

সহজ ভাষায় প্রোগ্রামিং সি/সি++

আহমদ ওয়াহিদ মাসুদ

ফাংশনের ব্যবহার সি প্রোগ্রামের অত্যন্ত গুরুত্বপূর্ণ একটি বিষয়। আর ফাংশনের সবচেয়ে গুরুত্বপূর্ণ এবং জটিল বিষয় হলো রিকার্সন। রিকার্সন নিয়ে গত পর্বে ছোট্ট একটি উদাহরণ দেয়া হয়েছিল। কিন্তু রিকার্সন বিষয়টি ত্রিকমতো বুঝতে না পারলে সি ল্যাঙ্গুয়েজে আতঙ্কিত হওয়া সম্ভব নয়। এ লেখায় রিকার্সনের গুণ কিংু সহজ এবং জটিল উদাহরণ এবং তার মাধ্যমের সহজ ব্যাখ্যা দেয়া হয়েছে। মনে রাখা ভালো, রিকার্সন শুধু যে সি-তেই থাকে এমন কোনো কথা নেই। সি সব ল্যাঙ্গুয়েজের বেসিক আর তাই আজকের অনেক আধুনিক ল্যাঙ্গুয়েজ যেমন জাভা, সি #, এমনকি ওয়েব ডেভেলপমেন্টের ল্যাঙ্গুয়েজগুলোতেও রিকার্সনের বহুল ব্যবহার দেখা যায়।

প্রথমে রিকার্সনের মাধ্যমে কিভাবে ফিবনাক্সি নাম্বারের ধারা বের করা যায়, তা দেখানো হয়েছে। তার আগে জেনে রাখা ভালো, ফিবনাক্সি নাম্বার কাকে বলে। এটি একটি বিশেষ ধরনের ধারা। এ ধারটি শুরু হয় 1 থেকে (মতভেদে 0 থেকে)। নিচে ফিবনাক্সি নাম্বারের ধারটি দেয়া হলো:

$$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, \dots$$

$$F(n) = F(n-1) + F(n-2) \quad [F(0) = 0, F(1) = 1]$$

উপরের ধারাটি দেখেই বোঝা যাচ্ছে ফিবনাক্সি নাম্বার আসলে কেমন হতে পারে। সহজ কথায় বলতে গেলে, nতম ফিবনাক্সি নাম্বার হবে তার আগের দুটি নাম্বারের যোগফল। সুতরাং ওপরের সিরিজের সপ্তম নাম্বার ফলে 1+0, যা আগের দুটি নাম্বার 1 এবং 0-এর যোগফল।

ফিবনাক্সি নাম্বার কী তা দেখানো হলো। এখন বরা যাক, একটি সমস্যা দেয়া হলো যে nতম ফিবনাক্সি নাম্বার বের করার একটি প্রোগ্রাম লিখতে হবে। তাহলে শুধু রিকার্সন ব্যবহার করেই প্রোগ্রামটি কোড করতে পারবে কিনা সম্ভব হবে। অন্যভাবে শোভেও সম্ভব, কিন্তু রিকার্সন ব্যবহার না করলে কোড খুব কঠিন হয়ে যাবে। আরেকটি কথা জেনে রাখা ভালো, রিকার্সনের পাশাপাশি আরেকটি কনসেপ্ট আছে, যার নাম হলো ইটারেশন। রিকার্সন আর ইটারেশনের মধ্যে মূল পার্থক্য হলো, ইটারেশনে এমন কোনো ফাংশন থাকে না যে নিজেকে নিজে থেকে কল করে। অর্থাৎ সাধারণ রুল লুপ দিয়ে যদি এই সমস্যা সমাধান করা হয়, তাহলে বলা হবে কোডটি ইটারেশন ব্যবহার করে তৈরি করা হয়েছে। ইটারেশন এবং রিকার্সনের মধ্যে আরেকটি পার্থক্য হলো রিকার্সন ব্যবহার করে কোড করলে সেই কোডের গুণগত মান অনেক ভালো হয়। কারণ ইটারেশনের রান

```

টাইম রিকার্সনের থেকে অনেক বেশি।
এখন সমস্যটির সমাধান দেয়া হলো। এর
পরে তা ব্যাখ্যা করা হবে।
int fib(int m)
void main()
{
    int i,n;
    printf("enter the value of n:");
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        printf("%d\n",fib(i));
    }
}
int fib(int m)
{
    if(m<=2)
        return 1;
    else
        return fib(m-1)+fib(m-2);
}

