

খেলেতে খেলেতে প্রোগ্রামিং শিখা

মো: রাসেল হোসেন

ভূমিকা

প্রোগ্রামিং এর জগতে তোমাকে স্বাগতম। প্রোগ্রামিং খুব কঠিন কিছুনা আবার সহজ ও না। একজন ভালো প্রোগ্রামার হওয়া কঠিন কিছু না তবে খুব সহজ ও না। একজন ভালো প্রোগ্রামার এর চাহিদা সারা পৃথিবী জুড়ে। প্রোগ্রামিং যেমন অনেক মূল্যবান তেমনি প্রোগ্রামিং এর মাঝে আছে অনেক অনেক মজা ও আনন্দ লুকান। একবার যদি কারো প্রোগ্রামিং ভালোলেগে যায় তবে কম্পিউটার এবং প্রোগ্রামিং হবে তার খেলার সাথী, তার বন্ধু, তার পরম আপন। তারকাছে কম্পিউটার হবে তার পৃথিবী। একজন ভালো প্রোগ্রামার হতে গেলে থাকা চাই প্রোগ্রামিং এর প্রতি ভাললাগা এবং প্রোগ্রামিং শিখার প্রচুর ইচ্ছা। নিজের মাঝে থাকা চাই মনোবল। একজন মানুষের কনফিডেন্ট তাকে নিতে পারে সাফল্যের শীর্ষে। আমি বিশ্বাস করি যদি কারো মাঝে নিচের কয়েকটি বৈশিষ্ট থাকে তবে সে ভালো প্রোগ্রামার হতে পারবে:-

১ | প্রোগ্রামিং শিখার প্রচুর ইচ্ছা

২ | কঠোর পরিশ্রম

৩ | ধৈর্য

৪ | অনেক অনেক বেশি প্র্যাক্টিচ

অনেক অনেক প্র্যাক্টিচ ই পারে একজনকে ভালো প্রোগ্রামার বানাতে। ড. মুহাম্মদ জাফর ইকবাল স্যার এর কথা মনেপরে গেলো। কোনো এক কনফারেন্স স্যার বলেছিলেন যদি তুমি ভালো প্রোগ্রামার হতে চাও তবে রাতে ঘুমানোর আগে একবার ভেবো আজ সারাদিনে কয় হাজার লাইন কোড লিখেছো। তারমানে প্র্যাক্টিচ প্র্যাক্টিচ এবং প্র্যাক্টিচ পারে একজন কে ভালো প্রোগ্রামার হিসাবে গড়তে। একজন ভালো প্রোগ্রামার হতে গেলে মুক্ত চিন্তা করা শিখতে হবে। বিভিন্ন ধরনের গণিত নিয়ে, প্রবলেম নিয়ে চিন্তা করতে হবে এবং সমাধান করতে হবে।

বইটি যেভাবে সাজিয়েছি:

এই বইটি কিছুটা এলোমেলো ভাবে সাজিয়েছি যাতে নবীনরা সহজে বুঝতে পারে। যখন যা প্রয়োজন তাই নিয়ে লিখেছি যাতে সহজে বোঝা যায়।

উৎস:

আমার মনের সকল দুঃখ ও আনন্দ কে এবং যে আমাকে অনুপ্রেরনা দিয়েছে তাঁকে।

সূচিপত্র

অধ্যায়	আলোচ্য বিষয়	পৃষ্ঠা নং
প্রথম অধ্যায়	শুরুর পূর্বে কিছু কথা	4-5
দ্বিতীয় অধ্যায়	codeBlocks সম্পর্কে ধারণা	6-9
তৃতীয় অধ্যায়	Header file, key word, data type, statement, variable declaration	10-13
চতুর্থ অধ্যায়	Printf () ও scanf() function সম্পর্কে ধারণা	13-19
পঞ্চম অধ্যায়	Punctuation, আরো data type এবং bit ও byet সম্পর্কে ধারণা	19-24
ষষ্ঠ অধ্যায়	Relational operator logical operator and condition(if else)	24-28
সপ্তম অধ্যায়	Loop (for loop while loop do while loop)	29-38

প্রথম অধ্যায় শুরুর পূর্বে কিছু কথা

মনেকরো তুমি রনি তুমি ক্লাস সিন্ধে পড়। আর আমি রাসেল কোন ক্লাসে পড়ি তা ভুলেগেছি। তুমি কি জানো ভালো প্রোগ্রামাররা মাঝে মাঝে নিজের নাম ভুলে যায়? আমি ভালো প্রোগ্রামারনা তাই ভুলিনি। মনেকরো আমরা বন্ধু। আমি তোমার সাথে গল্প করবো এবং প্রোগ্রামিং নিয়ে খেলবো আর এই গল্প ও খেলার ছলে আমরা প্রোগ্রামিং শিখবো। মাঝে মাঝে আমি তোমাকে প্রশ্ন করবো মাঝে মাঝে তোমাকে বোঁকা বানাবো। চিন্তা করোনা আমি নিজেই প্রশ্নের উত্তর দিব। ভাবছো তোমাকে বোঁকা বানাবো? আমি নিজেই অনেক বড় বোঁকা। তাহলে এবার চলো শুরু করি প্রোগ্রামিং শিখা? দাঁড়াও তুমি কি জানো? কম্পিউটার কি? কম্পিউটার কিভাবে কাজ করে? জাননাতো? তাহলে বলি:-

কম্পিউটার: কম্পিউটার ইলেক্ট্রনিক প্রযুক্তির একটি সার্থক প্রয়োগ। কম্পিউটার হলো একটি গণনা যন্ত্র যা গাণিতিক কাজ যুক্তি বা সিদ্ধান্ত মূলক কাজ খুব দ্রুত করতে পারে। কম্পিউটার শুধু যন্ত্রই নয় আরো অনেক কিছু। একটা ক্যালকুলেটর শুধু গণনা করতে পারে। একটা মিডিয়া প্লেয়ার শুধু গান শুনাতে পারে, TV শুধু ছবি দেখাতে পারে ও গান শুনতে পারে। কিন্তু কম্পিউটার একসাথে অনেক কাজ খুব দ্রুত করতে পারে। এখন বুঝলে তো কম্পিউটার কি?

কম্পিউটার যেভাবে কাজ করে: কম্পিউটার কি ভাবে কাজ করে তা জানার আগে একটু লক্ষ্য করো আমরা (মানুষ) কি ভাবে কাজ করি? কম্পিউটার এর কাজ করার পদ্ধতিকে যদি মানুষের কাজ করার পদ্ধতির সাথে তুলনা করি তবে সহজে বুঝতে পারবো কম্পিউটার কিভাবে কাজ করে। আমরা সাধারণত আমাদের চোখ দিয়ে দেখি, কান দিয়ে শুনি। আমাদের এই দেখা আর শোনার উপর ভিত্তি করে আমাদের মস্তিষ্ক সিদ্ধান্ত নেয় আমাদের কি করতে হবে। তারপর আমাদের হাত পা মস্তিষ্কের সিদ্ধান্তকে বাস্তবে রূপ দেয়। এই যে আমরা দেখি বা শুনি কম্পিউটার এর ভাষায় একে বলে ইনপুট। আমাদের মস্তিষ্ক যে সিদ্ধান্ত নেয় তাকে কম্পিউটার এর ভাষায় বলে প্রসেসিং এবং আমাদের হাত পা মস্তিষ্কের সিদ্ধান্তকে বাস্তবে রূপ দিল, কম্পিউটার এর ভাষায় একে বলে আউটপুট। আমরা যেমন বিভিন্ন তথ্য মনেরাথতে পারি কম্পিউটার ও তাপারে। আমরা যেমন কাজ করার জন্যে আমাদের শরীরের বিভিন্ন অংশ ব্যবহার করে থাকি তেমনি কম্পিউটার কতগুলো অংশ দিয়ে কাজ করে। সেগুলো হলো :-

১. প্রসেসসর (processor)
২. মেমরি (RAM random access memory)
৩. ইনপুট আউটপুট ডিভাইস (key board mouse monitor printer ect.)
৪. ডাটা storage (secondary memory hard disk)
৫. সফটওয়্যার বা প্রোগ্রাম

কম্পিউটার ইনপুট আউটপুট ডিভাইস এর মাধ্যমে ব্যবহারকারীর কাছ থেকে ইনপুট নেয়। সফটওয়্যার বা প্রোগ্রাম এর সহায়তায় প্রসেসসর প্রয়োজনীয় কাজ করে। আউটপুট ডিভাইস এর মাধ্যমে ব্যবহারকারীকে আউটপুট দেখায় এবং ভবিষ্যতে ব্যবহারের জন্যে ডাটা সংরক্ষণ করে রাখে।

আমরা প্রোগ্রামিং কেন শিখবো?

তোমার মাথায় প্রশ্ন এলো না? সফটওয়্যার বা প্রোগ্রাম কি? এইগুলো কি দিয়ে তৈরী? এইগুলো কি কম্পিউটারের কিবোর্ড, মাউস বা মনিটরের মত কিছু? না এইগুলো এমন কিছু না। কম্পিউটার একটা ইলেক্ট্রনিক ডিভাইস। প্রাণ না থাকলে যেমন আমাদের কোনো মূল্য নাই শুধু মাংশ স্তূপ তেমনি সফটওয়্যার বা প্রোগ্রাম ছাড়া কম্পিউটার শুধু মাত্র একটা ইলেক্ট্রনিক যন্ত্র। আর এই সফটওয়্যার বা প্রোগ্রাম তৈরী হয় বিভিন্ন প্রোগ্রামিং language এর মাধ্যমে। আর এই সফটওয়্যার বা প্রোগ্রাম তৈরী করে থাকে বিভিন্ন বড় বড় প্রোগ্রামাররা (programmer)। আমরা প্রোগ্রামিং শিখবো যাতে বিভিন্ন ধরনের প্রোগ্রাম তৈরী করতে পারি। একবার ভাবতো, তুমি যা বলছো কম্পিউটার ঠিক তাই করছে তাহলে তোমার কেমন লাগবে? তুমি কম্পিউটার কে যোগ করতে বললে যোগ করে, বিয়োগ করতে বললে বিয়োগ করে! তুমি যা বলছো কম্পিউটার তাই করে!

নিশ্চয় তোমার কাছে মজা লাগবে? যদি তুমি প্রোগ্রামিং পারো তবে কম্পিউটার তোমার কথা শুনবেই। এত সময় ধরে শুধু শুনছো প্রোগ্রামিং শিখবো, প্রোগ্রামিং শিখবো। প্রোগ্রামিং বলতে আমরা কি শিখবো? প্রোগ্রামিং বলতে আমরা প্রকৃতপক্ষে কম্পিউটার এর ভাষা শিখবো যার মাধ্যমে আমরা কম্পিউটার কে আমাদের মতো করে নির্দেশ দিতে পারবো। নিশ্চয় তোমার মাথায় প্রশ্ন আসছে এইটা আবার কোন ভাষা? বাংলা, ইংরাজি, আরবি না ফার্সি? না এইটা বাংলা, ইংরাজি, আরবি বা ফার্সির মতো তেমন কোনো ভাষা নয়। এইবার একটি ইতিহাস টাইপ কিছু বলব বিরক্ত হয়না প্লিজ। কম্পিউটার এর ভাষা কে বলে মেশিন ল্যাঙ্গুয়েজ। যেটা শুধু শূন্য (0) আর এক (1) নিয়ে লিখা। এখন তোমার মাথায় প্রশ্ন আসতে পারে তার মানে আমাকে কি শূন্য (0) আর এক (1) দিয়ে লিখতে হবে? শুনেই কঠিন লাগছে? ভয় পেয়না, তোমাকে মেশিন ল্যাঙ্গুয়েজ শিখতে হবেনা। কারণ মেশিন ল্যাঙ্গুয়েজ লিখা অনেক জটিল এবং এই ল্যাঙ্গুয়েজ ব্যবহার করে সামান্য কাজের জন্যে প্রোগ্রাম লিখতে অনেক সময় লাগে। তাছাড়া মেশিন ল্যাঙ্গুয়েজ এক কম্পিউটার এর জন্য লিখা প্রোগ্রাম অন্য কম্পিউটার এ কাজ করে না। তাই প্রথম দিকের প্রোগ্রামাররা প্রোগ্রামিং কে সহজ করার জন্যে নতুন এক ল্যাঙ্গুয়েজ উদ্ভাবন করেন যার নাম এসেম্বলি (Assembly) ল্যাঙ্গুয়েজ। এই এসেম্বলি (Assembly) ল্যাঙ্গুয়েজের মাধ্যমে প্রোগ্রামিং এ 0 আর 1 এর পরিবর্তে ইংরাজি শব্দ (English word) ব্যবহার করা হয়। এই ল্যাঙ্গুয়েজ কে মেশিন ল্যাঙ্গুয়েজে রূপান্তর করার জন্যে অ্যাসেম্বলার (Assembler) ব্যবহার করা হতো যা এসেম্বলি ল্যাঙ্গুয়েজ কে মেশিন ল্যাঙ্গুয়েজ এ রূপান্তর করতো। এই সব ল্যাঙ্গুয়েজ কে বলা হয় লওয়ার (lower) লেবেল ল্যাঙ্গুয়েজ। পরবর্তিতে প্রোগ্রামিংকে আরো সহজ করার জন্যে ১৯৫০ সালের পর বিভিন্ন প্রোগ্রামাররা প্রোগ্রামিং কে ইংলিশ ল্যাঙ্গুয়েজে ডেভেলপ করার চেষ্টা করেন যা হাই (high) লেবেল ল্যাঙ্গুয়েজ নামে পরিচিত। এদের মধ্যে c, c++, java, php, oracol অন্যতম। আমরা মূলত C ল্যাঙ্গুয়েজ শিখবো।

কম্পাইলার কি?

রনি আমি প্রথমেই ধারণা করছিলাম তুমি অনেক গাধা, আসলে তুমি তাই। ভাবতে পারো আমি তোমাকে গাধা বললাম কেনো? কারণ তুমি জানলে ইংলিশ এ তোমার মত করে প্রোগ্রাম লিখবে কিন্তু তুমি কি ভুলে গেছো কম্পিউটার 0 আর 1 ছাড়া আর অন্য কোনো কিছু বোঝে না? তুমি কোড লিখবে ইংলিশ এ তাহলে কম্পিউটার বুঝবে কেমন করে? কি ভয় পেলে? তাহলে তুমি প্রোগ্রামার হতে পারবেনো? তোমার মাথায় একটা ব্যাপার কাজ করছে না, যদি এমন কিছু থাকতো যা আমার ইংলিশে লিখা প্রোগ্রাম 0,1 এ কনভার্ট (convert) করতো তাহলে কতো না ভালো হতো। আমি প্রোগ্রামার হতে পারতাম। অবস্যই এমন কিছু আছে যা তোমার ইংলিশ এ লিখা প্রোগ্রাম কে 0, 1 এ কনভার্ট (convert) করবে। আর যে এই কাজ করে থাকে তার নাম কম্পাইলার (compiler)। তার মানে কম্পাইলার (compiler) হলো এমন এক মাধ্যম যা তোমার ইংলিশ এ লিখা প্রোগ্রাম কে 0 1 প্রোগ্রাম এ রূপান্তর করতে পারে। তার মানে তুমি কোডে লিখে কম্পিলার কে দিলে কম্পাইলার তা 0 1 এ কনভার্ট করবে। তাই তোমার চিন্তা করতে হবে না ভয় ও পেতে হবে না যে তুমি প্রোগ্রামার হতে পারবে না। তুমি ইংলিশ এ প্রোগ্রাম লিখ কোনো সমস্যা নাই। তুমি খুব ভালো প্রোগ্রামার হতে পারবে যদি তুমি সি (C) ল্যাঙ্গুয়েজ জানো।

