত

পয়েন্টার প্রোগ্রামিং এ দরুন একটি টুল। পয়েন্টার সম্পর্কে জানার আগে কিছু ব্যাসিক জিনিস জানা যাক, যেগুলো বুঝতে কাজে দিবে।

[illegible]

ব্লক্সাম এর এক একটি সেল এক একটি বাইট। আর প্রত্যেকটা বাইট এর একটি করে এন্ড্রেস রয়েছে। আর প্রতিটা বাইটে ৮টি করে বিট রয়েছে।



আমরা যখন বলি আমাদের র‍‌ন্য়াম 8 Giga byte, তখন আমাদের কম্পিউটারের র‍‌ন্য়ামে মোট ৪ 000 000 000 bytes ডেটা স্টোর করা যাবে, এবং এদের প্রত্যেকের একটি করে এড্রেস রয়েছে। প্রথমটি

০ পরের টি 1, এর পরের টির এড্রেস 2 এভাবে বাড়তে থাকে। যদিও কম্পিউটার এ এড্রেস গুলো রিপ্রেজেন্ট করে হেক্সাডেসিমেল নাম্বার সিস্টেমে।

আমরা যখন একটি ভ্যারিয়েবল ডিক্লেয়ার করি আমাদের প্রোগ্রামে, এবং যখন আমরা প্রোগ্রামটি এক্সিকিউট/রান করি তখন কম্পিউটার ঐ ভ্যারিয়েবল এর জন্য কিছু মেমরি এলোকেট করে। কতবাইট মেমরি এলোকেট করবে, তা নির্বর করে ঐ ভ্যারিয়েবল এর ডেটা টাইপ এবং কম্পাইলার এর উপর।

সাধারণত কম্পাইলার গুলো একটা int এর জন্য 2 byte মেমরি এলোকেট করে। তেমনি একটি char ভ্যারিয়েবলের জন্য 1 byte মেমরি এলোকেট করে। floating-point নাম্বার এর জন্য 4 byte মেমরি এলোকেট করে।

যেমন যখন কম্পিউটার দেখে এমন একটি ডিক্লারেশন int a; তখন এটি বুঝতে পারে এটি একটি ইন্টিজার ভ্যারিয়েবল এবং এর জন্য ২ বাইট মেমরি এলোকেট করা দরকার। তখন র‍প্তাৎম এর খালি যায়গা থেকে এটি এই ইন্টিজারের জন্য ২ বাইট মেমরি এলোকেট করে।

আমরা সহজেই একটি ভ্যারিয়েবলের মেমরি লোকেশন বের করতে পারি, নিচের প্রোগ্রামটি দেখা যাক:

```
1#include <stdio.h>
2int main()
3{
4int a =5;
5printf("Memory address of variable a is: %x",&a);
6return 0;
7}
```

উপরের প্রোগ্রামটি রান করলে এমন কিছু দেখাবে: Memory address of variable a is: 2686732। এক কম্পিউটারে এক এক মান দেখাবে। এবং একবার একে এক ভ্যালু দেখাবে। কোন ভ্যারিয়েবল এর এর মেমরি এড্রেস জানার জন্য & [ampersand] ব্যবহার করা হয়। যাকে address-of operator [&] ও বলা হয়। যা দিয়ে আমরা অন্য একটি ভ্যারিয়েবল এর এড্রেস বা মেমরি লোকেশন পেতে পারি।

যখন আমরা প্রোগ্রামটি রান করি, তখন কম্পিউটার রপ্তায়াম এর খালি যায়গা থেকে ভ্যারিয়েবল a এর জন্য ২ বাইট মেমরি এলোকেট করে। কম্পিউটার অটোমেটিকেলি তখন a এর জন্য 2686732 এবং 2686733 নং সেল এলোকেট করে রাখে। আর মেমরি এড্রেস জানার জন্য শুধু মাত্র শুরু র এড্রেস জানলেই হয়। আমরা যখন a এর মেমরি এড্রেস প্রিন্ট করেছি, তখন শুধু শুরুর এড্রেস 2686732 ই পেয়েছি। যদি ও a ভ্যারিয়েবল এর জন্য 2686732 এবং 2686733 মেমরি এলোকেট করা হয়েছে এবং এর মান 5 এই দুই সেলে স্টোর করে রাখা হয়েছে। এখন আমরা যদি a এর মান পরিবর্তন করে অন্য আরেকটা ভ্যালু রাখি, যেমন 8, তখন রপ্তায়ামের 2686732 এবং 2686733 এ দুটো সেল এর মান ও পরিবর্তন হয়ে যাবে এবং এ দুটো সেলে 5 এর পরিবর্তে 8 স্টোর হবে। এবার পয়েন্টার কি জানা যাক।