```

লেখাই যাচ্ছে, এখানে রিকার্সন ব্যবহার করে কোড লেখা হয়েছে। fib() হলো এখানে রিকার্সিভ ফাংশন। প্রোগ্রামের কাজ হলো একটি নাম্বার ইনপুট নিতে হবে, সেই নাম্বারটি হবে n-এর ভ্যালু এবং প্রোগ্রাম তততম নাম্বার পর্যন্ত ফিবনাক্সি ধারটি প্রিন্ট করবে। এখন শুরুতেই বলা দরকার, এই প্রোগ্রামটি কেন ইটারেশন দিয়ে করা হলো না। যদি ইটারেশন দিয়ে করা হতো, তাহলে শুধু লুপ দিয়ে কোড করতে হতো, যেটি খুবই কষ্টসাধ্য একটি ব্যাপার। তাছাড়া কখনো কখনো ইটারেশন দিয়ে ত্রিকমতো কোড করাও সম্ভব হয়ে ওঠে না। কিন্তু রিকার্সন ব্যবহার করার জন্য কোড যেমন আকারে ছোট হয়েছে, তেমন সহজও হয়েছে। আর কোডের জটিলতা কম বলে এটি রান করতেও কম সময় লাগবে।

কোডের প্রথমেই ফাংশনের প্রোটোটাইপ দেয়া হয়েছে। ফাংশনে প্রোটোটাইপ কী জিনিস তা আগেও বলা হয়েছে, তারপরও আরেকবার বলে দেয়া হলো- যখন কোনো ইউজার ডিক্লাইড ফাংশন তৈরি করে তখন সেই ফাংশনের প্রোটোটাইপ দেয়া দরকার, যাতে প্রোগ্রাম বুঝতে পারে যে সেই নামে একটি ফাংশন আছে। এটি না দিলে এরর হওয়ার সম্ভাবনা থাকে। তাই প্রোটোটাইপ দেব কখনো বাদ না পড়ে, সেদিকে সবসময় খেয়াল রাখা দরকার।

প্রোটোটাইপের পরে মেইন ফাংশন শুরু হয়েছে। খেয়াল করলে দেখা যাবে মেইন ফাংশনেও কিন্তু লুপ ব্যবহার করা হয়েছে। এর অর্থ এই নয় যে, মূল প্রোগ্রামটি

ইটারেশনভিত্তিক। এখানে মূল লজিক বা মূল কাজ হলো fib() ফাংশনটি। সেটি রিকার্সন দিয়ে কল করা হয়েছে, তাই প্রোগ্রামটি রিকার্সনভিত্তিক।