এইবার আমরা সি (C) সম্পর্কে জানবো :-

সি (C) কে বলা হয় মাদার (mother) ল্যাঙ্গুয়েজ। কারণ C ল্যাঙ্গুয়েজ জানলে অন্য যেকোনো ল্যাঙ্গুয়েজ সহজে শিখা যায়। C কে বলা হয় structure programming। নিশ্চয় তোমার মাঝে বিরক্তি চোলে এসেছে। ভাবছো ধূর পরা শুরু করলাম প্রোগ্রামিং শিখবো বলে এখন এইসব কি আজোবাজে পরছি কিছুই বুঝতেছি না। যাও তোমাকে আর বিরক্ত করলাম না এইবার C শুরু করে দিলাম। কি আনন্দ লাগছে? C language এর একটা নির্দিষ্ট structure আছে যা নিচের মতো :-

```
#include<stdio.h>
int main()
{
}
```

যে কোনো প্রোগ্রাম বা প্রবলেম এর সলুসোন কোড লিখতে গেলে তোমাকে প্রথমে ওই কোড লিখতে হবে। তারপর প্রবলেম এর ওপর নির্ভর করে প্রয়োজনীয় কোড দ্বিতীয় বন্ধনীর মধ্যে লিখতে হবে।

দ্বিতীয় অধ্যায়

codeBlocks সম্পর্কে ধারণা

রনি তুমি আবার ও বোঁকামো করলে। একবার ও জানতে চাইলেনা এই কোড গুলো লিখব কোথায়? আমরা কোড লিখব codeBlocks নামের Software এ। যা দেখতে নিচের মতো: -

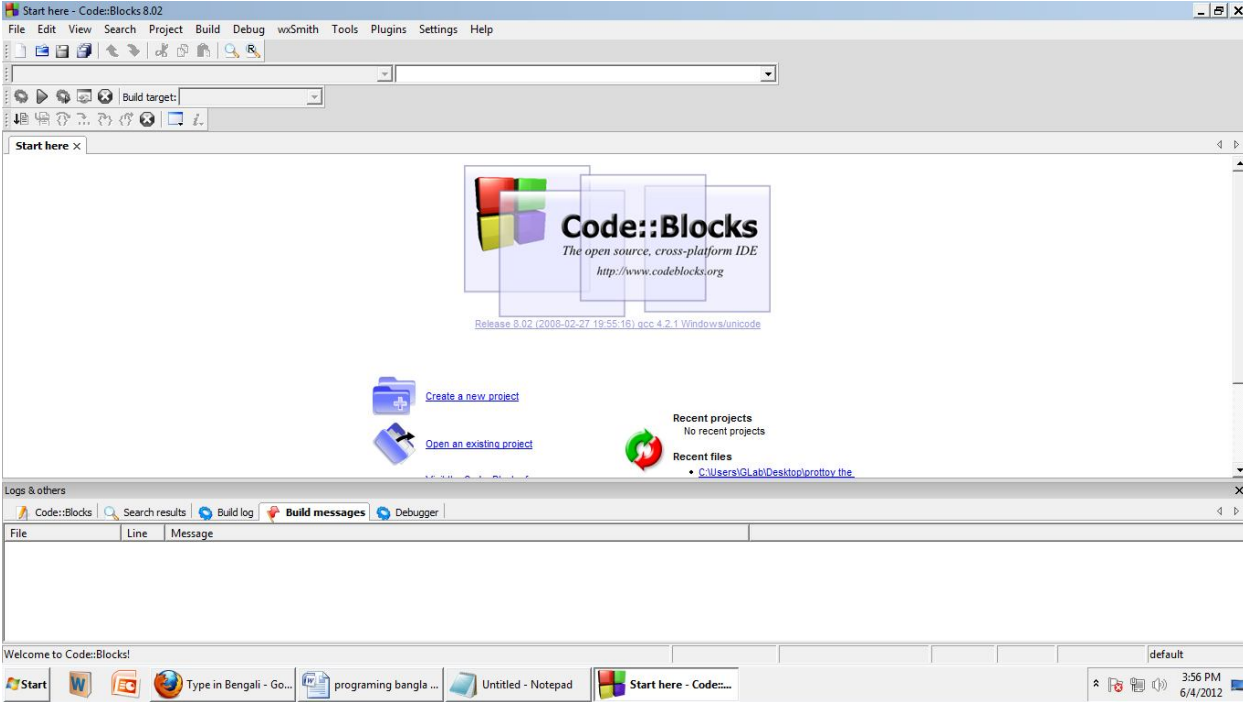


Figure 1: codeBlocks এর open view

প্রথমে তুমি codeBlocks open করো তাহলে উপরের মত (Figure 1) দেখবে। এইবার file এ গিয়ে New empty file এ click (নিচের ছবির মত কাজ করো Figure 2 অথবা `ctrl+shift+n` press) করলে পেয়ে যাবে কোড লিখার জায়গা। রান করার জন্যে F9 বাটন প্রেস করো। মনে রেখো আমরা যে ল্যাপ্সুয়েজ সিখছি তার নাম C তাই আমরা যে কোনো কোড সেভ (save) করবো যে কোনো নাম দিয়ে .c লিখব। যেমন: `print.c` / `russel.c` / `add.c` / `sum.c` ইত্যাদি।

```
#include<stdio.h>

int main()
{
    printf("My name is Russel");
}
```

উপরের কোড লিখ তারপর রান করে দেখো নিচের (Figure 3) মতো দেখবে: -

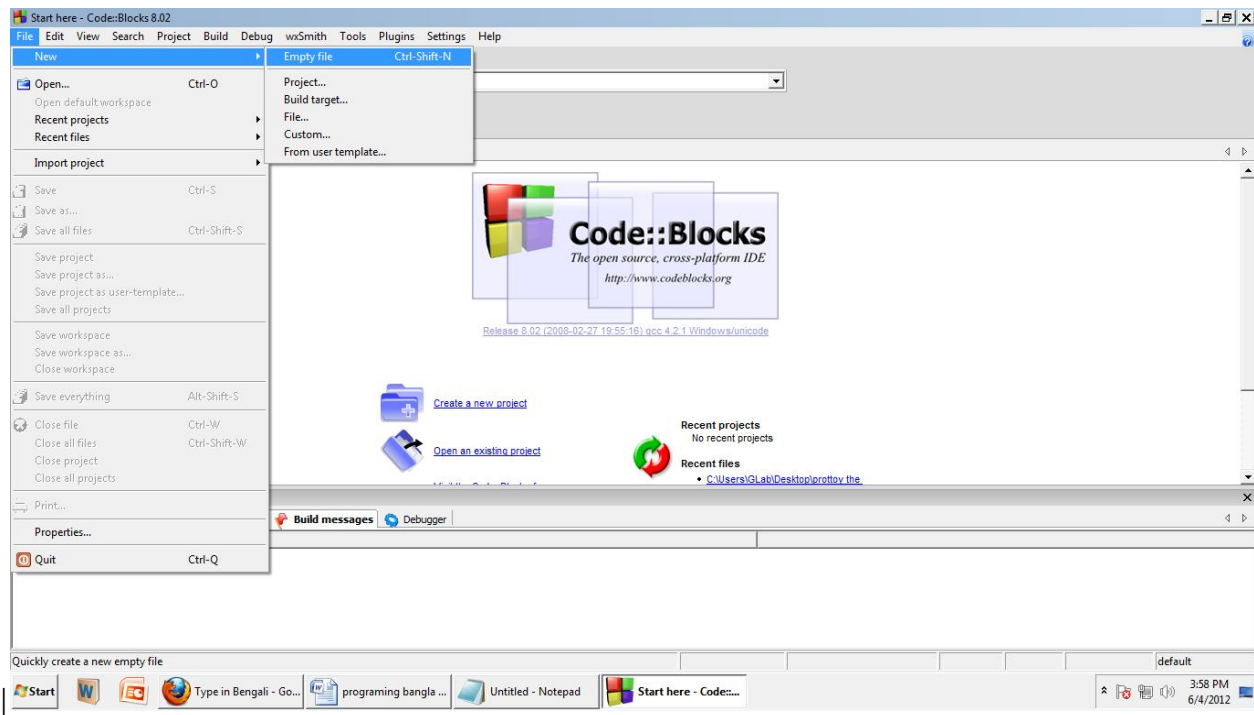


Figure2: কিভাবে codeBlocks এ code লিখার জায়গা আনতে হয়।

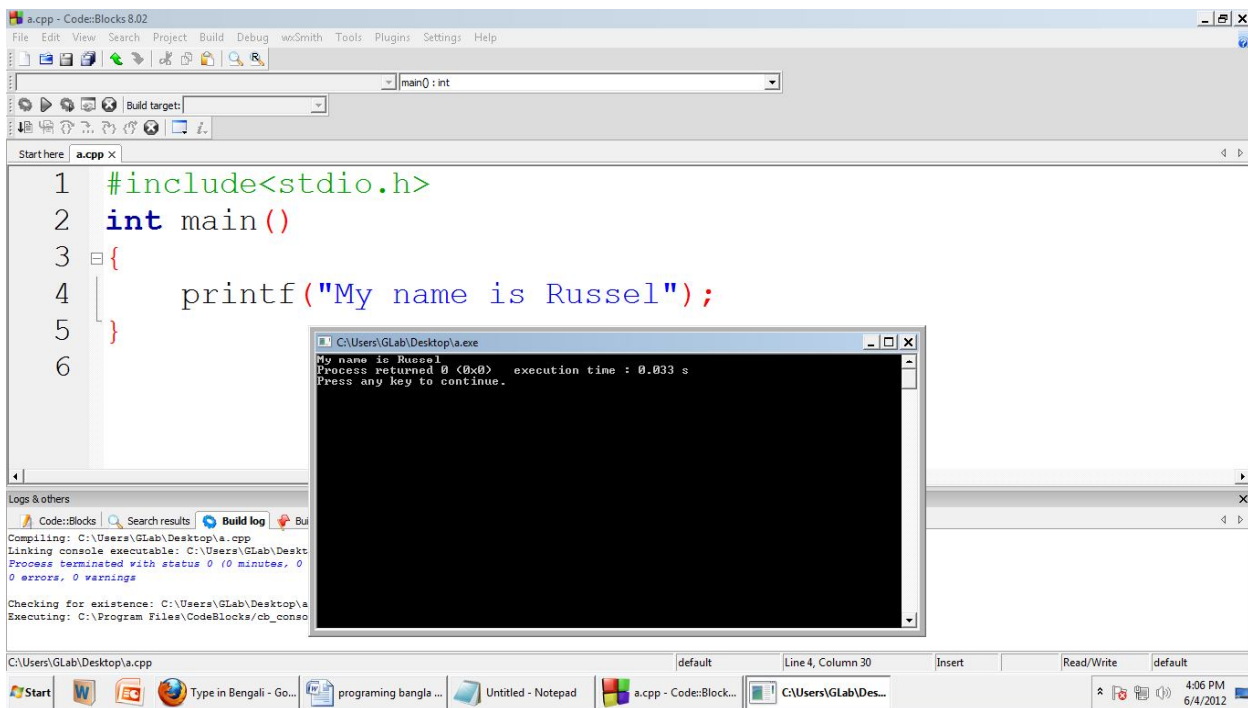


Figure3 : কিভাবে কোড রান করতে হয়

এই কালো window যে দেখাচ্ছে ওই কালো window তে input দেওয়া হয় output ও দেখা যায়। যদি কোনো কারণে কোড রান না করে তবে নিচের মতো কাজ করতে হবে।