পয়েন্টার হচ্ছে একটা ভ্যারিয়েবল যার ভ্যালু হচ্ছে আরেকটি ভ্যারিয়েবল এর মেমরি লোকেশন। পয়েন্টার একটা ডেটা, অ্যারে বা ভ্যারিয়েবল এর কম্পিউটার মেমরি লোকেশন রিপ্রেজেন্ট করে বা পয়েন্ট করে। অন্যান্য ভ্যারিয়েবল এর মত পয়েন্টার ভ্যারিয়েবল ব্যবহার করার আগে কম্পিউটার/কম্পাইলারকে বলতে হবে এটা একটি পয়েন্টার ভ্যারিয়েবল। নিচের মত করে একটি পয়েন্টার ভ্যারিয়েবল ডিক্লেয়ার করে।

```
data_type *name;
```

যেমন integer পয়েন্টারের জন্য: `int *i;`

asterisk [`*`] একটি ভ্যারিয়েবলের আগে ব্যবহার করে পয়েন্টার হিসেবে ডিক্লেয়ার করা হয়। যাকে indirection operator বা value-at-address operator বলা হয়। এখানে আরো কিছু ডেটা টাইপ এর পয়েন্টার ডিক্লেয়ারেশন এর উদাহরন দেওয়া হলো:

```
1 int*ip; /* pointer to an integer */
2 double*dp; /* pointer to a double */
3 float*fp; /* pointer to a float */
4 char*ch /* pointer to a character */
```

আমরা এখন দেখব কিভাবে পয়েন্টার ব্যবহার করতে হয় একটি প্রোগ্রামে।

```
1
2 #include <stdio.h>
3 int main ()
4 {
5     int a = 5; /* variable declaration */
6     int *ip; /* pointer variable declaration */
7     ip = &a; /* store address of "a" in pointer variable*/
8     printf("Address of a variable: %x\n", &a );
9     /* address stored in pointer variable */
10    printf("Address stored in ip variable: %x\n", ip );
11    return 0;
12 }
```

এখানে আমরা একটি ভ্যারিয়েবল `a` ডিক্লেয়ার করেছি। এরপর একটি পয়েন্টার ভ্যারিয়েবল ডিক্লেয়ার করেছি। তারপর পয়েন্টার ভ্যারিয়েবলে `a` এর মেমরি এড্রেস রেখেছি। তারপর `&` অপারেটর দিয়ে `a` ভ্যারিয়েবল এর এড্রেস প্রিন্ট করে দেখলাম। এবং পয়েন্টার ভ্যারিয়েবল এর ভ্যালু প্রিন্ট করে দেখলাম। উভয় এর মান ই একই।

আমরা ইচ্ছে করলে এখন `ip` পয়েন্টার ভ্যারিয়েবল দিয়ে `a` এর মান বের করতে পারি।

```
1 #include <stdio.h>
2 int main ()
3 {
```

```
3
4 int a = 5;
5 int *ip;
6 ip = &a;
7 /* access the value using the pointer */
7 printf("Value of *ip variable: %d\n", *ip );
8 return 0;
9 }
10
```

আমরা যখন প্রোগ্রামটি রান করব, তখন ip যে ভ্যারিয়েবলটির এড্রেস শো করবে, তার মান প্রিন্ট করবে। লক্ষ্য করি, যখন আমরা পয়েন্টার ভ্যারিয়েবল দিয়ে কোন ভ্যারিয়েবল এর এড্রেস বের করতে চাইবো, তখন শুধু পয়েন্টার ভ্যারিয়েবল লিখলেই হবে। কিন্তু যখন আমরা পয়েন্টার ভ্যারিয়েবল দিয়ে মূল ভ্যারিয়েবল এর ভ্যালু বের করতে চাইবো, তখন পয়েন্টার ভ্যারিয়েবল এর আগে * যোগ করতে হবে। যেমন প্রথম প্রোগ্রামে আমরা ip [পয়েন্টার ভ্যারিয়েবল] প্রিন্ট করায় আমরা এড্রেস পেয়েছি। এবং পরের প্রোগ্রামে ip এর আগে একটা * দিয়ে *ip প্রিন্ট করায় আমরা মূল ভ্যারিয়েবলের মান পেয়েছি।