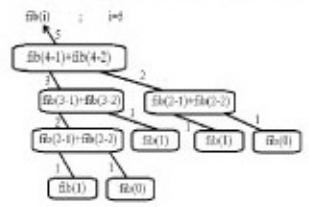
প্রথমে n-এর মান চাওয়া হয়েছে। এখানে n হলো সেই পজিশন যে পর্যন্ত ফিবনাক্সি নাম্বার বের করতে হবে। পরে ফর লুপে সেই পজিশন পর্যন্ত প্রোগ্রাম ফিবনাক্সি নাম্বার রিটার্ন করবে এবং প্রিন্ট করবে। এখানে প্রিন্ট ফাংশনের ভেতরে সরাসরি fib() ফাংশন কল করে মান প্রিন্ট করা হয়েছে। এটি ফাংশন কল করার আরেকটি উপায়। গত পর্বে ফাংশন কল করার বিভিন্ন উপায় দেখানো হয়েছে। এখানে সরাসরি ফাংশন কল করা হয়েছে এবং সরাসরি ফাংশনের রিটার্ন মানটি প্রিন্ট করা হচ্ছে। যেহেতু ফাংশনের রিটার্ন মান এখানে প্রিন্ট করা হচ্ছে, তাই এই ফাংশনের সবসময় একটি রিটার্ন ভ্যালু থাকবে। অর্থাৎ এই fib() ফাংশনটি সবসময় কোনো না কোনো মান রিটার্ন করবেই। আর ফাংশনটি যাতে কোনো মান রিটার্ন করতে পারে, তাই তার রিটার্ন টাইপের দিকে খেয়াল রাখতে হবে। যেমন এখানে ফাংশনটি সবসময় ইন্টজার টাইপের মান রিটার্ন করছে। তাই ফাংশনের রিটার্ন টাইপ হবে ইন্টজার। অর্থাৎ ফাংশনের নামের আগে int লিখতে হবে। এখানে রিটার্ন টাইপের গুণ খেয়াল রাখা খুবই গুরুত্বপূর্ণ। কারণ রিটার্ন টাইপ যদি ঠিক না থাকে তাহলে ফাংশনটি কোনো মান রিটার্ন করবে না। সুতরাং কোনো মান প্রিন্ট হবে না। যেকা নতুন প্রোগ্রামার তাদের কাছে এ বিষয়টি একটি সঠিক মনে হতে পারে। তাই অন্যভাবেও এই ফিবনাক্সি নাম্বারগুলো জিটি করা সম্ভব। এখানে কোডে মেইন ফাংশনের মধ্যে নাম্বারগুলো প্রিন্ট করা হয়েছে। তাই ইউজারকে ডিক্লাইড ফাংশনটির রিটার্ন টাইপের দিকে খেয়াল রাখতে হচ্ছে। কিন্তু ইউজার যদি সরাসরি fib() ফাংশনের ভেতরেই প্রিন্ট করার কথা চিন্তা থাকবে না। অর্থাৎ ফাংশনের রিটার্ন টাইপ নিয়ে আর খুব একটা চিন্তা করতে হবে না। দু'ভাবেই কাজটি করা যায়। যেভাবে সহজ মনে হবে, সেভাবেই করা উচিত। তাহলে পরে কোড বুঝতে সমস্যা হবে না।

লোকাল প্যারামিটার আর ফরমাল প্যারামিটার নিয়ে যাতে কোনো সমস্যা না থাকে, তাই এটি নিয়ে আলোচনা করা হয়েছে। একটি ফাংশনের প্যারামিটার কী, তা আমরা জানি। ফাংশনের পরে প্রথম বন্ধনীর মাঝে যে মান বা ভেরিয়েবল থাকে তাইই ফাংশনের প্যারামিটার বলা হয়। প্যারামিটার ব্যবহারের উদ্দেশ্য একটাই, তা হলো এক ফাংশন থেকে আরেক ফাংশনে ভেরিয়েবল বা মান পাঠানো। যেমন মেইন ফাংশনের যে ডিক্লার জিটের কমান্ড আছে সেখানে ভালোমতো লক্ষ করলে দেখা যাবে, সেখানে fib() ফাংশনকে কল করা হয়েছে এবং এইই সাথে সেই ফাংশনে একটি মান পাঠানো হয়েছে। সেই মানটি হলো ভেরিয়েবল i। এখানে ফাংশনের প্রথম বন্ধনীর ভেতরে যেহেতু একটি ভেরিয়েবল লেখা হয়েছে এবং