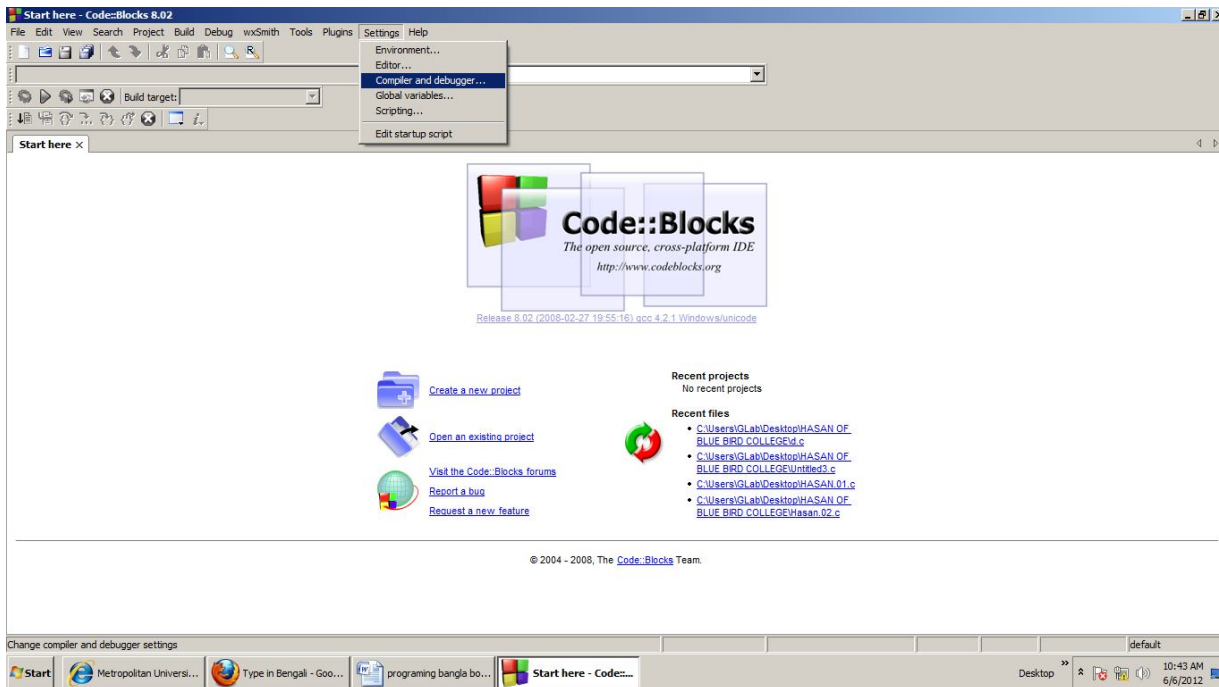


Figure4: settings>> compiler and debugger

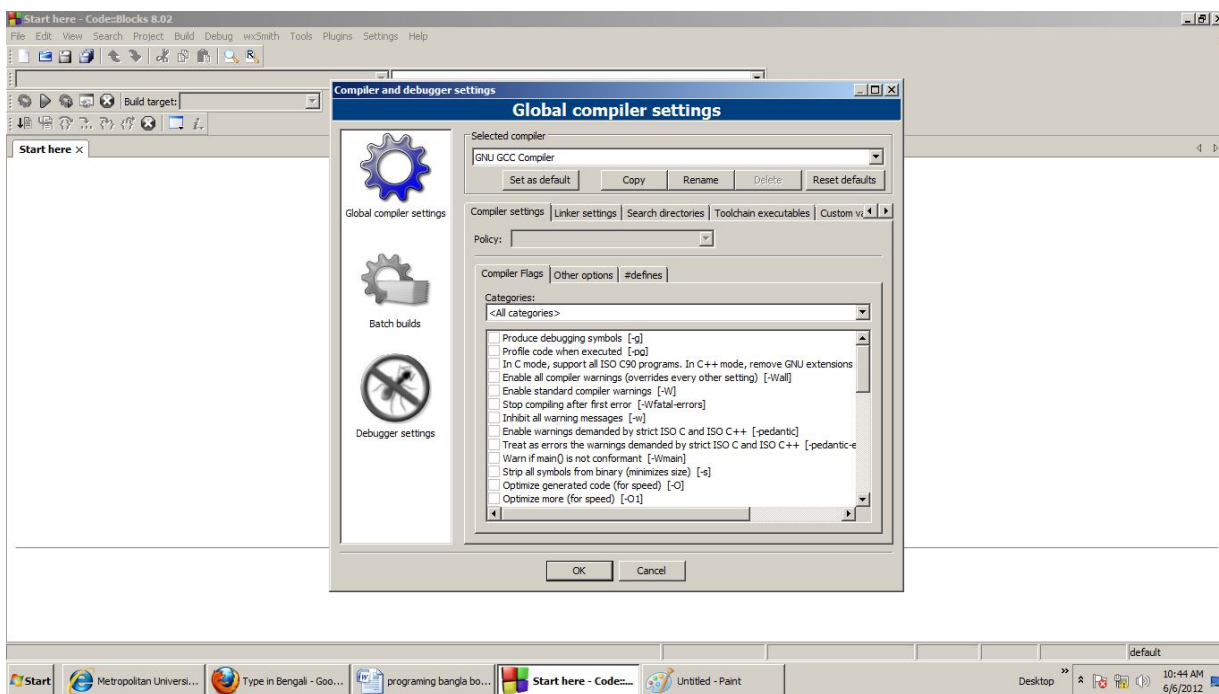
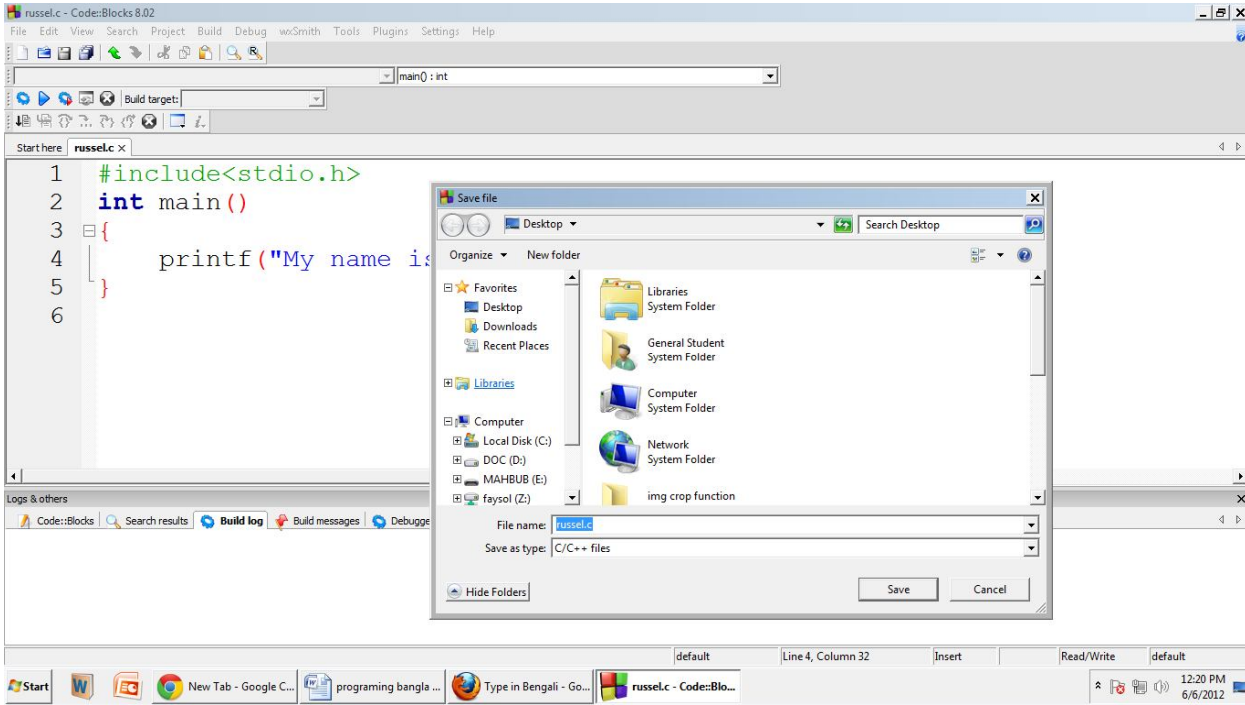


Figure5: compiler and debugger >> reset defaults

Settings এ গিয়ে compiler and debugger এ click করলে একটা window পাবে figure5 এর মতো। এই window এর Reset defaults এ click করলে একটা configuration message দিবে Reset the compiler's setting to the defaults? Ok তে click করলে আর একটা configuration message দিবে Reset the compiler's setting to the defaults? Are you REALLY sure? Ok তে click করলে আর একটা dialog box আসবে এইটাতে ক্লিক করলে কাজ শেষ। এইবার codeBlocks এ ফিরে যাবার জন্যে compiler and debuggers settings (Figure5) এর ok button এ click করো।

ধরো তুমি একটা কোড লিখো এইবার সেভ (save) করবে। তাহলে এর file menu তে গিয়ে file এ click কর। তারপর save এ ক্লিক করলে নিচের মতো widow পাবে। এই window তে file name এ যেকোনো নাম লিখে .C দাও। আমি russel.c লিখছি। তারপর সেভ বাটন এ ক্লিক করো।



তুমি চাইলে যে কোনো folder এ সেভ করতে পারো। সেফ্রে save file window তে একটা folder বানিয়ে তারভিতর করতে পারো। মনেকরো তুমি একটা file এর নাম ভুল লিখো বা নাম পরিবর্তন করতে চাও। তাহলে আগের মতো file menu তে গিয়ে file এ click কর। তারপর save as এ ক্লিক করলে আগের মতো widow পাবে। save file window তে গিয়ে file name এর নাম পরিবর্তন করে লিখো। তারপর সেভ বাটন এ ক্লিক করো।

তৃতীয় অধ্যায়

Header file, key word, data type, statement, variable declaration

নিচের কোডটি লক্ষ্য করো:-

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

```
int main()
```

```
{  
}
```

উপরের কোডের #include<stdio.h> কে বলা হয় header file.

Headers file কি?

মনেকরো তোমার প্রিয় বন্ধু গালিব। গালিবের কতগুলো বন্ধু আছে তারা তোমাকে চিনে না। হঠাৎ একদিন বিকেলে গালিব তোমাকে না বলে তার বাবার সাথে কোথায় যেন গেছে। তুমি বিকেলে একা অনুভব করছো। তাই খেলার মাঠের দিকে একা একা হাঠছো। খেলার মাঠে দেখলে গালিবের বন্ধুরা ক্রিকেটে খেলছে। তুমি গিয়ে বললে আমাকে খেলতে নিবে? তারা বললো আমরা তোমাকে চিনি না তাই খেলতে নিবো না। এখন তুমিই বলো যদি গালিব থাকতো তবে কি বলত? নিশ্চয় বলতো সে আমার বন্ধু তাকে খেলতে নাও। তখন তারা নিশ্চয় তোমাকে চিনবে এবং তোমাকে খেলতে নিবে? তারমানে গালিব থাকায় অপরিচিত মানুষ তোমাকে চিনতে পেরেছে? ঠিক তাই। আমরা যখন কোড লিখব তখন কতগুলো ফাংশন (ফাংশন কি তা পরে জানবো) লিখব। যাদের নির্দিষ্ট কোনো কাজ কম্পাইলার (compiler) এ আগে থেকে নির্ধারণ করা। ওই ফাংশন গুলোর পরিচিতি হিসাবে কাজ করে header file। এমন অনেক header file আছে যা আমরা পরে জানবো। #include<stdio.h> প্রকৃতপক্ষে ব্যবহার করা হয় ইউজার এর কাছ থেকে ইনপুট(input) নেওয়ার ও আউটপুট(output) দেখানোর ফাংশন গুলো ব্যবহার করার জন্যে।

int কে বলা হয় কি ওয়ার্ড (key word) .

কি ওয়ার্ড (key word) কি?

আমরা কম্পাইলার এর নাম শুনেছি যেখানে বিভিন্ন ধরনের ওয়ার্ড (word) আগে থেকে তৈরী করা আছে যাদের কাজ নির্দিষ্ট করে দেওয়া। এইসব ওয়ার্ড কে বলে কি ওয়ার্ড (key word)। আরো সহজ করে বলি ধরো তুমি একটা পিকনিক এর আয়োজন করছো। তোমার বন্ধু গালিবকে দ্বায়িত্ব দিলে বাস ভাড়া করতে, জনি কে দ্বায়িত্ব দিলে বাজার করতে, বাদল কে দ্বায়িত্ব দিলে খাবারের এবং বশির কে চাঁদা তোলার দ্বায়িত্ব দিলে। তাহলে তারা কি করবে? নিশ্চয় নিজ নিজ দ্বায়িত্ব পালন করবে? ঠিক কম্পাইলার এ কতগুলো ওয়ার্ড আগে থেকে তৈরী করা আছে যাদের কাজ নির্দিষ্ট এই সব ওয়ার্ড কে বলে কি ওয়ার্ড(key word)। এইবার বুঝলে কি ওয়ার্ড কি? না বুঝলে ধৈর্য ধরো পরে আরো ভাল করে জানতে পারবে। কতগুলো কি ওয়ার্ড হলো:- int , float, char, double, auto, goto , do, for, while, if, else ,else if, case , long , unsigned, continue, void, break ,return , struct , switch ইত্যাদি। কি ওয়ার্ড এর ব্যবহার আমরা পরে জানবো। উপরের প্রোগ্রাম বা কোডে এ main() বলে একটা শব্দ দেখতে পাইতেছো ওইটা কে বলে ফাংশন (function)। ফাংশন কি তা আমরা পরে জানবো। শুধু মনে রাখো প্রোগ্রামিং এ যে কোনো ওয়ার্ড এর পর () first bracket থাকলে তাকে ফাংশন বলে। মনে করো আমি লিখলাম add()। তার মানে add() একটা ফাংশন। এখন বলি কম্পাইলার main() ফাংশন থেকে কাজ শুরু করে এবং লাইন বাই লাইন বা statement বাই statement কাজ করে। যদি কোনো কারণ বসতো একটা statement এ ভুল পায় তবে সে আর পরের statement এ যাবেনা।

এখন প্রশ্ন হলো statement কি?

যে কোনো লাইন (line) এ একটা সেমিকোলন (;) এর আগ পর্যন্ত কে একটা statement বলে। নিচের কোডটি (code) দেখো :-

```
#include<stdio.h>
int main()
{
    int a,b,c;
    scanf("%d %d",&a,&b);
}
```

এখানে int a,b,c; এর পর একটা সেমিকোলন (;) তারমানে এইটা একটা statement | scanf("%d %d",&a,&b); এর পর সেমিকোলন (;) তার মানে এইটা ও একটা statement | অনেকের ধারণা হতে পারে statement মানে একটা লাইন (line) | ধারণাটা পুরা সত্য না | কারণ নিচের কোড দেখো এইখানে একই লাইনে লিখা int a,b,c; float x,y; কিন্তু পুরাটা একটা statement না এইখানে দুইটা statement । তাই statement আলাদা আলাদা লাইনে ও হতে পারে আবার একই লাইনে একের অধিক statement হতে পারে |

```
int main()
{
    int a,b,c; float x,y;
    scanf("%d %d",&a,&b);
}
```

এইবার একটু data type ও variable declaration সম্পর্কে জানি ?

একটু ভাবতো আমরা যখন কম্পিউটার এ কাজ করি তখন কি করি ? হয় বয়স, বেতন, টাকা ইত্যাদি হিসাবে নাম্বার (number) লিখি অথবা নাম, ঠিকানার জন্যে অক্ষর (A-Z /a-z) লিখে থাকি | কম্পিউটার এ এই নাম্বার বা অক্ষর গুলোকে একত্রে ডাটা (data) বলে | তার মানে আমরা কম্পিউটার এ দুই ধরনের ডাটা (data) নিয়ে কাজ করি

১. Number

২. Character

নাম্বার হিসাবে আমরা দুই ধরনের নাম্বার ব্যবহার করি |

১. পূর্ণ সংখ্যা

২. দশমিক সংখ্যা

এই পূর্ণ সংখ্যা কে বলে integer number, আর দশমিক সংখ্যা কে বলে floating point number, আর বিভিন্ন অক্ষর বা (, . ; " ' \ | + - ! @ # \$ % ^ & * () _ +) ইত্যাদি কে বলে character | integer থেকে নেওয়া হয়েছে int, floating থেকে নেওয়া হয়েছে float, character থেকে নেওয়া হয়েছে char | আমরা যখন variable declare করি তখন int , float , char ব্যবহার করি | data এর বিভিন্ন ধরন কে বলে data type |

হঠাৎ একটা নতুন শব্দ পেলাম variable declaration? ভাবছো এইটা আবার কি? আগে তো কখনো শুনিনি? একটু বুঝিয়ে বলি | আমরা যখন অঙ্ক করি তখন অনেক সময় লিখে থাকি :-

মনেকরি পিতার বয়স X বছর, কলমের দাম ক টাকা , জমির দৈর্ঘ্য a মিটার ইত্যাদি | প্রকৃতপক্ষে এদের মান কি নির্দিষ্ট ? না একএক সময় একএক মান থাকে | আমরা যে এই মনে করে থাকি এইগুলিই প্রকৃতপক্ষে variable বা চলক | আমরা যখন প্রোগ্রাম লিখি তখন অনেকসময় মেমরি (memory) ডাটা রাখতে হয় | অনেকসময় ডাটা আমরা নির্দিষ্ট করে দিয়ে থাকি আবার অনেক সময় ব্যবহারকারীর (user)

কাছ থেকে ইনপুট আকারে নিয়ে থাকি। ডাটা আমরা যেখান থেকে নেইনা কেন প্রোগ্রাম চলাকালে ডাটাগুলো রাখার জন্যে মেমরিতে স্থান দখল করতে হয়। এই স্থান দখল করতে নাম দিতে হয়। এই নাম দেওয়া কে বলে variable declaration। ধারণাটা ক্লিয়ার হলনাতো? আরো সহজ করে বলি। আমরা (মানুষরা) থাকার জন্যে কি করি? বাড়ি তৈরী করি। বাড়িতে কতগুলো ঘর থাকে এবং কোন ঘরে কে থাকবে তা নির্দিষ্ট করা হয়। ধরো এক রুম এ মা বাবা, এক রুম এ বরো অপু, একরুম এ আমি আর ভাইয়া থাকবো। ঠিক তেমনি কম্পিউটার এর মেমরিতে ডাটা রাখার জন্যে কতগুলো স্থান নির্ধারণ করা হয়। এই স্থান নির্ধারণ করাকে বলে variable declaration। আরো ক্লিয়ার হওয়ার জন্যে একটু ধৈর্য ধরে নিচের কোড দেখো :- তার আগে মনেকর নিচের ছবি টি কম্পিউটার এর মমরী যেখানে অনেকগুলো ঘর আছে।

ছবি : কম্পিউটার এর মেমরি

এবার নিচের কোড দেখো:-

```
#include<stdio.h>
int main()
{
    int a, b;
    float x, y;
    char p, q;
}
```

int a; লিখা মানে মেমরিতে একটা ঘরের নামা দেওয়া a। তারমানে মেমরিতে একটা ঘর দখল হবে a নামে যেখানে int type data রাখা যাবে। int b; লিখা মানে মেমরি তে একটা ঘরের নামা দেওয়া b। তারমানে মেমরিতে একটা ঘর দখল হবে b নামে যেখানে int type data রাখা যাবে। float x; লিখা মানে মেমরিতে একটা ঘরের নামা দেওয়া x। তারমানে মেমরিতে একটা ঘর দখল হবে x নামে যেখানে float type data রাখা যাবে। float y; লিখা মানে মেমরি তে একটা ঘরের নামা দেওয়া y। তারমানে মেমরিতে একটা ঘর দখল হবে y নামে যেখানে float type data রাখা যাবে। char p; লিখা মানে মেমরি তে একটা ঘরের নামা দেওয়া p। তারমানে মেমরি তে একটা ঘর দখল হবে p নামে যেখানে char type data রাখা যাবে। char q; লিখা মানে মেমরি তে একটা ঘরের নামা দেওয়া q। তারমানে মেমরি তে একটা ঘর দখল হবে q নামে যেখানে char type data রাখা যাবে। নিচের ছবিটি দেখো মেমরীর ভিন্ন ভিন্ন ঘরে নাম দেওয়া। তার মানে মেমরিতে যে জায়গা খালি পাবে সেই জায়গা দখল করবে। আর এই ভাবে int a; int b; float x; float y; char p; char q; লিখা কে বলে variable declaration. আর variable এর নাম হিসাবে শুধু a-z, A-Z লিখা যাবে তানয় তুমি চাইলে যে কোনো word, name বা sentence লিখতে পারো। তবে কতগুলো শর্ত মেনে লিখতে হবে।

উপরের মত প্রোগ্রাম লিখলে মেমরি নিচের মত হবে :-

a				z					y
x			b			q			p

ছবি : কম্পিউটার এর মেমরি

শর্ত গুলো হলো :-

১. যে word লিখবে তা key word হওয়া যাবে না।

২. word এর শুরুতে সংখ্যা(number) থাকতে পারবে না।

৩. underscore(_) এবং dollar sign(\$) ছাড়া অন্য character যেমন (, . / ; : ' " \ | + - * & ^ % # @ ~ `) থাকতে পারবে না।