অধ্যায় ১১-টেস্ট এক্সাম ও সমস্যা সমাধান

এবার এই জার্নটুকু থেকে আপনাদের কিছু প্রব্লেম দিব সম্বন্ধ করেন।

প্রোগ্রামিং চ্যালেঞ্জ ১

Print Your Name On Screen

প্রোগ্রামিং চ্যালেঞ্জ ২

-1+-1=???

প্রোগ্রামিং চ্যালেঞ্জ ৩

0*10000-1+9=???

প্রোগ্রামিং চ্যালেঞ্জ ৪

Input 25 And Root It

প্রোগ্রামিং চ্যালেঞ্জ ৫

Make A Calculator Using Scanf And Printf Fuction

আচ্ছা এই কাজ গুলি শেষ হলে আমাদের মেইল কর অথবা মেসেজ কর।

Mail:-sizan009@gmail.com

Website:-<http://teamerror.wc.it>

Facebook:-<http://facebook.com/sizan.first>

অধ্যায় ১২-প্রয়োজনীয় বিষয়

প্রোগ্রামিং প্রতিযোগিতা

প্রোগ্রামিং প্রতিযোগিতা হচ্ছে প্রোগ্রামারদের মধ্যে লড়াই। এর মানে কিন্তু এই নয় যে প্রোগ্রামাররা একে অপরের সঙ্গে মারামারি করবে আর শেষ পর্যন্ত যে টিকে থাকবে সে-ই বিজয়ী। আসলে প্রোগ্রামিং প্রতিযোগিতা হচ্ছে একটি পরীক্ষার মতো যেখানে প্রত্যেককে একটি নির্দিষ্ট সময়ে নির্দিষ্টসংখ্যক প্রোগ্রামিং সমস্যার সমাধান করতে দেওয়া হবে। যে সবচেয়ে বেশি সমস্যার নির্ভুল সমাধান করবে সে বিজয়ী হবে। আর দুজন যদি সমানসংখ্যক সমস্যার সমাধান করে, তবে তাদের মধ্যে যে কম সময়ে করেছে সে বিজয়ী। তবে স্কুল-কলেজের পরীক্ষার সঙ্গে এর পার্থক্য হচ্ছে, এখানে বই থেকে সরাসরি প্রশ্ন করা হয় না। তাই মুখস্থ করার কোনো সুযোগ নেই। বিচারকেরা অনেক সময় নিয়ে প্রোগ্রামিং প্রতিযোগিতার সমস্যা তৈরি করেন। এর মধ্যে সহজ সমস্যাও থাকে আবার খুব কঠিন সমস্যাও থাকে।

স্কুল-কলেজের ছাত্রছাত্রীদের জন্য সবচেয়ে বড় প্রতিযোগিতা হচ্ছে আইওআই (IOI- International Olympiad in Informatics)। 1989 সাল থেকে প্রতিবছর এ প্রতিযোগিতা অনুষ্ঠিত হচ্ছে। একেক বছর একেক দেশে প্রতিযোগিতা অনুষ্ঠিত হয়। বিগত চার বছর যাবৎ বাংলাদেশ এ প্রতিযোগিতায় অংশগ্রহণ করে আসছে। এখন পর্যন্ত আমাদের সেরা অর্জন হচ্ছে 2009 সালে আবিরুল ইসলামের রৌপ্য পদক (সিলভার মেডেল)। IOI-তে অংশগ্রহণ করার জন্য বাংলাদেশ দল গঠনের কাজটি করা হয় দুই ধাপে। প্রথমে বিভাগীয় ইনফরমেটিক্স অলিম্পিয়াড। তারপর বিভাগীয় পর্যায়ে বিজয়ীদের নিয়ে জাতীয় ইনফরমেটিক্স অলিম্পিয়াড অনুষ্ঠিত হয়। জাতীয় অলিম্পিয়াডের বিজয়ীদের মধ্য থেকেই দলের সদস্য বাছাই করা হয়।