যেহেতু এর মাধ্যমেই ভেরিয়েবল পাঠানো হচ্ছে, তাই এটি হলো ফাংশনটির প্যারামিটার। এখন কোডের শুরুতে দেখা যাক। মেইন ফাংশন শুরু হওয়ার আগেই fib() ফাংশনের যে প্রোটোটাইপ লেখা আছে সেখানেও প্রথম বন্ধনীর ভেতরে একটি ভেরিয়েবল লেখা আছে। সেখানে int m লেখা আছে। এটি কোনো ভেরিয়েবল পালন করছে না বা কোনো ভেরিয়েবল রিসিভও করছে না। তারপরে এটি ফাংশনটির প্যারামিটার। লক্ষ করলে দেখা যাবে, মেইন ফাংশনের নিচে সেখানে fib() ফাংশনটির শুরু হয়েছে সেখানেও প্যারামিটার হিসেবে int m লেখা আছে। মেইন ফাংশনের ভেতরে যে প্যারামিটার ছিল সেটি হলো আ্যকচুয়াল প্যারামিটার। আর প্রোটোটাইপে বা ফাংশনের বন্ধনীর শুরুতে যে প্যারামিটার আছে, তা হলো ফরমাল প্যারামিটার। এ দুই ধরনের প্যারামিটারের মাঝে যে মূল পার্থক্য আছে, তা সবসময় খোয়াল রাখতে হবে। আর তা হলো আ্যকচুয়াল প্যারামিটারের সরাসরি ভেরিয়েবল বা মান দেয়া থাকে। অর্থাৎ যতক পঠানো হবে, তাই আ্যকচুয়াল প্যারামিটার। আর যে গ্রহণ করবে, সে হলো ফরমাল প্যারামিটার। কিন্তু ফরমাল প্যারামিটার সবসময় একই থাকে। ফরমাল প্যারামিটারকে জেনারেল ফরমে লেখা হয়। অর্থাৎ ফরমাল প্যারামিটার কখনো পরিবর্তন হয় না, বরং তা সব আ্যকচুয়াল প্যারামিটারের জন্যই কাজ করে। আর এ দু'ধরনের প্যারামিটার যেভাবে কাজ করে তা এখানে সহজে বলে দেয়া হলো। এই কোডে fib() এর প্যারামিটার হিসেবে i-কে পঠানো হচ্ছে এবং int m তাকে গ্রহণ করছে। অর্থাৎ m-এর মান হয়ে গেল i; খোয়াল রাখতে হবে; এবং m-এর ভাটি টাইপ ফেল একই হয়।

ইউজারের জন্য সবচেয়ে গুরুত্বপূর্ণ বিষয়টি হলো, রিকার্সিভ ফাংশনের ভেতরের কাজটি কিভাবে হচ্ছে তা ট্রিকমতকো বুঝতে পারা। অর্থাৎ রিকার্সনের প্রধান লজিকটুকু ধরতে পারা। এখানে রিকার্সনের প্রধান অংশে কিছ একটাই, তা হলো ইউজার ডিফাইন্ড ফাংশনের একদম শেষের লাইনটি। তবুও সহজে বোঝার জন্য পুরো ফাংশনের বাখ্যা দেয়া হয়েছে। রিকার্সনের লজিক ব্যাখ্যা করার আগে মূল সমস্যার কথা একবার চিন্তা করা যাক। মূল সমস্যা হলো একটি বিশেষ ধরনের ধারা খ্রিষ্ট করতে হচ্ছে, কিন্তু কতদূর খ্রিষ্ট করতে হবে সেটি ইউজার ইনপুট দিয়ে নির্ধারণ করে দেনে। এটা বিশেষ ধারারটি দিকে এগাবার নজর দেয়া যাক। ধারারটির প্রথম দুটি সংখ্যা একই। এরপর থেকে ভিন্ন সংখ্যা শুরু হয়েছে। ধরুন, এই ধারারটি একটি ফর লুপের বিভিন্ন পর্যায়। লুপটির পর্যায় শুরু হয়েছে i-1 থেকে। এখন প্রথম দুটি পর্যায় লুপটি 1 খ্রিষ্ট করবে, এরপর থেকে ভিন্ন সংখ্যা। তাহলে বলা যায় যে i-এর মান 2 হওয়া পর্যন্ত একই মান খ্রিষ্ট হওয়া শুরু হবে। এখন i-এর মানটিকেই প্যারামিটার হিসেবে পাঠানো হচ্ছে। তার মান i-এর মান এবং m-এর মান সমান। অর্থাৎ m-এর মান