নিচের টেবিল এ কতগুলো valid invalid variable দেওয়া হলো: -

সঠিক	ভুল	ভুল এর কারণ
Var_1	Var-1	থাকতে পারবে না
My_age	My#age	# থাকতে পারবে না
Id_no	id@no	@ থাকতে পারবে না
myname	My name	Space থাকতে পারবে না
Do_it	do	Key word থাকতে পারবে না

আশা করি variable declaration ক্রিয়ার হবার কথা। এরপর ও যদি ক্রিয়ার না হয় হতাশ হওয়ার কিছুই নাই। আরো কিছুসময় পড়তে থাক এবং practice করো আসা করি ক্রিয়ার হবে।

চতুর্থ অধ্যায়

Printf () ও scanf () function সম্পর্কে ধারণা

printf() function সম্পর্কে ধারণা :

printf() একটি function। যার কাজ compiler এ আগে থেকে নির্ধারিত করে দেওয়া এবং এর কাজ হলো ব্যবহার করি (user) কে output দেখানো বা computer screen এ কিছু show করা।

printf() function লিখার নিয়ম : printf("যা লিখতে চাও");

এইখানে **যা লিখতে চাও** বলতে বুঝানো হয়েছে তুমি user কে কি দেখাতে চাও। ধরো আমি একটা প্রোগ্রাম লিখব যা রান (run) করলে computer screen এ দেখাবে **My name is Russel**। তাহলে আমি কি করবো? তেমন কঠিন কিছুই না শুধু নিচের মত কোড লিখব:-

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

```
int main()
```

```
{
```

```
    printf("My name is Russel");
```

```
}
```

এই code রান করলে দেখাবে My name is Russel। তার মানে printf() function এর "" এর ভিতর যা লিখবে তাই computer screen দেখাবে। এইবার তুমি নিজের মত যা ইচ্ছা তা লিখে দেখতে পারো।

এবার একটু লক্ষ্য কর আমি এটি প্রোগ্রাম লিখতে চাই যার কম্পাইল করলে আউটপুট হবে নিচের মতো:-

My name is Russel

I am a new programmer

তাহলে কোড কেমন করে লিখবো? নিচের মত করে?

```
#include<stdio.h>
int mian()
{
    printf("My name is Russel");
    printf("I am a new programmer");
}
```

কি ভাবছো? উপরে যে কোড লিখছি তা যদি রান করি তাহলে আউটপুট কি যেমনটা চাইলাম তেমনটা পাবো? না উপরের কোড রান করলে আউটপুট হবে My name is Russell am a new programmer। অবাক হলে কোড print করলাম দুই লাইন কিন্তু আউটপুট এলো এক লাইনে এইটা কেমন হোলো? হাঁ কম্পিউটার অনেকটা বোকা। সে নিজে থেকে কিছুই করতে পারেনা। সে এইটা ও জানেনা যে এইখানে দুইটা লাইন আছে আমাকে দুই লাইন প্রিন্ট করতে হবে। সে এতো গাঁধা যে সে দুই লাইন কে এক লাইনে প্রিন্ট করছে। তাই তাকে বলে দিতে হয় প্রথম লাইন প্রিন্ট করার পর নিচের লাইনে গিয়ে প্রিন্ট করতে। আর এই কাজটি করে থাকে \n। বুঝলেনাতো? নিচের কোড দেখো:-

```
#include<stdio.h>
int mian()
{
    printf("My name is Russel\n");
    printf("I am a new programmer");
}
```

অথবা নিচের কোড দেখো:-

```
#include<stdio.h>
int mian()
{
    printf("My name is Russel\nI am a new programmer");
}
```

কি বুঝলে? যেখান থেকে নিচের লাইনে লিখতে হবে ঠিক সেখানে \n লিখতে হবে। কি শিখে ফেললে? তাহলে নিচের আউটপুট পেতে কোড কি হবে নিজে লিখো আমি আর বলতে পারবোনা।

আউটপুট হবে :-

He

Is

A

Boy

এতো সময় তো শুধু sentence প্রিন্ট করলাম কিন্তু number প্রিন্ট করাতো হলোনা | number প্রিন্ট করতে গেলে কি করবে? তেমন কিছুই না আমরা data type শিখে এসেছিনা? একটু মনে করো এবং মনে রাখো যেকোনো data input নিতে বা output দেখাতে নিচের টাবিল মনে রাখতে হবে:-

Variable integer type হলে	%d ব্যবহার করতে হবে
Variable floating type হলে	%f ব্যবহার করতে হবে
Variable character type হলে	%c ব্যবহার করতে হবে

কি ব্যাপারটা ক্লিয়ার হলনা তাইতো? ব্যাপারনা নিচের কোড দেখো:-

```
#include<stdio.h>
int mian() {
    int a,b,c;    a=5;  b=8;  c=a+b;
    printf( "%d",c) ;
}
```

এই কোডটি যদি রান করি তবে প্রিন্ট করবে 13 | লক্ষ করো প্রথম লাইন int a,b,c; হলো variable declaration | তারমানে এই লাইন পরে মেমরীতে a,b,c নামে জায়গা দখল হবে | ২য় লাইনে a=5 তারমানে মেমরীতে যে জায়গার নাম a সেখানে value হবে 5 | ৩য় লাইন এ b=8 তারমানে মেমরীতে যে জায়গার নাম b সেখানে value হবে 8 | ৪র্থ লাইন এ c=a+b তারমানে মেমরীতে যে জায়গার নাম c সেখানে value হবে 13 | এবার printf() function এর দিকে তাকাও | কি দেখো? Printf() function এর "" এর ভিতর %d তারমানে integer type value প্রিন্ট করবে যা c variable এর ভিতর আছে | ক্লিয়ার হলনাতো? আরও সহজ করে বলি যে type data প্রিন্ট করতে চাও(integer হলে %d float হলে %f character হলে %c) তা printf() function "" এর ভিতর বলতে হবে তারপর কমা (,) দিয়ে যে variable এর ভিতর থেকে প্রিন্ট করতেহবে তা বলেদিতে হবে| আর একটু ক্লিয়ার হয়ার জন্যে নিচের কোড ভালকরে লক্ষ কর:-

কোডটি রান করলে output হবে 5 7.12 A | কারণ printf() function লক্ষ করো প্রথমে %d তারমানে একটা integer value প্রিন্ট করবে যা a variable এর ভিতর আছে | তারপর %f তারমানে একটা floating value প্রিন্ট করবে যা b variable এর ভিতর আছে| তারপর %c তারমানে একটা character value প্রিন্ট করবে যা c variable এর ভিতর আছে |

```
#include<stdio.h>
int mian()
{
    int a;
    float b;
    char c;
    a=5; b=7.12; c='A';
    printf( "%d %f %c",a,b,c) ;
}
```

যদি printf() function এভাবে লিখা হতো printf("%d %f %c",c,a,b) ; তাহলে output উলটাপালটা হতো কারণ printf() function এর প্রথমে %d তারমানে একটা integer value প্রিন্ট করবে কিন্তু "" এর পর প্রথম variable c যা character type | %d এরপর %f যা floating type value প্রিন্ট করবে কিন্তু "" এর পর ২য় variable a যা integer type | %f এরপর %c যা character type value প্রিন্ট করবে কিন্তু "" এর পর ৩য় variable b যা floatine type | তারমানে printf() function এর "" এর ভিতর যে সিরিয়াল থাকবে কমা(,) এর পর সেই সিরিয়াল এ variable লিখতে হবে | আশা করি printf() function মোটামোটি ক্লিয়ার হলো আরও ক্লিয়ার হবে আরও পরে এবং আরও practice এর পর|

scanf() function সম্পর্কে ধারণা:

scanf() একটি built in function | যা কম্পাইলার (compiler) এ আগে থেকে তৈরি করা এবং যার কাজ আগে থেকে নির্ধারিতো করা | printf() function ব্যবহারকারী (user)কে output দেখায় আর scanf() function এর কাজ

ব্যবহারকারী (user) এর কাছ থেকে input নেওয়া | scanf() function লিখার নিয়ম অনেকটা printf() function এর মতো | তবে কিছুটা পার্থক্য আছে | scanf() function লিখার নিয়ম নিচের মত:-
scanf("যে type value নিতেচাও",& যে variable এর ভিতর value রাখতে চাও);
আশা করি বুঝতে পারছ কেমন করে scanf() function লিখতে হয়| তারপর আর ক্রিয়ার হয়ার জন্য নিচের কোড দেখো যেখানে আমি দুইটা input নেবো user এর কাছ থেকে এবং user কে যোগফল দেখাব|

```
#include<stdio.h>

int main()
{
    int num1,num2,sum;
    scanf("%d %d",&num1,&num2);
    sum=num1+num2;
    printf("%d",sum);
}
```

উপরের কোডের দিকে তাকাও আমি scanf() function এর "" এর ভিতর ২টা %d লিখছি তার কারণ আমি user এর কাছ থেকে ২টা integer number input নেবো এবং ১ম input হিসাবে user যে value দিবে তা num1 variable এর ভিতর রাখবো, আর ২য় input হিসাবে user যে value দিবে তা num2 variable এর ভিতর রাখবো | তারমানে user যদি input হিসাবে 15 30 দেয় তবে output হিসাবে 45 দেখাবে |

এই কোডটা একটু বিশ্লেষণ করে দেখি ? মি. রনি উপরের কোডের দিকে তাকাও|

আমরাজানি কম্পাইলার(compiler) main() ফাংশন থেকে কাজ শুরু করে এবং লাইন বাই লাইন কাজ করে? তাহলে কোডের দিকে চেয়ে দেখো main() ফাংশন এর ভিতর {} এর শুরুতে অর্থাৎ ১ম লাইন int num1,num2,sum; | তারমানে কম্পিউটারের মেমরীতে num1,num2 এবং sum নামে ৩টা জায়গা দখল করবে? ২য় লাইনে লিখা

scanf("%d %d",&num1,&num2); scanf() ফাংশন এর ভিতর প্রথমে %d তারমানে user এর কাছ থেকে integer type number input নেওয়ার জন্য অপেক্ষা করবে| ধরো user ১ম input হিসাবে 55 দিলো| তাহলে 55 num1 variable এর ভিতর রাখবে| তারমানে num1=55 | ধরো user ২য় input হিসাবে 65 দিলো | তাহলে 65 num2 variable এর ভিতর রাখবে| তারমানে num2=65 হবে| তৃতীয় লাইন sum=num1+num2 তারমানে sum=55+65 বা sum=120 হবে| এবার চতুর্থ লাইন printf("%d",sum) এবং printf() ফাংশন এর ভিতর %d তারমানে user কে integer number show করবে যা sum নামের variable এর ভিতর আছে|

আশাকরি scanf() ফাংশন সম্পর্কে ধারণা মোটামোটি ক্রিয়ার| আরও ক্রিয়ার হয়ার জন্য আরেকটি কোড লিখি যা একসাথে integer float এবং character input হিসাবে নিবে আর output হিসাবে float character এবং integer দেখাবে| রনি তুমি কি ভাবতে পারছো কোড কেমন হবে? পারলেনা তো? ঠিক আছে আমি লিখছি তুমি ভাল করে লক্ষ কর|

```
#include<stdio.h>

int main()
{
    int input1;
```



```

float input2;
char input3;
scanf("%d %f %c",& input1,& input2,& input3);
printf("%f %c %d", input2, input3, input1);
}

```

এই কোডটি কিভাবে কাজকরবে তা খুজে বেরকরা এবং কোড বিশ্লেষণ করা তমার নিজের দায়িত্ব। আমি আর পারবনা।

এইবার কিছু operator নিয়ে কথা বলা যেতে পারে:

Operator নাম শুনে মনেকরছো যা কনোেকিছু অপারেট বা চালনা করে থাকে? না Operator বলতে এমন কিছু বুঝায় না। আমরা অংক করতে যে যোগ, বিয়োগ, গুণ, ভাগ চিহ্ন ব্যবহার করেথাকি সেই চিহ্ন কে মূলতো Operator বলা হয়। কম্পিউটার এ গাণিতিক Operator গুলোহলো:

+	যোগ করে
-	বিয়োগ করে
*	গুণ করে
/	ভাগ করে
%	ভাগশেষ দেয়

রনি তোমার কি মনে আছে ক্লাস ফোর এ সরল করার সময় পড়ছিলে সরল অংক করার জন্য BODMAS মেনে চলতেহতো? কম্পিউটার ও BODMAS মেনেচলে। তুমি মনেহয় ভুলেগেছো BODMAS মানে কি। ঠিক আছে আমি মনিকরিয়ে দিতেছি: B= ব্রাকেট O= এর D= ভাগ M= গুন A= যোগ S= বিয়োগ। কি মনে পরছে? এইবার ধর তুমি দুইটা নাম্বার এর গড়(average) বেরকরবে তাহলে নিচের কনটি অনুসরণ করবে? তুমিই ঠিক করো। মনেকর variable declar করলাম float a, b, avg; এবং a=5; b=7; লিখলাম।

1. avg=a+b/2
2. avg=(a+b)

নিশ্চয় ১নং না নিয়ে ২ নং নেবে? কারন যদি ১নং নাও তবে answer কি আসবে? avg=5+7/2 বা avg=5+3.5 বা avg=8.5 আর যদি ২নং নাও তবে answer হবে avg=(5+7)/2 বা avg=12/2 বা avg=6 | এবার তুমি বল কোনটা সঠিক?আচ্ছা বলতো আমি গড়(average) বেরকরার জন্যে variable float নিলাম কেন? বলতে পারলেনা তো? কারন বলি ভাগ করলে অনেক সময় ভাগফল দশমিক সংখ্যা হতেপারে। আর যেহেতু দশমিক সংখ্যা রাখার জন্যে float type variable দরকার তাই avg variable কে float tyep নিয়েছি। এখন বুঝলে? যদি দেখো উত্তর দশমিক সংখ্যা হতেপারে তাখন যে variable এর ভিতর উত্তর রাখবে তা float tyep variable এর ভিতর রাখবে। মনেথাকবে?