বিশ্ববিদ্যালয় পর্যায়ের ছাত্রছাত্রীদের জন্য সবচেয়ে বড় প্রোগ্রামিং প্রতিযোগিতা হচ্ছে এসিএম আইসিপিসি (ACM ICPC- ACM International Collegiate Programming Contest)। এর জন্য দল বাছাই অনেকটা বিশ্বকাপ ফুটবলের মতো হয়। প্রতি মহাদেশ থেকে প্রতিযোগিতার মাধ্যমে দল নির্বাচন করা হয়। একটি দলে তিনজন সদস্য এবং একজন প্রশিক্ষক থাকেন। মজার ব্যাপার হচ্ছে দলের সদস্যদের কিন্তু কম্পিউটার বিজ্ঞানের শিক্ষার্থী হতে হবে, এমন কোনো কথা নেই। যেকোনো বিভাগের শিক্ষার্থী এই প্রতিযোগিতায় অংশগ্রহণ করতে পারে। বাংলাদেশের প্রতিযোগিতা ঢাকায় ICPC Regional Contest-এ অংশগ্রহণ করে। এছাড়া ভারত ও আশেপাশের দেশের ICPC Regional Contest-এও বাংলাদেশের প্রতিযোগীদের অংশগ্রহণের সুযোগ রয়েছে এবং প্রায়ই আমাদের দেশের কয়েকটি দল ওইসব প্রতিযোগিতায় অংশগ্রহণ করে। ICPC Regional Contest-এ বিজয়ী দলগুলো সুযোগ পায় চূড়ান্ত পর্বে (ICPC World Finals) অংশগ্রহণ করার। 1998 সালের পর থেকে প্রতি বছরই বাংলাদেশ থেকে কমপক্ষে একটি দল চূড়ান্ত পর্বে অংশগ্রহণের যোগ্যতা লাভ করে যা আমাদের

দেশের প্রোগ্রামারদের কৃতিত্বের পরিচয় বহন করে। তোমরা এ বিষয়ে উইকিপিডিয়াতে আরও তথ্য পাবে এই লিংকে:

http://en.wikipedia.org/wiki/ACM_ICPC_Dhaka_Site

এছাড়া ইন্টারনেটে অনুষ্ঠিত হয় আরও নানা ধরনের প্রোগ্রামিং প্রতিযোগিতা যেখানে স্কুল-কলেজ-বিশ্ববিদ্যালয়ের ছাত্র, শিক্ষক ও পেশাজীবীরা অংশগ্রহণ করতে পারেন। এদের মধ্যে গুরুত্বপূর্ণ তিনটি হচ্ছে Google Code Jam (<http://code.google.com/codejam>), Topcoder (<http://www.topcoder.com/tc>) এবং Codechef (<http://www.codechef.com/>)। এই প্রতিযোগিতাগুলো অত্যন্ত কঠিন, তাই এতে অংশগ্রহণের জন্য পর্যাপ্ত দক্ষতা থাকতে হবে। তবে এসব প্রতিযোগিতায় কিন্তু বাংলাদেশের প্রোগ্রামাররা বেশ ভালো অবস্থানে রয়েছে।

প্রোগ্রামিং প্রতিযোগিতায় ভালো করতে হলে প্রোগ্রামিংয়ে দক্ষতার পাশাপাশি অ্যালগরিদম ও গণিতে বিশেষভাবে দক্ষ হতে হয়। জ্যামিতি, কম্বিনেটরিক্স, সংখ্যাতত্ত্ব ইত্যাদির যথেষ্ট জ্ঞানের পাশাপাশি সমস্যা সমাধানের দক্ষতা অর্জন করতে হয়। এজন্য লেখাপড়ার পাশাপাশি নিয়মিত প্রোগ্রামিং সমস্যা সমাধানের কোনো বিকল্প নেই। আর বিভিন্ন ওয়েবসাইটে নিয়মিত প্রোগ্রামিং প্রতিযোগিতার আয়োজন করা হয় যেখানে তুমি অংশগ্রহণ করতে পারো ইন্টারনেটের মাধ্যমে।