যতখন 2-এর কম থাকবে ততখন ফাংশনটি 1 রিটার্ন করবে। কারণ ফাংশনটি 1 রিটার্ন করলেই সেই 1 খ্রিষ্ট হবে। আর m-এর মান যখন 2-এর বেশি হবে তখন ফাংশনটি ভিন্ন সংখ্যা রিটার্ন করবে এবং সেটি খ্রিষ্ট হবে। তাই নিচের ফাংশনটির প্রথম কন্ডিশন হলো যতখন m-এর মান 2-এর ছোট বা সমান হবে ততখন পর্যন্ত 1 রিটার্ন করবে। এখন ধরা যাক, ফর লুপটির পর্যায় 2-এর বেশি হয়ে গেল। তাহলে ফাংশনটি আর 1 রিটার্ন করবে না, তখন ভিন্ন সংখ্যা রিটার্ন করবে। এখন এই ভিন্ন সংখ্যাটি কী হবে, তা নির্ধারণ করা এই রিকার্সনের তথ্য এই প্রোগ্রামের মূল কাজ। আমরা জানি, ফিবনাক্সি ধারার যেকোনো সংখ্যা তার আগের দুটি সংখ্যার যোগফলের সমান। অর্থাৎ যদি nতম সংখ্যা নির্ণয় করতে বলা হয়, তাহলে তা (n-1) এবং (n-2)-এর যোগফলের সমান হবে। ধরা যাক, তৃতীয় সংখ্যাটি বের করতে বলা হয়েছে। তার মান প্রথম এবং দ্বিতীয় সংখ্যার যোগফল বের করতে বলা হলো। লক্ষ্যণীয়, i-এর মান 2 বা এর কম হলে fib() ফাংশনটির সরাসরি একই সংখ্যা রিটার্ন করে। রিকার্সনে মূলত এই ধারারটিই ব্যবহার করা হয়েছে। fib() ফাংশনের দ্বিতীয় শর্ত তখনই কাজ করবে, যখন ফর লুপের পর্যায় 2-এর বেশি হবে। এখানে যে কমান্ডটি



লেখা হয়েছে সেটি রিকার্সনের কমান্ড। এখানে [fib(m-1)+fib(m-2)] রিটার্ন করার কমান্ড দেয়া হয়েছে। বোকাই যাচ্ছে, এখানে যেহেতু ফাংশনটি নিজেই নিজেকে কল করছে, তাই এটি রিকার্সন। nতম পজিশনের আগের দুটি পজিশনের মান বের করে যোগ করা হয়েছে এই কমান্ডের মূল কাজ। আর এখানে প্যারামিটার হিসেবে (m-1) এবং (m-2) পাঠানো হচ্ছে। অর্থাৎ ফাংশনটি নিজেকে কল করছে এবং আগের দুটি পজিশনের মান বের করার কমান্ড দিচ্ছে। আর মান দুটি বের করা হলে তাদের যোগফল মেইন ফাংশনে রিটার্ন করা হচ্ছে।

চিত্র-3-এ দেখানো হলো কিভাবে রিকার্সিভ ফাংশন কাজ করে। ধরা যাক, ফর লুপ পঞ্চম পর্যায় চলছে। সুতরাং i-এর মান এখন 5। তাহলে fib() ফাংশনে m-এর মান হবে 5। তাই ফাংশনের দ্বিতীয় কন্ডিশনে প্রোগ্রাম চলে যাবে। সেখানে ফাংশনটি নিজেই নিজেকে আবার কল করছে, কিন্তু এখানে প্যারামিটার হিসেবে ভিন্ন মান পাঠানো হচ্ছে। চিত্রে দেখানো করলে দেখা যাবে, কিভাবে ফাংশনের রিকার্সন হয়ে একটি কিভাবে প্যারামিটারের মান কমছে। কয়েক কমান্ডে একসময় প্রভ্রিটি ফাংশনের