রনি আমি এখন আশাকরতে পারি তুমি এখন variable declare করতে পারো, user এর কাছথেকে input নিতে পারো এবং user কে output দেখাতে পারো। তারমানে তোমাকে যদি বলি user এর কাছথেকে input নিবে তারপর যোগ, বিয়োগ, গুন, ভাগ, ভাগশেষ গড় ইত্যাদি ক্যালকুলেসন করবে তাহলে নিশ্চয় পারবে? তুমি যেহেতু ক্লাস সিদ্ধ এ পড়ো তারমানে বীজগণিত এর সব সূত্র জানো? যদি ভুলে যেয়ে থাকো তবে মনে করে দিয়ে।

1. $(a+b)^2 = a^2 + 2ab + b^2$
2. $(a+b)^3 = a^3 + 3a^2b + 3ab^2 + b^3$ কি এই দুইটা দেখে বাঁকি গুলো মনে পরছে? তাহলে তোমাকে কত গুলো কাজ দেই। তুমি user এর কাছথেকে input হিসাবে a b নিবে তারপর প্রতিটি সূত্রের জন্য প্রগ্রাম লিখবে যাতে প্রতিটি সূত্রের উত্তর পাওয়া

যায়।আমি তোমার সুবিধার জন্যে একটা সূত্রের জন্যে কোড লিখে দিলাম বাঁকি গুলো তুমি নিজে করবে। $(a+b)^2 = a^2 + 2ab + b^2$ এর কোড নিচে লিখে দিলাম।

```
#include<stdio.h>
int main()
{
    int num1,num2,ans;
    scanf( "%d %d",&num1,&num2) ;
    ans= num1* num1+2* num1* num2+ num2* num2;
    printf( "%d",ans) ;
}
```

রনি আমি জানি তুমি যেহেতু এতটুকু পড়ে এসেছো তারমানে তুমি variable declar করতে পারো, user এর কাছ থেকে input নিতেপারো user কে output দেখাতে পারো। আমি তোমার printf() function আরো ক্লিয়ার করার জন্যে নিচে সূত্রের কোড লিখে দিব যার output হবে নিচের মতো।

সূত্র: $(a+b)^2 = a^2 + 2ab + b^2$

user input হিসাবে a এবং b এর value দিবে যা integer number। আমার program output হিসাবে দিবে $a^2 + 2ab + b^2$ এর মান। কিন্তু যার output হবে নিচের মতো:-

$a^2 + 2 * a * b + b^2 = P$

ধরো user input হিসেবে দিলো 3 2 তাহলে output হবে একম 3²+2*3*2+2²=25 |

কি খুব কঠিন লাগছে তাহলে নিচের কোড মনোযোগ দিয়ে দেখো :-

```
#include<stdio.h>
int main()
{
    int num1,num2,ans;
    scanf( "%d %d",&num1,&num2) ;
    ans=num1*num1+2*num1*num2+num2*num2;
    printf( "%d^2+2*%d*%d+%d^2=%d",num1,num1,num2,num2,ans) ;
}
```

এখানে num1 বলতে a num2 বলতে b আর ans বলতে p বুঝিয়েছি। printf() function লক্ষ্য করো প্রথমে %d বলতে সূত্রের a বা code এর num1 variable বুঝিয়েছি। দ্বিতীয় %d বলতে সূত্রের 2ab এর a বুঝিয়েছি যাকোড এর num1 variable এর ভিতর আছে। তৃতীয় %d বলতে সূত্রের 2ab এর b বুঝিয়েছি যাকোড এর num2 variable এর ভিতর আছে। চতুর্থ %d বলতে সূত্রের b বুঝিয়েছি যাকোড এর num2 variable এর ভিতর আছে। আর পঞ্চম %d বলতে সূত্রের solution বা answer বুঝিয়েছি যাকোড এর ans variable ভিতর আছে।

অনুশীলনি:

রনি তুমি জানো ভালো প্রোগ্রামার হতে গেলে নিজে অনেক অনেক প্রবলেম সলভ করতে হয়। তাই আমি তোমার জন্যে কতগুলো প্রবলেম দিলাম যেগুলো তুমি নিজে সলভ করবে। আমি জানি তুমি পারবে।

১। তুমি একটা program লিখবে যা রান করলে ইউজার কে বলবে Enter a number: | ইউজার number দিলে দেখাবে you enter X | এখানে X বলতে ইউজার যে number দিবে তা | মনেকর ইউজার 55 দিয়ে enter দিলো | তাহলে তোমার প্রগ্রাম দেখাবে you enter 55 |

২। তুমি একটা প্রগ্রাম লিখবে যা রান করলে ইউজার কে বলবে Enter a number1: ইউজার একটা number দিয়ে enter দিলে বলবে Enter a number2: ইউজার আর একটা number দিয়ে enter দিলে বলবে the sum of your two number is X | এখানে X বলতে ইউজার যে number দুটি দিবে তার যোগফল।

৩। তুমি একটা প্রগ্রাম লিখবে যা রান করলে ইউজার কে বলবে Enter a number1: ইউজার একটা number দিয়ে enter দিলে বলবে Enter a number2: ইউজার আর একটা number দিয়ে enter দিলে বলবে $P+Q=X$ | এখানে P বলতে ইউজারের দেওয়া ১ম number, Q বলতে ইউজারের দেওয়া ২য় number এবং X বলতে ইউজার যে number দুটি দিবে তার যোগফল। উদাহরণস্বরূপ মনেকরো user তোমার program রান করলো এবং input দিলো নিচের মতো তাহলে output হবে নিচের মতো

Enter a number1: 55

Enter a number2: 125

55+125=180

৪. প্রতিটি বীজগণিত এর সূত্রের জন্যে কোড লিখো হবে output হবে formate হবে সূত্রের মতো | এ ক্ষেত্রে তুমি উপরের কোড এর সাহায্য নিতে পারো।

উদাহরণস্বরূপ:

১। $a^3 + 3a^2b + 3ab^2 + b^3$

২। $a^3 - 3a^2b + 3ab^2 - b^3$

পঞ্চম অধ্যায়

Punctuation, আরো data type এবং bit ও byet সম্পর্কে ধারণা

রনি আমি জানি যেহেতু তুমি উপরের সব অধ্যায় পড়ে এসেছো তারমানে তুমি variable declare করতে পারো user input নিতে পারো user output দেখাতে পারো। এখন তোমাকে নতুন কিছু বলবো যা আমি আগে তোমাকে বলতে পারতাম কিন্তু বলিনি কারণ আমি জানি তখন বললে তোমার কাছে কঠিন লাগতো। মনেকরো আমি variable declar করলাম int a,b,x,y,p,q; তারপর লিখলাম a=5; b=7; তারমানে a এর value 5 b এর value 7 | এখন লিখলাম x=a; y=b; p=y; q=x;

তাহলে এখন বলো p=? q=? বলতে পারলেনা তো? আমি বলে দিতেছি x=a তারমানে x এর ভিতর a এর value আবার y=b তারমানে y এর ভিতর b এর value | যেহেতু p=y তারমানে p এর ভিতর y এর value আর y এর ভিতর b এর value তারমানে p এর ভিতর 7 | আবার যেহেতু q=x তারমানে q এর ভিতর x এর value আর x এর ভিতর a এর

value তারমানে q এর ভিতর 5 | তারমানে p=7 q=5 |

এইবার নিচেরটি নিজে বলো আমি আর পারবনা।

int a,b,c; a=15; b=35; c=a; a=b; b=c: এইবার বলো a=? b=?

তুমি $ax^2+bx+c=0$ এই equation এর সাথে পরিচিত হয়ে থাকবে |

যেখান থেকে X এর মান বের করার formula $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ | আমরা প্রোগ্রামিং এর মাধ্যমে x এর মান বের করবো। এই equation solve করতে গেলে আমাদের প্রথমে $\sqrt{b^2 - 4ac}$ এর value বের করতে হবে | $\sqrt{b^2 - 4ac}$ এর value বের করতে গেলে $b^2 - 4ac$ value বের করতে হবে | ধরলাম $p = b^2 - 4ac$ এবং $q = \sqrt{p}$ তাহলে উপরের দাড়াবে $x = \frac{-b \pm q}{2a}$ | এখন আমি কি নিচের মতো করে লিখতে পারি? $x1 = (-b+q)/2a$ এবং $x2 = (-b-q)/2a$ | তাহলে x1 এবং x2 এর ভিতর answer থাকবে। কিন্তু তুমি কি একবার ও ভেবে দেখেছো কম্পিউটার এ squar root ($\sqrt{\quad}$) চিহ্ন নাই। তাহলে উপায়? উপায় একটা নিশ্চয় আছে আর তা হলো sqrt() ফাংশন | এইটা লিখতে হয় sqrt(X) | এখানে X হলো তুমি যার squar root ($\sqrt{\quad}$) করতে চাও | আর এই sqrt() ফাংশন কে কম্পাইলারের সাথে পরিচয় করে দিয়ার জন্যে header file হিসাবে লিখতে হয় #include<math.h>

এইবার নিচের কোডটি খুব ভালো করে লক্ষ্য করো:-

```
#include<stdio.h>
#include<math.h>
{
    float a,b,c,p,q,x1,x2;
    scanf("%f %f %f",&a,&b,&c);
    p=b*b-4*a*c;
    q=sqrt(p);
    x1=(-b+q)/(2*a);
    x2=(-b-q)/(2*a);
    printf("%f %f",x1,x2);
}
```

এইবার একটু বিট(bit) এবং বাইট (byet) সম্পর্কে জানবো:

রনি তোমাকে আগেই বলছি কম্পিউটার 0 এবং 1 ছাড়া আর কিছুই বোঝেনা। এই 0 এবং 1 এর প্রতিটিকে বলে বিট। আর ৮ বিট মিলে হয় এক বাইট। এক বাইট বলতে ৮ টা 0 বা ৮ টা 1 অথবা ৮ টা 0 ও 1 মিশ্রিত হতে পারে। এখন তোমাকে বলি আমরা যে variable declar করি তা মেমরীতে নিচের মত জায়গা নেয়।

integer 2 byet, floating 4byet এবং character 1 byet জায়গা নেয়।

Computer number কে কিভাবে মেমরীতে রাখে?

ধরো তুমি variable declar করলে `int num;` এবং `num=5` লিখলে | তারমানে কম্পিউটার 5 কে মেমরীতে 00000000000000101 রাখবে। কারণ 5 এর বাইনারি 101 যেহেতু variable integer তাই মেমরীতে 16 বিট(2 byet) জায়গা নেবে। ডান দিক থেকে পড়লে এর 16 নং bit কে বলা হয় সাইন বিট (sign bit) | অর্থাৎ 16 নং bit নির্ধারণ করে নাম্বার পজেটিভ না নেগেটিভ। তুমি যদি positive number রাখো তবে 16 নং bit হবে 0 আর যদি negative number রাখো তবে 16 নং bit হবে 1 | যদি তুমি `num= -5` রাখো তবে -5 কে মেমরীতে 10000000000000101 রাখবে। তারমানে তুমি integer variable এর মাঝে $2^{15}=32768$ বা -32768 পর্যন্ত রাখতে পারবে। এর বড় number রাখতে চাইলে ভুল answer দেখাবে। তুমি টেস্ট করার জন্য variable `int a,b,c;` declar করো তারমাঝে `a=32768` এবং `b=32768` রেখে `c=a+b` যোগ করেদেখো answer ভুল আসবে 65536 আসবেনা। এখন যদি 32768 এর থেকে বরো নাম্বার রাখতে হয় তবে কি করবে? যদি তুমি জানো যে user যে নাম্বার input দেবে তা negative হবেনা তাহলে তুমি unsigned ব্যবহার করতে পারো। তাহলে তুমি $2^{16}=65536$ বরো নাম্বার রাখতে পারবে। কিন্তু যদি 65536 এর থেকে বড় নাম্বার নিয়ে কাজ করতেহয় তবে কি করবে? এই ক্ষেত্রে তুমি long int ব্যবহার করতে পারো যা মেমরীতে 4 byte জায়গা নেবে। তারমানে তুমি $2^{31}=2147483648$ বড় নাম্বার রাখতে পারবে। একই ভাবে যদি কোনো নাম্বার float এ রাখা সম্ভব না হয় তবে তুমি double use করতে পারো।

এইসব Data type এর ক্ষেত্রে input নেওয়া ও output দেখানোর জন্যে নিচের টাবেল দেখো:-

Variable unsigned integer type হলে	%u ব্যবহার করতেহবে
Variable long int type হলে	%ld ব্যবহার করতেহবে
Variable double type হলে	%lf ব্যবহার করতেহবে

Value define করা:

যদি এমন হয় কোনো variable বারবার লাগছে কিন্তু তার value সবসময় same তবে তার মান(value) আগে define করাযায়। variable value করে নিচের মতো করে:-

```
#define variable name value
```

বুঝলে নাতো? ধরো তুম একটা বৃত্তের ক্ষেত্রফল বেরকরবে। তাহলে তমার পাই এর মান লাগবে। তুমি চাইলে তা আগেই নির্ধারণ করে দিতে পারো। #define কে main() এর বাহিরে প্রগ্রামের সূরুতে লিখতে হয়। নিচের কোড দেখো:-

```
#include<stdio.h>
#define pi 3.1416
int main() {
    float r,area;
    scanf( "%f",&r) ;
    area=pi*r*r;
    printf("The area of the circle wjich radius %f is %f",r,area);
}
```

Punction and error: আমরা ব্যাকরণ এর ক্ষেত্রে যেমন নিয়ম মেনে চলি প্রগ্রামিং এ ও নিয়ম মেনে চলতেহয়। এই নিয়ম মেনে চলাই হলো Punction | আমরা কোডে বিভিন্ন কমা, সেমিকোলন, "" লিখে থাকি। নিয়ম মেনে এইগুলো লিখায় Punction | যদি এর কোনোনা মিস হয় তবে তাকে বলে error। কোডে error পেলে কোড রান করবেনা। নিচেরমতো message দিবে এবং message এ বলেদিবে কতো নং লাইনে ভুল। figure1 দেখ:-

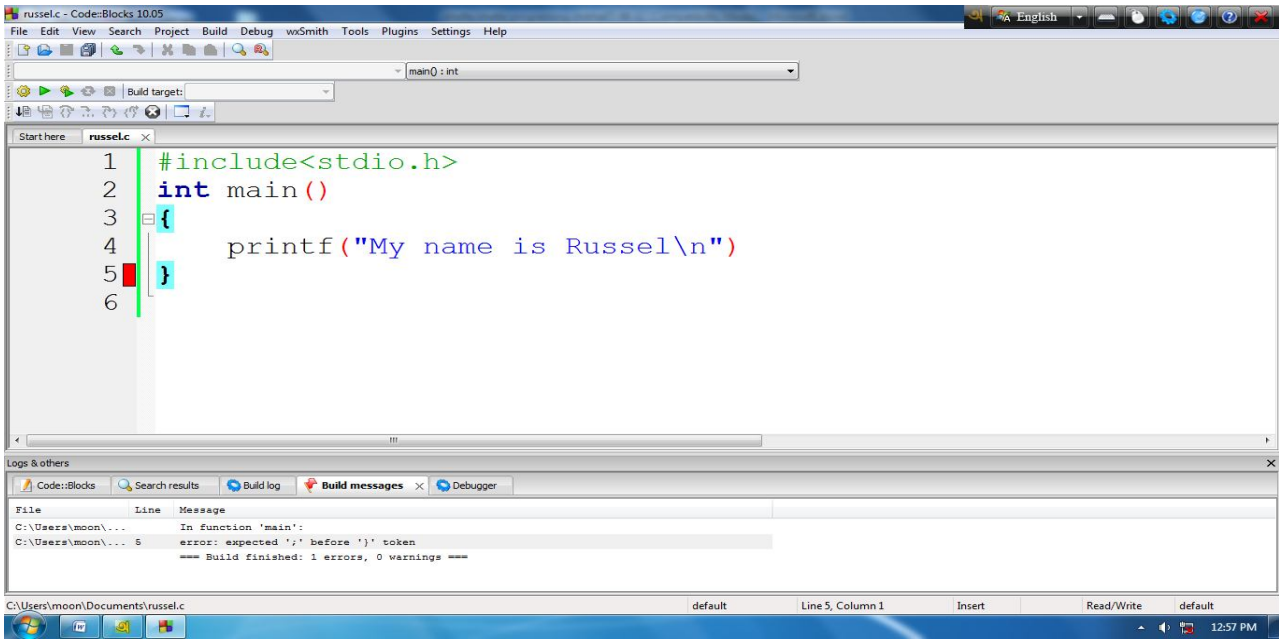


figure1: সেমিকোলন(;) না দেওয়াই error message

Backslash character constants: computer কি বোর্ডে বিভিন্ন ক্যারেকটার(a, R,@,\) আছে যেগুলো প্রগ্রামিং এ সরাসরি ব্যবহার করতে পারিনা। এইধরনের ক্যারেকটারকে বলে Backslash character constants। এইগুলো লিখার নিয়ম নিচেদেওয়া হলো:-

নাম	মিনিং
\a	Audible alert
\b	Back space
\f	Form feed
\n	New line
\r	Carring return
\t	Horizontal tab
\v	Vertical tab
\o	NULL
\'	Single quote
\"	Double quote
\?	Question mark
\\	backslash

Formate spacificator: আমরা printf() ফাংশনের মধ্যমে user কে আউটপুট দেখাতে পারি কিন্তু Formate spacificator এর মধ্যমে আউটপুট কে আর সন্দূর ভাবে দেখান যায়। যেমন left alignment, right alignment, দশমিক এর পর কতোঘর দেখাতে চাও।
 right alignment এর জন্যে printf(“%nd”,variable name); এখানে n বলতে কতঘর right alignment করতেচাও।
 উদাহরণ: নিচের কোড দেখো:-

```
#include<stdio.h>
int main()
{
    int a=6,b=66,c=666,d=6666;
    printf( "%5d \n%5d\n %5d \n %5d\n",a,b,c,d) ;
}
```