প্রোগ্রামিং ক্যারিয়ার

গণিত যেমন কেবল গণিতবিদেরাই ব্যবহার করেন না, বরং বিজ্ঞানের সব শাখায় রয়েছে এর ব্যবহার, তেমনই প্রোগ্রামিংও কিন্তু কেবল কম্পিউটার বিজ্ঞানী বা কম্পিউটার ইঞ্জিনিয়ারদের জন্য নয়। বিশ্ববিদ্যালয়ে পড়তে গেলে বিজ্ঞান ও প্রকৌশলের সব বিভাগের শিক্ষার্থীদের জন্য প্রোগ্রামিং জানাটা খুব গুরুত্বপূর্ণ।

পেশা হিসেবে প্রোগ্রামিংয়ের আলাদা একটি গুরুত্ব আছে আমাদের জন্য। যেহেতু বিভিন্ন ধরনের প্রতিযোগিতার মাধ্যমে নিজেকে মেলে ধরার অনেক সুযোগ এখানে রয়েছে, তাই বাংলাদেশ থেকে লেখাপড়া করে সরাসরিই বিশ্বের নামকরা সফটওয়্যার নির্মাতা প্রতিষ্ঠান যেমন- গুগল, মাইক্রোসফট, ফেসবুক ইত্যাদিতে কাজ করার সুযোগ তৈরি হয়েছে। প্রতিবছরই বাংলাদেশ থেকে কয়েকজন প্রোগ্রামার নিজের মেধা ও জ্ঞানকে কাজে লাগিয়ে এই সুযোগের সদ্ব্যবহার করছেন। অনেক ক্ষেত্রেই ওইসব প্রতিষ্ঠানে কাজ করার জন্য আবেদন করার প্রয়োজন হয় না, তারা নিজে থেকেই বিভিন্ন দেশের সেরা প্রোগ্রামারদের খুঁজে বের করে।

বিশ্ববিখ্যাত সব প্রতিষ্ঠানে কাজ করা ছাড়াও প্রোগ্রামারদের জন্য আরেকটি সুবিধা হচ্ছে

Telecommuting। অর্থাৎ কোনো অফিসে না গিয়ে কাজ করার সুযোগ। উন্নত বিশ্বের অনেক কোম্পানি তাদের নিজ দেশে প্রোগ্রামারদের দূরপ্রাপ্যতার কারণে উন্নয়নশীল অর্থনীতির দেশের প্রোগ্রামারদের কাজের সুযোগ দেয়, আর সে ক্ষেত্রে নিজ দেশে বসেই কাজ করা যায়। কারণ ওই কোম্পানিগুলো জানে যে আমাদের মত দেশের অর্থনীতি উন্নয়নশীল হলেও প্রোগ্রামাররা মোটেও অদক্ষ নন, বরং বিশ্বমানের প্রোগ্রামার। বাংলাদেশের বেশকিছু প্রোগ্রামার এখন বাংলাদেশে বসেই ইন্টারনেটের মাধ্যমে কাজ করছেন আমেরিকা, কানাডা ও ইউরোপের বিভিন্ন দেশের সফটওয়্যার কোম্পানিতে।

আরেকটি মজার ব্যাপার হচ্ছে, কেউ যদি ধরাবাঁধা চাকরি করতে না চায় তবে তার জন্য ফ্রিল্যান্স প্রোগ্রামিংয়ের সুযোগ রয়েছে। ইন্টারনেটে অনেক ওয়েবসাইট আছে যেখানে ছোট-মাঝারি-বড় বিভিন্ন ধরনের সফটওয়্যারের প্রজেক্ট থাকে যেগুলোতে বিড (bid) করে কাজ করা যায়। বাংলাদেশে এখন শত শত প্রোগ্রামার ফ্রিল্যান্স প্রোগ্রামিংয়ের সঙ্গে জড়িত। এর জন্য কেবল কম্পিউটার ও ইন্টারনেট সংযোগ থাকলেই চলবে। ফ্রিল্যান্স কাজ করার জন্য বিপুল ধৈর্যের প্রয়োজন। আর ইংরেজি ভাষায় যোগাযোগের দক্ষতা থাকতে হয়। তবে ব্যক্তিগতভাবে আমি মনে করি, ছাত্রাবস্থায় এ ধরনের কাজ না করাই ভালো। কারণ ছাত্রজীবনে লেখাপড়া করার ও মৌলিক বিষয়গুলো আয়ত্ত্ব করার যে সময় ও সুযোগ মেলে, জীবনের পরবর্তী পর্যায়ে কখনোই সেই সুযোগ পাওয়া যায় না। তাই তোমাদের প্রতি আমার পরামর্শ থাকবে যে ছাত্রজীবনে অর্থ উপার্জনের দিকে মনোযোগ না দিয়ে প্রচুর লেখাপড়া এবং সঙ্গে সঙ্গে নানা ধরনের সামাজিক ও সাংস্কৃতিক কর্মকাণ্ডে জড়িত থাকার চেষ্টা করবে, যেগুলো তোমার ভালো লাগে।