Output:

```
6
66
666
6666
```

left alignment এর জন্যে printf("%-nd",variable name) ; এখানে -n বলতে কতঘর left alignment করতেচাও।
উদাহরণ: নিচের কোড দেখো:-

```
#include<stdio.h>
int main()
{
    int a=6,b=66,c=666,d=6666;
    printf( "%5d \n%5d\n %5d \n %5d\n",a,b,c,d) ;
}
```

Output:

```
6
66
666
6666
```

printf("%.nf") ; এখানে n বলতে দশমিক এর পর কতঘর দেখাতে চাও।

```
#include<stdio.h>
int main()
{
    float a=6.123456;
    printf( "%f \n",a) ;
    printf( "%.2f \n",a) ;
}
```

Output:

```
6.123456
6.12
```

অনুশীলনি:

১। user এর কাছ থেকে বর্গের এক বাহুর দৈর্ঘ্য input হিসাবে নেবে তারপর বর্গের ক্ষেত্রফল output হিসাবে দেখাবে নিচের মত করে।

The area of the square is X এখানে X মানে বর্গের ক্ষেত্রফল।

২। user এর কাছ থেকে আয়তক্ষেত্রের দৈর্ঘ্য ও প্রস্থ input হিসাবে নেবে তারপর আয়তক্ষেত্রের ক্ষেত্রফল output হিসাবে দেখাবে নিচের মত করে।

The area of the rectangle is X [এখানে X মানে আয়তক্ষেত্রের ক্ষেত্রফল।]

৩। user এর কাছ থেকে একটা integer number input হিসাবে নেবে তারপর output হিসাবে দেখাবে নিচের মত করে।

The square of X is Y

The cube of X is P

এখানে X user এর দেওয়া input Y user এর দেওয়া input এর বর্গ এবং P user এর দেওয়া input এর ঘনো।

৪। user এর কাছ থেকে ত্রিভুজের ভূমি ও উচ্চতা input হিসাবে নেবে তারপর ত্রিভুজের ক্ষেত্রফল output হিসাবে দেখাবে নিচের মত করে।

The area of the triangle is X [এখানে X মানে ত্রিভুজের ক্ষেত্রফল।]

ষষ্ঠ অধ্যায়

Relational operator logical operator and condition(if else)

রনি তোমার নিজের কথা ভাবো। ঈদ এলে বা মাঝে মাঝে school এ যাবার সময় বায়না ধরোনা? মাঝে মাঝে বন্ধুদের বলোনি অনেক শর্তের কথা? আমি মনে করে দেই। ঈদ এলে তোমার বাবকে কাখনো বলনি যে যদি নতুন পানজাবী দাও তবে নামাযে যাবো না হলে যাবোনা। আজ school এ যাবার সময় ১০০ টাকা দিলে school এ যাবো, না দিলে যাবোনা। যদি তুই আমার বাসায় না আসিস তবে আমি তোমার বাসায় যাবনা। তুমি যেমন শর্ত দাও, শর্ত মানো। কম্পিউটার তেমন শর্ত মানে। আমরা কম্পিউটার এর শর্ত শেখার আগে কতগুলো operator সম্পর্কে জানবো।

Relational operator: নিচের টাবিল এ Relational operator দেওয়া হল:-

operator এর নাম	কাজ	পড়ার নিয়ম
<	ছোটো	Less than
<=	ছোটো অথবা সমান	Less than or equal
>	বড়ো	Greater than
>=	বড়ো অথবা সমান	Greater than or equal
==	সমান সমান	Equal to
!=	সমান নয়	Not equal

$a < b$ a থেকে b ছোটো হলে সত্য হবে

$a <= b$ a থেকে b ছোটো বা a এবং b সমান হলে সত্য হবে

$a > b$ a থেকে b বড়ো হলে সত্য হবে

$a >= b$ a থেকে b বড়ো বা a এবং b সমান হলে সত্য হবে

$a == b$ a এবং b সমান হলে সত্য হবে

$a != b$ a এবং b সমান নাহলে সত্য হবে

logical operator: logical operator দুইটি AND (&&) ও OR(||)

যদি দুইটা variable এর মাঝে logical operator থাকে তবে নিচের মত কাজ করে:-

AND operator: condition1 && condition2

সত্য হবে যদি condition1 এবং condition2 দুইটি condition ই সত্য হয়। অনেকটা এইরকম **যদি এবং যদি** তাহলে সত্য।

OR operator: condition1 || condition2

সত্য হবে যদি condition1 অথবা condition2 এর যে কোনো একটি condition সত্য হয়। অনেকটা এইরকম **যদি অথবা যদি** তাহলে সত্য।

if এবং else এর ধারণা: if মানে যদি আর else মানে যদি না। if else লিখার নিয়ম নিচে দেওয়া হল:-

```
if(condition) { Statement; }
```

```
else { Statement; }
```

যদি if(condition) সত্য হয় তবে if এর Statement কাজকরবে আর যদি if(condition) সত্য নাহয় তবে else এর Statement কাজকরবে।

if এবং else এর ধারণা আরো ক্লিয়ার হওয়ার জন্যে একটা প্রব্লেম সল্ভ করি তাহলে ক্লিয়ার হবে। if এবং else নিয়ে প্রগ্রাম লিখতে গেলে যে কোনো একটা শর্ত ধরে নিতে হয়। একটা উদাহরণ দেয়। ধরো আমি হঠাৎ অন্ধ হয়ে গেলাম। আমার সামনে তুমি এসে বললে রাসেল আমি তোমাকে দুইটা জিনিস দিবো একটা খাতা আর একটা কলম। তোমাকে বলতে হবে কোনটা কলম? এখন আমি কি করবো? আমি খাতা আর কলম এর বৈশিষ্ট্য জানি। তাই আমি এক হাতে খাতা আর অন্য হাতে কলম নিলাম। ধরে নিলাম আমার বাম হাতে কলম আছে। এইবার কলমের যে বৈশিষ্ট্য আমি জানি তা চেক করবো। যদি মিলে যায় তারমানে আমার বাম হাতে যা আছে তা কলম। আর যদি না মিলে তবে আমাকে কি চেক করতে হবে ডান হাতে কি আছে? আমি চেক না করেই বলেদিলে পারি আমার দান হাতে কলম আছে।

এখন প্রব্লেম এ আসি। আমি user এর কাছ থেকে input হিসাবে দুইটা নাম্বার নিবো। আমাকে বলতে হবে কোন নাম্বার বড়? ধরো আমি যে input নিবো তার একটা a এর মাঝে আর একটা b এর মাঝে রাখব। তারপর ধরে নিবো a value বড়। যদি বড় নাহয় তবে আমি অনাআয়েশে বলতেপারি b এর মাঝে যে value আছে তা বড়। এইবার নিচের কোড দেখ:-

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

```
int main()
```

```
{
```

```
    int a,b;
```

```
    scanf("%d %d",&a,&b);
```

```
    if(a>b) printf("%d is greater than %d",a,b);
```

```
    else printf("%d is greater than %d",b,a);
```

```
}
```

ধরো user input হিসাবে 15 এবং 20 দিলো। তাহলে কি হবে? a=15 এবং b=20 হবে। তারপর if(a>b) চেক করবে। তাই if(15>20) সত্য নয়। তাহলে else এর statement printf("%d is greater",b); প্রিন্ট করবে। তাহলে আউটপুট হবে 20 is greater 15। আশাকরি এই একটা problem এই if else মোটামোটি ক্লিয়ার হওয়ার কথা।

এখন বলো একটা নাম্বার কখন জোড় আর কখন বিজোড় হয়? পারলেনা তো? যে কোনো নাম্বার কে 2 দ্বারা ভাগ করলে যদি ভাগশেষ 0 হয় তবে তা জোড় আর যদি 0 নাহয় তবে তা বিজোড় হবে। এখন বলো যদি হিসাবে একটা নাম্বার দেয় তবে তা জোড় না বিজোড় তার জন্যে প্রগ্রাম লিখতে পারবে? জানি পারবে তারপরে ও আমি লিখে দিলাম।

```
#include<stdio.h>
int main()
{
    int a;
    scanf("%d",&a);
    if(a%2==0) printf("%d is even",a);
    else printf("%d is odd",b);
}
```

এই কোড এ আমি ধরে নিছি a জোড় তাই a কে ২ দ্বারা ভাগ করে দেখলাম যদি ভাগশেষ ০ হয় তবে কোড আর যদি ভাগশেষ ০ না হয় তবে বিজোড়।

if, else if এবং else এর ধারণা:

if, else if এবং else লিখার সিস্টেম:

```
if(condition) {statement}
else if(condition) {statement}
else if(condition) {statement}
else {statement}
```

এইবার একটা প্রব্লেম সল্ভ করবো যাতে লজিকাল অপারেটর ব্যবহার করবো এবং if, else if ও else এর ধারণা ক্লিয়ার করবো। ধরো user input হিসাবে ৩টা নাম্বার দিবে আমাকে বলতে হবে কোনটা সবচেয়ে বড়? তুমি বলো তিনটা নাম্বার এর মাঝে একটা বড় হতে গেলে ওই নাম্বার কে অশ্যই অপর দুই নাম্বার থেকে বড় হতে হবে? ধরো user যে দিবে তা আমি তিনটা variable a b c এর মাঝে রাখবো। এখন তুমি বলো যদি a বড় হতে হয় তবে a কে অশ্যই b এবং c থেকে বড় হতে হবে? যদি b বড় হতে হয় তবে b কে অশ্যই a এবং c থেকে বড় হতে হবে? যদি c বড় হতে হয় তবে c কে অশ্যই a এবং b থেকে বড় হতে হবে? তুমি কি এখন কোড লিখতে পারবে? আচ্ছা আমি লিখছি।

```
#include<stdio.h>
int main()
{
    int a,b,c;
    scanf("%d %d %d",&a,&b,&c);
    if((a>b)&&(a>c)) printf("%d is greater",a);
    else if((b>a)&&(b>c)) printf("%d is greater",b);
    else printf("%d is greater",c);
}
```

nisted if else: যদি একটা if else এর ভিতর আর একটা if else থাকে তাখন তাকে nisted if else বলে। nisted if else লিখার নিয়ম:

```
if(condition)
{
    if(condition) {statement}
    else {statement}
}
```

```

else if(condition)
{
    if(condition) {statement}
    else {statement}
}

```

```

else
{
    if(condition) {statement}
    else {statement}
}

```

nisted if else এর ক্ষেত্রে বাহিরের If(condition) সত্য হলে ভিতরে যাবে নাহলে পরের else if condition এ যাবে। পরেরটা সত্য হলে ভিতরে যাবে নাহলে পরের else if condition এ যাবে। কোনো condition সত্য নাহলে else এর ভিতর যাবে। ধারণা ক্রিয়ার করার জন্যে একটা প্রব্লেম সলভ করা যেতে পারে। মনেকরো user input হিসাবে একটা নাম্বার দিবে তোমাকে বলতে হবে ওই নাম্বার 100 থেকে 1000 এর ভিতর কিনা এবং ঐ নাম্বার জোড় না বিজোড়? output হবে :

the number is between 100 and 1000 and the number is even অথবা
the number is between 100 and 1000 and the number is odd অথবা
the number is not between 100 and 1000 and the number is even অথবা
the number is not between 100 and 1000 and the number is odd
কি কোড লিখতে পারবে? জানি তুমি পারবে তারপরে ও আমি লিখে দিলাম।

```

#include<stdio.h>
int main()
{
    int a;
    scanf("%d",&a);
    if(a>=100)&&(a<=1000)
    {
        if(a%2==0) printf("the number is between 100 and 1000 and the number is even");
        else printf("the number is between 100 and 1000 and the number is odd");
    }
    else
    {
        if(a%2==0) printf("the number is not between 100 and 1000 and the number is even");
        else printf("the number is not between 100 and 1000 and the number is odd");
    }
}

```

অনুশীলনি:

১। user input হিসাবে একটা নাম্বার দিবে। তোমাকে বলতে হবে নাম্বার positive না negative |

উদাহরণঃ user input হিসাবে 15 দিলে দেখাবে 15 is positive

user input হিসাবে -445 দিলে দেখাবে -445 is negative

২। user input হিসাবে দুইটা নাম্বার দিবে। তোমাকে বলতে হবে কোন নাম্বার ছোট?

উদাহরণঃ user input হিসাবে 15 এবং 25 দিলে দেখাবে 15 is smaller than 25

user input হিসাবে 45 এবং 1 দিলে দেখাবে 1 is smaller than 45

৩। user input হিসাবে দুইটা নাম্বার দিবে। তোমাকে বলতে হবে প্রথম নাম্বার কে দ্বিতীয় নাম্বার দ্বারা ভাগ করা যায় কি না?

উদাহরণঃ user input হিসাবে 15 এবং 25 দিলে দেখাবে 15 is not divisible by 25

user input হিসাবে 45 এবং 3 দিলে দেখাবে 45 is divisible by 3

৪। user input হিসাবে তিনটা নাম্বার দিবে তোমাকে best two এর average বের করতে হবে।

উদাহরণঃ

user input হিসাবে 15 25 এবং 35 দিলে দেখাবে the average of 25 and 35 is 30

user input হিসাবে 45 5 এবং 3 দিলে দেখাবে the average of 45 and 5 is 20

৫। user input হিসাবে একটা ছাত্রের প্রাপ্ত নাম্বার দিবে। তোমাকে বলতে হবে ওই ছাত্র কোন গ্রেড পেলো।

0-32= F 33-39= D 40-49= C 50-59= B 60-69= A⁻ 70-79= A 80-100= A⁺

৬। এক ভারসিটি টিচার ৩টা class test (20) mid term exam(30) attendance (10) final (40) হিসাবে exam নিলো। তোমাকে class test এর best two এর average নিতে হবে | তারপর প্রাপ্ত গ্রেড বলতে হবে।

0-32= F 33-39= D 40-49= C 50-59= B 60-69= A⁻ 70-79= A 80-100= A⁺

user input হিসাবে প্রথমে class test তারপর midterm তারপর attendance তারপর final marks দিবে।

উদাহরণঃ Input 15 16 18 28 10 36 output A⁺ Input 10 12 14 20 8 30 output A

৭। user এর কাছ থেকে input হিসাবে ৩টা integer number নিবে। তুমি একটা প্রগ্রাম লিখবে যা বলবে এই ৩টা নাম্বার দিয়ে ত্রিভুজ বানানো যাবে কিনা? যদি যায় তবে প্রিন্ট করো yes না হলে প্রিন্ট করো no।

উদাহরণঃ

১। user input হিসাবে ৩টা integer number দিলো 4 3 5 output হবে yes

2। user input হিসাবে ৩টা integer number দিলো 1 2 3 output হবে no

Loop কি?

কোন এক স্থান থেকে যাগা শুরু করে আবার সেই স্থানে ফিরে আসাকে Loop বলে। মনেকর তুমি ক্রিকেট মাঠের বাউণ্ডারির এক স্থান থেকে দৌড় শুরু করে আবার আগের জায়গায় ফিরে আসলে। তাহলে তুমি যে পথ অতিক্রম করলে তাই প্রকৃতপক্ষে Loop।

Loop কেন ব্যবহার করবো?

প্রোগ্রামে কনো এক বা একাধিক নির্দিষ্ট বার কাজ করানোর জন্যে Loop ব্যবহার করা হয়। Loop সম্পর্কে জানার আগে লক্ষ করো। ধরো আমি variable declare করলাম int a; লিখলাম এবং a=5। তারমানে a এর value 5। এখন যদি নিচের মতো করে লিখি তাহলে কি হবে?

1. a=a+1;
2. a=a+1;
3. a=a+1;
4. printf("%d",a);

১নং লাইনে a+1 তারমানে 5+1 এবং যা a এর ভিতর রাখবে। তারমানে এখন ১নং লাইন শেষে a=5+1 বা a=6। তারমানে ২নং লাইনে a এর মান 6। ২নং লাইনে আবার a=a+1 তারমানে a=6+1 বা a=7। একই ভাবে ৩নং লাইনে আবার a=a+1 তারমানে a=7+1 বা a=8। তাহলে ৪নং লাইনে a এর মান প্রিন্ট করবে 8। একই ভাবে a=a+2 লিখলে a মান 2 বাড়বে। a=a+5 লিখলে a মান 5 বাড়বে। একে বলে increment। আবার যদি লিখি a=a-1 তাহলে a এর 1 মান কমবে। যদি লিখি a=a-5 তাহলে a এর মান 5 কমবে। একে বলে decrement। Loop কেন ব্যবহার করবো তা জানার জন্যে নিচের কোড লক্ষ করো:- যেখানে আমি চাইতেছি একটা প্রোগ্রাম লিখব যা ১-৫ পর্যন্ত প্রিন্ট করবে। তাহলে কি কোড নিচের মতাকরে লিখবে?

<pre>#include<stdio.h> int main() { int a=1; printf("%d ",a); a=a+1; printf("%d ",a); a=a+1; printf("%d ",a); a=a+1; printf("%d ",a); a=a+1; printf("%d ",a); }</pre>	<p>এই কোড এর আউটপুট হবে 1 2 3 4 5। কারন প্রথমে প্রিন্ট করবে a=1। তারপর a=a+1 তারমানে a=1+1 বা a=2 তাই প্রিন্ট করবে 2। তারপর আবার a=a+1 তারমানে a=2+1 বা a=3 তাই প্রিন্ট করবে 3। তারপর আবার a=a+1 তারমানে a=3+1 বা a=4 তাই প্রিন্ট করবে 4। তারপর আবার a=a+1 তারমানে a=4+1 বা a=5 তাই প্রিন্ট করবে 5। তারমানে কি? কোড ঠিক আছে? হ্যাঁ কোড ঠিক আছে কিন্তু ভাবোতো তমাকে প্রিন্ট করতেহবে 1-1000। তারমানে কি? তুমি 1000 বার a=a+1 লিখবে? 1000 বার প্রিন্ট করবে? তাহলে কোড কতোলাইন হোবে? ব্যাপার কি সহজ হবে? এখন নিশ্চয় তমার মাথায় কাজকরছে যদি এমন হতো আমি একবার প্রিন্ট করবো তাহলে 1-1000 পর্যন্ত প্রিন্ট করবে তাহলে কতো ভাল হতো তাই না? হ্যাঁ কনো statement একবার লিখে যতবার ইচ্ছা কাজ করানর জন্যে loop ব্যবহার করা হয়। loop মেইনলি for loop, while loop, do while loop।</p>
---	---

for loop: for loop কি তা জানার আগে তোমার প্রিয় কুকুর টম এর কথা ভাবো। সে তমার প্রতিতিকথা পালঙ্করে তাইনা? মনেকরো তুমি বৃত্তাকার ভাবে কতগুল বল রাখছো। এইবার টম কে বললে বল গুল আনার জন্যে। এবং এই একই কাজ তাকে ৫বার করতে বললে। টম যথেষ্ট বুদ্ধিমান কিন্তু অলস। সে কাজ শুরু করবার আগে ধরে নিবে আমি কাজ করতে যাইতেছি। এইটাই কি শেষ? যেহেতু টম অনেক অলস তাই সে দেখবে শেষ হলেই ঘুমাবে। একবার কাজ করবে সাথেসাথে গুণবে কতো বার কাজ করল? একবার কাজকরেই আবার চেক করবে কাজ শেষ হলো কিনা। এইবার for loop এ আসা যাক। for loop ৪টা অংশ নিয়ে কাজকরে। for loop লিখতে হয় নিচেরমতো:-

for(initial condition; ending condition; increment/ decrement)

```
{
    Statement;
}
```

এখানে initial condition বলতে এইখান থেকে কাজ শুরু। ending condition বলতে বুঝায় এই condition মিথ্যা হলে আর for loop এর Statement এ জাবেনা। increment বলতে মান বৃদ্ধি decrement বলতে মান কমানো। Statement বলতে যে কাজ করতে চাও।

এবার একটা প্রোগ্রাম লিখব যা for loop ক্রিয়ার করবে: ধরো একটা প্রোগ্রাম লিখবো যা ১-১০ পর্যন্ত প্রিন্ট করবে।

<pre>#include<stdio.h> int main() { int i; for(i=1;i<=10;i=i+1) { printf("%d",i) ; } }</pre>		<p>এবার তোমার টমের কথা আর একবার ভাবো কোনো এক মার্চে একটা ব্ল্যাক বোর্ড রেখে তাকে একটা চক দিয়ে বললে মার্চে একবার ঘুরে আসবে আর যতবার ঘুরবে তত নং লিখবে। যদি ১০ বার হয়ে যায় তবে থেমেযাবে। টম শুরু করবে প্রথমে ধরেনিবে সে ১ নং বার কাজকরতে যাইতেছে এবং সে চেক করবে ১০ বার হলো কিনা। যদি ১০এর কম বা সমান হয় তবে থেমে লিখবে আর ১০ এর বেশি হলে থেমে যাবে। এইবার নিচের টেবিল দেখো:-</p>		
i=1	i<=10	i<=10 True/ false	If true print i	i=i+1
1=1	1<=10	True	1	i=1+1
i=2	2<=10	True	2	i=2+1
i=3	3<=10	True	3	i=3+1
i=4	4<=10	True	4	i=4+1
i=5	5<=10	True	5	i=5+1
i=6	6<=10	True	6	i=6+1
i=7	7<=10	True	7	i=7+1
i=8	8<=10	True	8	i=8+1
i=9	9<=10	True	9	i=9+1
i=10	10<=10	True	10	i=10+1
i=11	11<=10	False		

উপরের কোডে প্রথমে for এর ভিতর i=1 এইটা initial condition। তারমানে i এর মান 1। এইবার চেক করবে 1<=10 অর্থাৎ 1 থেকে 10 ছোটো বা সমান কিনা? যেহেতু 10 থেকে 1 ছোটো তারমানে condition সত্য তাই for এর Statement এ যাবে অর্থাৎ i এর মান হিসাবে 1 প্রিন্ট করবে। তারপর i=i+1 তারমানে i=1+1 বা i=2। একিভাবে চেক করবে 2<=10 অর্থাৎ 2 থেকে 10 ছোটো বা সমান কিনা? যেহেতু 10 থেকে 2 ছোটো তারমানে condition সত্য তাই for এর Statement এ যাবে অর্থাৎ i এর মান হিসাবে 2 প্রিন্ট করবে। এইভাবে কাজ করতেই থাকবে। এক সময় i=11 হবে তখন 11<=10 condition মিথ্যা হবে কারণ 10 থেকে 11 ছোটো বা সমান নয় তাই condition মিথ্যা হবে এবং for loop এর কাজ শেষ হবে। উপরের for loop printf("%d",i) এ i print করছি তাই অউটপুট 1 2 3 4 5 6 7 8 9 10 হবে। যদি আমি i print না করে অন্য কিছু লিখতাম তবে তা ১০ বার প্রিন্ট করবে কারণ loop ১০ বার ঘুরবে তাই। আরো ক্রিয়ার হয়ার জন্যে নিচের কোড দেখো যে কোড রান করলে ১০ বার প্রিন্ট করবে * এবং অপর কোড রান করলে ১০ বার প্রিন্ট

<pre>#include<stdio.h> int main() { int i; for(i=1;i<=10;i=i+1) { printf(" * "); } }</pre>	<pre>#include<stdio.h> int main() { int i; for(i=1;i<=10;i=i+1) { printf("Russel "); } }</pre>	<p>১ম কোড রান করলে লুপ ১০ বার ঘুরবে আর আউটপুট হিসাবে প্রিন্ট করবে * * * * * * * * * *। কারণ লুপ ঘুরবে ১০ বার উপরের টেবল দেখো। একই ভাবে ২য় কোড রান করলে আউটপুট হিসাবে প্রিন্ট করবে Russel Russel Russel Russel Russel Russel Russel Russel Russel Russel। আমি \n না দেওয়ায় পাশাপাশি প্রিন্ট করছে। যদি \n দিতাম তবে নিচে নিচে প্রিন্ট করতো।</p>
---	---	--

for loop আরো ক্লিয়ার করার জন্যে আর একটা প্রোগ্রাম লিখব যা 1-10 পর্যন্ত সব নাম্বার এর যোগফল দেখাবে। তারমানে 1+2+3+4+5+6+7+8+9+10=55। তারমানে আউটপুট হবে 55।

<pre>#include<stdio.h> int main() { int i,sum=0; for(i=1;i<=10;i=i+1) { sum=sum+i; } printf("%d",sum); }</pre>	<p>কোডটি লক্ষ্য করো এখানে প্রথমে ধরেনিলাম sum=0। তারপর একটা for loop ঘুরায়লাম 1-10 পর্যন্ত। এবং i কে increment করলাম 1 করে এবং তা পূর্বের sum এর সাথে যোগ করলাম। প্রথমে i=1 এবং 1<=10 সত্য তাই sum=sum+i বা sum=0+1 বা sum=1। তারপর i=i+1 তারমানে i=1+1 বা i=2 হবে। 2<=10 সত্য হবে আবার sum=sum+i বা sum=1+2 বা sum=3 হবে। আবার i=i+1 তারমানে i=2+1 বা i=3 হবে। 3<=10 সত্য হবে আবার sum=sum+i বা sum=3+3 বা sum=6 হবে। এভাবে i=10 পর্যন্ত সত্য হবে এবং sum=sum+i করতে থাকবে। যখন i=11 হবে তখন 11<=10 মিথ্যা হবে এবং লুপ এর বাহিরে আসবে। সেখানে printf("%d",sum); আছে যা sum এর ভেলু প্রিন্ট করবে। উপরের টেবল এর মতো বানাতে নিজে চেষ্টা করো।</p>
---	--

এতক্ষণ আমরা শুধু আমাদের প্রয়োজন অনুযায়ী লুপ ঘুরায়লাম। অর্থাৎ আমরা ১০ পর্যন্ত লুপ ঘুরায়লাম ১-১০ পর্যন্ত প্রিন্ট করলাম ১-১০ পর্যন্ত যোগ করলাম। কিন্তু যদি ইউজার ইনপুট দেয় এবং তার নাম্বার পর্যন্ত কাজ করতে হয় তবে কি করবো? ধরো ইউজার একটা নাম্বার দিল তোমাকে বলতে হবে এই নাম্বার এর মাঝে কতগুলো বিয়োড় নাম্বার আছে? তাহলে কি করবে? ইউজার এর কাছ থেকে নাম্বার নিতে হবে ইতারপর এই নাম্বার পর্যন্ত লুপ ঘুরাতে হবে এবং প্রতিটি নাম্বার চেক করতে হবে তবে কাউন্ট করতে হবে। বুঝলেনাতো? নিচের কোড দেখো:-

<pre>#include<stdio.h> int main() { int i,count=0,n; scanf("%d",&n); for(i=1;i<=n;i=i+1) { if(i%2!=0) count=count+1; } printf("%d", count); }</pre>	<p>এখানে প্রথমে আমি count=0 ধরছি কারণ প্রথমে কন বিয়োর নাম্বার নাই। তারপর i=1 ধরলাম এবং চেক করলাম i<=n হয় তবে লুপের ভিতর ঢুকবে। লুপের ভিতর i কে চেক করবে যদি i কে 2 দ্বারা ভাগকরা না যায় অর্থাৎ ভাগফল ০ হয় তবে count এর মান 1 বাড়াবে। তারপর i=i+1 তারমানে i এর মান 1 বাড়াবে এবং i<=n চেক করবে। যদি সত্য হয় তবে লুপ এর ভিতর যাবে এবং আবার ৩ i কে চেক করবে বিয়োড় কিনা? যদি বিয়োড় হয় তবে count এর মান 1 বাড়াবে। এভাবে কাজ করতেই থাকবে। যখন i<=n মিথ্যা হবে তখন লুপ এর বাহিরে যাবে এবং printf("%d",count); এর কাজ করবে অর্থাৎ count এর মান প্রিন্ট করবে। লুপ বোঝার জন্যে নিচের টেবল দেখো:- n=4 output=2 কারণ 1-4 এর ভিতর 2টা বিয়োড় নাম্বার আছে।</p>
--	--

i=1	i<=n	True or false	check is i odd?	If odd count+1	i=i+1
i=1	1<=4	True	1%2=1	Count=1	i=1+1
i=2	2<=4	True	2%2=0	Count=1	i=2+1
i=3	3<=4	True	3%2=1	Count=2	i=i+1
i=4	4<=4	True	4%2=0	Count=2	i=4+1
i=5	5<=4	False			

লুপ এর পর printf("%d", count); যেখানে count এর শেষ মান ২। তাই আউটপুট হবে ২।

আর একটি প্রোগ্রাম লিখব যেখানে ইউজার দুইটা নাম্বার যোগ করবে কিন্তু কতবার যোগ করবে তা আগেই বলে দিবে। অর্থাৎ ইউজার a এবং b যোগ করবে কিন্তু সে n বার যোগ করবে।

<pre>#include<stdio.h> int main() { int i,n; scanf("%d",&n); for(i=1;i<=n;i=i+1) { int a,b,sum; scanf("%d %d",&a,&b); sum=a+b; printf("Case : %d %d",i, sum); } }</pre>	Samole input	Sample output
	4	
	10 12	Case 1: 22
	45 20	Case 2: 65
	102 204	Case 3: 306
	512 621	Case 4: 1133
<p>উপরের কোডে প্রথমে ইউজার এর কাছ থেকে একটা নাম্বার ইনপুট হিসাবে নিবে যা n এর মাঝে রাখবে। তারপর for loop এর কাজ শুরু করবে। লুপ n পর্যন্ত ঘুরবে। যেহেতু n=4 তাই লুপ 4 বার ঘুরবে এবং প্রতিবার ইউজার এর কাছ থেকে দুইটা নাম্বার ইনপুট নিবে যা a এবং b এর মাঝে রাখবে। তারপর a এবং b এর মান যোগ করে sum এর ভিতর রাখবে। তারপর printf("Case : %d %d",i, sum) প্রিন্ট করবে। তারমানে আউটপুট হবে Sample output এর মতো। এখানে Case বলতে কত বার কাজ করল তাই বুঝান হয়েছে।</p>		

আশা করি for loop মোটামোটি ক্রিয়ার তাই নিচের অনুশীলনি নিজে সল্ভ করো:-

অনুশীলনি

১। ইউজার তোমাকে n গুলো নাম্বার দিবে তোমাকে n গুলো নাম্বারের গুণফল দেখাতে হবে।	Sample input	sample output
	3 2 4 5	The multiplication of all number is 40
২। ইউজার তোমাকে n গুলো নাম্বার দিবে তোমাকে n গুলো নাম্বারের গড়(average) দেখাতে হবে।	4 2 4 8 6	Average=5
৩। ইউজার তোমাকে n গুলো নাম্বার দিবে তোমাকে বলতে হবে n গুলো নাম্বারের মধ্যে সবচেয়ে বড় নাম্বার কোনটা	4 2 4 12 8	Maximum=12
৪। ইউজার তোমাকে n গুলো নাম্বার দিবে তোমাকে বলতে হবে n গুলো নাম্বারের মধ্যে সবচেয়ে ছোট নাম্বার কোনটা?	4 2 4 12 8	Minimum=2
৪। ইউজার তোমাকে n গুলো নাম্বার দিবে তোমাকে বলতে হবে n গুলো নাম্বারের মধ্যে বিয়োড় নাম্বার কয়টা?	7 1 2 4 5 6 7 3	Total odd number=4
৫। ইউজার এর কাছ থেকে ১টা নাম্বার নিতে হবে তার factorial কত	5	The factorial is 120
৬। ইউজার এর কাছ থেকে input ২টা নাম্বার a এবং b নিতে হবে a^b কত বলতে হবে?	2 3	8
	3 4	81
৭। ইউজার এর কাছ থেকে input হিসাবে একটা নাম্বার n নিবে। n থেকে 1 পর্যন্ত প্রিন্ট করতে হবে।	5	5 4 3 2 1
	6	6 5 4 3 2 1
৮। ইউজার এর কাছ থেকে input হিসাবে একটা নাম্বার n নিবে। বলতে হবে 1 থেকে n পর্যন্ত কতগুলো নাম্বার আছে যা 3 দ্বারা ভাগ করা যায়।	12	4
	30	10
৮। ইউজার এর কাছ থেকে input হিসাবে একটা নাম্বার n নিবে। বলতে হবে 1 থেকে n পর্যন্ত কতগুলো নাম্বার আছে যা 3 ও 4 দ্বারা ভাগ করা যায়।	15	1
	30	2
৯। ইউজার এর কাছ থেকে input হিসাবে একটা নাম্বার n নিবে। বলতে হবে 1 থেকে n পর্যন্ত কতগুলো নাম্বার আছে যা 3 অথবা 4 দ্বারা ভাগ করা যায়।	15	6
	30	12
১০। ইউজার এর কাছ থেকে input ২টা নাম্বার a এবং b নিতে হবে এবং বলতে হবে 1 থেকে a পর্যন্ত কতগুলো নাম্বার আছে যা b দ্বারা ভাগ করা যায়।	15 3	4
	10 2	4