সব শেষ কথা হচ্ছে, প্রোগ্রামিং এমন একটি কাজ যেখানে সব সময়ই তোমার নিজেকে উন্নত করার সুযোগ আছে। তাই লেখাপড়া করার মানসিকতা থাকতে হবে, পড়তে হবে নানা বইপত্র, ঘাঁটতে হবে ইন্টারনেট। নিজে কোনো সমস্যায় পড়লে প্রথমেই ইন্টারনেট ঘেঁটে দেখবে যে সমস্যাটির সমাধান ইতিমধ্যে কেউ করে রেখেছে কি না। বিভিন্ন ফোরাম ও ব্লগে সাধারণত প্রোগ্রামাররা কিছু কমন সমস্যার সমাধান দিয়ে রাখে। তবে প্রোগ্রামিং শেখার সময় কিন্তু সমাধানের জন্য ইন্টারনেট ঘাঁটবে না, নিজে চেষ্টা করবে।

বই ও ওয়েবসাইটের তালিকা

তুমি যদি ইতিমধ্যে এই বইটি পড়ে ফেলো এবং এবারে ভালোভাবে সি শিখতে চাও, তবে Herbert Schildt-এর Teach Yourself C বইটি পড়তে পারো। আবার Brian Kernighan ও Dennis Ritchie-এর লেখা The C Programming Language বইটিও পড়তে পারো। লেখকদের একজন, Dennis Ritchie, সি ল্যাঙ্গুয়েজ ডিজাইন করেছেন। আর কেউ যদি তোমার কাছে জানতে চায় শুরুতে সি শিখতে হলে কোন ইংরেজি বইটি ভালো তবে Stephen G. Kochan-এর Programming in C বইটির

কথা বলে দেবে। এটি সি শেখার জন্য চমৎকার ও সহজ একটি বই। Schaums Outlines সিরিজের Programming with C বইটিও ভালো। বইতে প্রচুর উদাহরণ আর অনুশীলনী আছে।

সি শেখার পরে তুমি সি প্লাস প্লাস বা জাভা শিখতে পারো। সি প্লাস প্লাস শেখার জন্য ভালো বই হচ্ছে Teach Yourself C++ (লেখক: Herbert Schildt) আর জাভার জন্য Java How to Program (লেখক: Paul Deitel and Harvey Deitel)। তারপর অন্য ল্যাঙ্গুয়েজ শিখতে গেলে আর বই কেনার দরকার নেই। ইন্টারনেটে প্রচুর টিউটোরিয়াল আছে। সেগুলো পড়ে শিখে ফেলবে।

সি এবং পাইথনের জন্য চমৎকার দুটি বই আছে অনলাইনে -

<http://learnpythonthehardway.org/book/>
<http://c.learnthethehardway.org/book/>

তুমি যদি কম্পিউটার বিজ্ঞানে পড়তে চাও, কিংবা প্রোগ্রামিং কন্টেস্টে ভালো করতে চাও, তাহলে তোমার Discrete Mathematics ভালো করে শিখতে হবে। এর জন্য Kenneth H. Rosen-এর Discrete Mathematics বইটি খুব ভালো। আগাগোড়া পড়ে ফেলবে। সঙ্গে সঙ্গে অনুশীলনীর সমস্যাগুলো সমাধানের চেষ্টা করবে। Discrete Mathematics শেখার পরে শিখতে হবে অ্যালগরিদম। অ্যালগরিদম শেখার শুরু আছে কিন্তু শেষ নেই। আর শুরু করার জন্য তোমরা পড়তে পারো Introduction to Algorithms (লেখক: Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein) এটি অ্যালগরিদমের মৌলিক বিষয়গুলো শেখার জন্য আমার দেখা সবচেয়ে ভালো বই।