4.2 আচ্ছা রনি বলতো আমি যদি for loop কে for(; ;) লিখা যাবে? কি অবাক হলে? কোন Initial condition, ending condition, increment/decrement কিছু নাই তাহলে এইটা কি কাজ করবে? হাঁ কাজ করবে যদি Initial condition for loop এর আগে (বা মাঝে মাঝে for loop এর ভিতর ও থাকতে পারে) থাকে for loop এর ভিতর increment/decrement থাকে এবং ending condition থাকে। কি বুঝলে না? নিচের কোড দেখ:-

#include<stdio.h> int main() { int i=1; for(; ;) { printf("%d",i) ; if(i==10) break; i=i+1; } }	এই Loop এর বাহিরে i=1 তারপর for loop। loop এর ভিতর printf("%d",i) তারমানে i প্রিন্ট করবে। তারপর চেক করবে if(i==10) break যদি i এর মান 10 হয় তবে loop থেকে বের হয়ে যাবে। এখানে break একটা কি ওয়ার্ড এর কাজ যে কোন loop কে থেমে দেওয়া। ##যে কোন loop তার Initial condition থেকে শুরু হয় তারপর ending condition চেক করে সত্য হলে loop এর ভিতর যাবে কাজ করবে তারপর increment/decrement এ যাবে। তারপর আবার ending condition চেক করে সত্য হলে loop এর ভিতর যাবে কাজ করবে তারপর increment/decrement এ যাবে। এভাবে ending condition মিথ্যা না হওয়া পর্যন্ত কাজ করবে। মিথ্যা loop এর ভিতর আর যাবে না।
--	--

#Nisted for loop: যখন একটা for loop এর ভিতর আরেকটা for loop থাকে তখন তাকে Nisted for loop বলে। এখানে বাহিরের loop কে বলে outer loop আর ভেতরের loop কে বলে inner loop। outer loop এর এক বারের জন্যে inner loop তার ending condition মিথ্যা না হওয়া পর্যন্ত কাজ করবে। inner loop এর ending condition মিথ্যা হলে আবার outer loop এ আসবে। outer loop এর ending condition মিথ্যা না হওয়া পর্যন্ত চলতেই থাকবে। যানি কিছুই বোঝনি। নিচের কোড ভাল করে লক্ষ করো:-

#include<stdio.h> int main() { int i,j; for(i=1;i<=5;i=i+1) { for(j=1;j<=5;j=j+1) { printf("%d %d\n",i,j) ; } } }	1 1 1 2 1 3 1 4 1 5 2 1 2 2 2 3 . . 5 5	এই কোডে দেখা দুইটা loop একটা for(i=1;i<=5;i=i+1) তার ভিতর for(j=1;j<=5;j=j+1) এবং ২য় loop এর ভিতর printf("%d %d\n",i,j) । তারমানে যখন i=1 এবং 1<=5 condition সত্য হবে। তখন ভিতরের loop এ যাবে এখানে j=1,2,3,4,5 এর জন্যে condition সত্য হবে পারমানে প্রিন্ট করবে 1 1, 1 2, 1 3, 1 4, 1 5। যখন j=6 হবে তখন ভিতরের loop এর condition মিথ্যা হবে তখন আবার বাহিরের loop এর i=i+1 কাজ করবে। তারমানে i=2 হবে এবং 2<=5 condition সত্য হবে। তখন আবার ভিতরের loop এ যাবে আবার condition j<=5 এর 1,2,3,4,5 মানের জন্যে সত্য হবে আবার প্রিন্ট করবে 2 1, 2 2, 2 3, 2 4, 2 5। একই ভাবে i এর 3,4 ও 5 এর জন্যে ভিতরের loop কাজ করবে এবং প্রিন্ট করবে 3 1, 3 2, 3 3, 3 4, 3 5; 4 1, 4 2, 4 3, 4 4, 4 5; 5 1, 5 2, 5 3, 5 4, 5 5; যখন i=6 তখন বাহিরের loop এর condition মিথ্যা হবে তাই আর ভিতরের loop আ যাবে না।	একবার ভাবতো তুমি ইনপুট হিসাবে ৫ দিলে আর অউটপুট পাসের বক্স এর মত এল তবে কেমন লাগবে? এই রাকম স্কেএ মনে রাখবে বাহিরের লুপ প্রতিটি লাইন এর জন্যে কাজ করে আর ভিতরের লুপ প্রতি লাইনের প্রতিটি মান এর জন্যে কাজ করে। একবার ভাবতো এইটা কিভাবে সম্ভব? কোন আইডিয়া এলো? না এলে নিচের কোড ভাল করে লক্ষ করো:-	1 1 2 1 2 3 1 2 3 4 1 2 3 4 5
---	---	---	---	---

<pre>#include<stdio.h> int main() { int i,j,n; scanf("%d",&n); for(i=1;i<=n;i=i+1) { for(j=1;j<=i;j=j+1) { printf("%d ",j); } printf("\n"); } }</pre>	<p>এখানে দেখা আমি ইউজারের কাছ থেকে ইনপুট হিসাবে একটা নাম্বার নিলাম যা n এর মাঝে রাখছি। এইবার একটা লুপ ঘুরালাম 1 থেকে n পর্যন্ত। তারমানে প্রথমে i=1 এবং 1<=5 condition সত্য হবে তাখন ২নং লুপে যাবে। ২নং লুপে j=1 এবং j<=i যেহেতু লুপে j=1 এবং i=1 তারমানে 1<=1 condition সত্য হবে তাই ২য় লুপের ভিতর যাবে এবং প্রিন্ট করবে j এর মান 1। তারপর j=j+1 তারমানে j=2 তাই 2<=1 condition মিথ্যা হবে এবং ২নং লুপ থেকে বেরহয়ে ১নং লুপে যাবে যেখানে printf("\n") তারমানে নিউ লাইন প্রিন্ট করবে। তারপর i=i+1 বা i=1+1 বা i=2 এবং i<=n বা 2<=5 condition সত্য হবে আবার ২নং লুপে যাবে যেখানে j=1 এবং j<=i বা 1<=2 condition সত্য হবে এবং প্রিন্ট করবে j এর মান 1 তারপর j=j+1 বা j=1+1 বা j=2। j<=i বা 2<=2 সত্য হবে এবং প্রিন্ট করবে j এর মান 2। তারপর j=j+1 তারমানে j=3 তাই 3<=2 condition মিথ্যা হবে এবং ২নং লুপ থেকে বেরহয়ে ১নং লুপে যাবে যেখানে printf("\n") তারমানে নিউ লাইন প্রিন্ট করবে। আবার ১নং লুপের i=i+1 কাজ করবে তারমানে i=2+1 বা i=3 হবে i<=n বা 3<=5 সত্য হবে এবং আবার ২নং লুপে যাবে যেখানে j=1 এবং j<=i বা 1<=3 condition সত্য হবে এবং প্রিন্ট করবে j এর মান 1 তারপর j=j+1 বা j=1+1 বা j=2 এবং 2<=3 condition সত্য হবে এবং প্রিন্ট করবে j এর মান 2 তারপর j=j+1 বা j=2+1 বা j=3 এবং 3<=3 condition সত্য হবে এবং প্রিন্ট করবে j এর মান 3 তারপর j=j+1 বা j=3+1 বা j=4 এবং 4<=3 condition মিথ্যা হবে এবং ২নং লুপ থেকে বেরহয়ে ১নং লুপে যাবে যেখানে printf("\n") তারমানে নিউ লাইন প্রিন্ট করবে। আবার i=4 এর জন্যে j এর 1 2 3 4 মানের জন্যে ২নং লুপ কাজকরবে এবং 1 2 3 4 প্রিন্ট করবে। j=6 এর জন্যে condition মিথ্যা হবে এবং ১নং লুপে গিয়ে নিউ লাইন প্রিন্ট করবে। i=5 এর জন্যে j এর 1 2 3 4 5 মানের জন্যে ২নং লুপ কাজকরবে এবং 1 2 3 4 5 প্রিন্ট করবে j=6 এর জন্যে 6<=5 condition মিথ্যা হবে এবং ২নং লুপ থেকে বেরহয়ে ১নং লুপে যাবে যেখানে printf("\n") তারমানে নিউ লাইন প্রিন্ট করবে। i=6 এর জন্যে 6<=5 condition মিথ্যা হবে এবং ১নং লুপ থেকে বেরহয়েযাবে।</p>
---	--

While loop: While loop এক ধরনের লুপ। While loop কিছুটা if else আবার for loop এর মতো কাজ করে। While loop এর গঠনঃ

```
while(condition)
{
    Statement;
}
```

While loop এ for loop এর মতো initial condition, ending condition, increment/ decrement এবং statement থাকে। তবে এর initial condition While এর বাহিরে থাকে। increment/ decrement ভিতরে থাকে।

If else হিসাবে While loop: If else এ condition true হলে কাজ করে আর While loop এ যতখণ পর্যন্ত condition false না হবে ততখণ Statement এর কাজ করবে। নিচের কোড দেখো:

<pre>#include<stdio.h> int main() { char c; printf("Press s for stop\n"); scanf("%c",&c); while(c!='s') { printf("You don't press s\n press s for stop\n"); scanf("%c",&c); } }</pre>	<p>এই কোড রান করলে message দিবে Press s for stop যদি ইউজার s বা অন্য কোন ক্যারেক্টার ইনপুট দেয় তবে তা c এর ভিতর রাখবে। তারপর while এর condition হিসাবে চেক করবে c এর ভিতর যা আছে তা s কিনা? যদি s না হয় তবে while এর condition সত্য কারণ condition হলো c!='s' (অর্থাৎ c এর ভিতরের ভেলু 's' নাহলে)। যেহেতু condition সত্য তাই while এর Statement হিসাবে message দিবে You don't press s press s for stop আবার ইউজার এর কাছ থেকে ইনপুট নিবে। যতখণ না ইউজার ইনপুট হিসাবে s দিবে ততখণ সে একই কাজ করতেই থাকবে। যদি ইউজার ইনপুট হিসাবে s দেয় তবে থেমে যাবে।</p>
---	---

for loop হিসাবে While loop: for lop এ যেমন initial condition, ending condition, increment/decrement এবং statement থাকে তেমনি While loop এ ও থাকে এবং তা নিচের মতো:-

```
initial condition
while(ending condition)
{
    Statement;
    increment/decrement
}
```

উদাহরণ:

<pre>#include<stdio.h> int main() { int n=10 while(n!=0) { printf("%d ",); n=n-1; } }</pre> <p>Output: 10 9 8 7 6 5 4 3 2 1</p>	<pre>#include<stdio.h> int main() { int n=0 while(n<=10) { printf("%d ",); n=n+1; } }</pre> <p>Output:0 1 2 3 4 5 6 7 8 9 10</p>	<p>প্রথম কোডে initial condition: n=10 ending condition: n!=0 Statement: printf("%d ",) এবং decrement n=n-1। অর্থাৎ যতখণ না n=0 হবে ততখণ কাজ করতেই থাকবে।</p> <p>দ্বিতীয় কোডে initial condition: n=0 ending condition: n<=10 Statement: printf("%d ",) এবং increment n=n+1। অর্থাৎ যতখণ না n=11 হবে ততখণ কাজ করতেই থাকবে।</p>
---	---	--

Infinite while loop: while loop এর condition হিসাবে 1 দিলে infinite loop হয়ে যায়। অর্থাৎ এই loop আর কখন থামবে না যদি কিনা ending condition হিসাবে break ব্যবহার করা হয়।

<pre>#include<stdio.h> int main() { int n; scanf("%d",&n); while(n!=0) { int a,b,c; scanf("%d %d ",&a,&b); c=a+b; printf("%d\n",c); n=n-1; } }</pre>	<pre>#include<stdio.h> int main() { while(1) { int a,b,c; scanf("%d %d",&a,&b); if(a==0&&b==0) break; c=a+b; printf("%d\n",c); n=n-1; } }</pre>	<pre>#include<stdio.h> int main() { int a,b,c; while(scanf("%d %d",&a,&b)) { if(a==0&&b==0) break; c=a+b; printf("%d\n",c); n=n-1; } }</pre>
<p>এই কোডে প্রথমে ইউজার এর কাছ থেকে ইনপুট হিসাবে একটা নাম্বার নেওয়া হয়েছে এবং তা n এর ভিতর রাখা হয়েছে। while এর condition হিসাবে n!=0 ব্যবহার করা হয়েছে। তারমানে যতখণ না n=0 হবে ততখণ কাজকরবে এবং প্রতিবার ইনপুট হিসাবে a এবং b নিবে এবং অউটপুট হিসাবে যোগফল দেখাবে।</p>	<p>এইটা infinite loop তাই থামবে না। প্রতিবার ইউজার এর কাছ থেকে ইনপুট হিসাবে a এবং b নিবে এবং অউটপুট হিসাবে যোগফল দেখাবে। যহি ইউজার কখনো ইনপুট হিসাবে a=0 b=0 দেয় তবে লুপ থেমে যাবে।</p>	<p>এই কোডটা ও অনেকটা infinite loop ইউজার ইনপুট হিসাবে ইনপুট দিলে কাজকরবে না দিলে কাজ করবে না। যদি ইউজার ইনপুট হিসাবে a=0 b=0 দেয় তবে লুপ থেমে যাবে।</p>

Do while loop: Do while loop নিয়ম নিচেরমতো:-

```
do
{
    Statement;
} while(condition);
```

Do while loop এ প্রথমে do এর ভিতরের Statement এর কাজকরবে পরে while এর condition চেক করবে।

উদাহরণ:

<pre>#include<stdio.h> int main() { char c; do { printf("Press s for stop\n"); scanf("%c",&c); }while(c!='s'); }</pre>	<p>এইখানে প্রথমে do এর Statement এর কাজকরবে। অর্থাৎ Press s for stop প্রিন্ট করবে এবং ইউজার এর কাছ থেকে ইনপুট হিসাবে ক্যারাক্টার নিবে তারপর while এর condition হিসাবে c!='s' চেক করবে। যদি ইউজার ইনপুট হিসাবে s দেয় তবে লুপ থেমে যাবে আর s না দিলে আবার do এর Statement এর কাজকরবে। যখন ইনপুট হিসাবে s দিবে লুপ থেমে যাবে।</p>
--	---

অনুশীলনি

১। নিচের টেবেল দেওয়া প্রতিটি অউটপুট পাওয়ার জন্যে কোড লিখো: ইউজার ইনপুট হিসাবে n দেবে।

n=5 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3 4 4 4 4 4 5 5 5 5 5	n=5 * * * * * * * * * * * * * * *	n=3 * * * * * * * * * * * * * * *	n=4 *	n=5 * * * * * * * * * * * * * * *	n=5 * * * * * * * * * * * * * * *
n=3 * * * * * * * * *	n=3 1 1 2 1 2 3 1 2 1	n=4 1 2 3 2 3 4 5 4 3 4 5 6 7 6 5 4	n=5 1 1 2 1 2 3 1 2 3 4 1 2 3 4 5	n=5 1 1 2 1 2 3 1 2 3 4 1 2 3 4 5	N=5 1 2 3 4 5 1 2 3 4 1 2 3 1 2 1
n=5 * * * * * * * * * * * *	n=5 * * * * * * * * * * * *	n=5 * * * * * * * * * * * * * * *	n=5 * * * * * * * * * * * * * * *		