প্রোগ্রামিং প্রতিযোগিতার জন্য কিছু লিংক:

<http://projecteuler.net/> এখানে অনেক মজার সমস্যা আছে যেগুলোর বেশিরভাগই প্রোগ্রাম লিখে সমাধান করতে হয়। এখানে প্রোগ্রাম জমা দেওয়া লাগে না, কেবল প্রোগ্রাম দিয়ে বের করা উত্তরটা জমা দিতে হয়।

<http://www.spoj.pl/> এখানেও অনেক ভালো সমস্যা আছে। সমাধান করে প্রোগ্রাম জমা দিলে প্রোগ্রাম সঠিক হয়েছে কি না তা জানা যায়। এই ওয়েবসাইটের একটি বৈশিষ্ট্য হচ্ছে সি, সি প্লাস প্লাস, জাভা, পার্ল, পাইথন, রুবি, পিএইচপি ইত্যাদি ব্যবহার করে প্রোগ্রাম লেখা যায়।

<http://uva.onlinejudge.org/> এই সাইটে নিয়মিত অনলাইন প্রোগ্রামিং প্রতিযোগিতার আয়োজন করা হয়। এ ছাড়াও অনুশীলনের জন্য প্রচুর সমস্যা দেওয়া আছে। নতুন প্রোগ্রামারদের জন্য এটি বেশ ভালো জায়গা।

<http://ace.delos.com/usacogate> এটি যদিও আমেরিকার ইনফরমেটিক্স অলিম্পিয়াড ট্রেনিং প্রোগ্রাম, কিন্তু সাইটে যেকোনো দেশের প্রোগ্রামাররাই রেজিস্ট্রেশন করে অনুশীলন করতে পারে। তোমরা যারা প্রোগ্রামিং প্রতিযোগিতায় ভালো করতে চাও,

তাদের অবশ্যই এখানে অনুশীলন করা উচিত।

<http://www.topcoder.com/tc> এখানেও নিয়মিত অনলাইন প্রোগ্রামিং প্রতিযোগিতা অনুষ্ঠিত হয়। এখানে ভালো ফলাফল করলে আবার টাকাও দেয় (কী আনন্দ!)। এ ছাড়া এখানে অনেক ভালো টিউটোরিয়াল ও আর্টিকেল আছে। এটি অভিজ্ঞ প্রোগ্রামারদের জন্য বেশ ভালো একটি সাইট।

<http://codeforces.com> এই সাইটে নিয়মিত বিভিন্ন ধরনের প্রোগ্রামিং কন্টেস্ট হয়। অভিজ্ঞ প্রোগ্রামারদের জন্য ভালো।

<http://www.codechef.com> এটিও প্রোগ্রামিং প্রতিযোগিতার জন্য একটি ভালো ওয়েবসাইট এবং অভিজ্ঞ প্রোগ্রামারদের জন্য।

<http://ioinformatics.org> আন্তর্জাতিক ইনফরমেটিক্স অলিম্পিয়াডের অফিসিয়াল ওয়েবসাইট।

<http://cm.baylor.edu/welcome.icpc> এসিএম আইসিপিএসির অফিসিয়াল ওয়েবসাইট।

প্রোগ্রামিং ছাড়াও বিজ্ঞান ও গণিতের নানা বিষয়ের জন্য এই ফোরামে অংশগ্রহণ করতে পারো: <http://matholympiad.org.bd/forum/>।

-----The End-----

আসলে সময়ের অভাবে খুব বেশি লিখতে পারিনি , আরো বেশি তথ্য দিতে চেয়েছিলাম।
আপাতত এটুকুই ।
খুব শীঘ্রই আসছে পরবর্তী ভাৰ্সন আরো তথ্য সহ এবং স্বয়ং সম্পূৰ্ণ।
বিস্তারিত জানতে চোখ রাখ <http://teamerror.wc.it>