প্যারামিটারের মান 2-এর কম হবে, তখন ফাংশন সরাসরি 1 রিটার্ন করবে। এভাবে রিটার্ন করতে করতে fib() ফাংশনকে যথোনে কল করা হয়েছে সেখানে 5 রিটার্ন হবে। এখন ফিবনাক্সি সংখ্যার ধারারটি বেয়াল করতে দেখা যাবে, পঞ্চম ধরনের সংখ্যাটি 5। এভাবে রিকার্সনের মাধ্যমে কন্ডিশন কোনো প্রোগ্রাম সহজেই সমাধান করা সম্ভব।

কোনো প্রোগ্রাম রিকার্সন ব্যবহার করা যেমন খুব সহজ আবার তেমনি খুব জটিল। রিকার্সনে কোড খুব কম লিখতে হয়। কিন্তু রিকার্সনের মূল লজিক বোঝাটা বেশ কঠিন। যেমন এই প্রোগ্রামে ইউজার ডিফাইন্ড ফাংশনের দ্বিতীয় কন্ডিশনটি হলো পুরো প্রোগ্রামের রিকার্সনের মূল লজিক। এই অংশটুকু যে ভালোভাবে বুঝতে পারবেন পুরো প্রোগ্রাম বোঝা তার জন্য কোনো ব্যাপারই নয়। কিন্তু শুধু কোড দেখে এই রিকার্সন লজিক বোঝা বেশ কঠিন একটি কাজ। তা হাতা হাতে আয়ত্তাভাণ্ড প্রোগ্রামিংয়ের কোর্সায় পড়তে। আর এটাও সত্যি যেকোনো কিছু বোঝা যত কঠিন তার থেকে জিনিগিটি তৈরি করা আরও বেশি কঠিন। তাই সবচেয়ে ভালো উপায় হলো, এ ধরনের রিকার্সন লজিক যখন তৈরি করা হয় তখন একটি ছবি একে দেখা। এখানে রিকার্সনের যে ডিগ্রি দেয়া হয়েছে, কেউ কেউ শুধু সেই ডিগ্রিটি দেখে রিকার্সনের লজিক ধরতে পারবেন। তাই এরকম জটিল কোড শেয়ার সময় যদি মূল লজিকের একটি চিত্র আঁকা যায় তাহলে অনেক সহজে সেই ডিগ্রি দেখে লজিক সম্পূর্ণ ক্যাঁয়া যায় এবং সে অনুযায়ী কোড লেখা যায়।

যদিও রিকার্সন খুব উন্নতমানের কেজিঞ্জয়ের একটি বৈশিষ্ট্য এবং এটি দিয়ে কোডের আকার অনেক ছোট হয়ে যায়। রিকার্সন ব্যবহারের সময় কিছু জিনিগি খুব ভালোভাবে খোয়াল করতে হবে। প্রথমেই বলা যাক, ইনফাইনিট রিকার্সনের ব্যাপারে। রিকার্সনের লজিক বোঝা এবং তৈরি করা কোনো সহজ কাজ নয়। যেখানে সময় ছোট বা বড় যেকোনো ধরনের ভুল হতেই পারে। কখনো কখনো রিকার্সনের লজিকে কিছু সূক্ষ্ম ভুল হয়ে থাকে, যার জন্য ফাংশন তার শেষ খুঁজে পায় না। যেমন রিকার্সনের কন্ডিশনে যদি সূক্ষ্ম ভুল হয়ে থাকে, যার জন্য ফাংশন তার শেষ খুঁজে পায় না। যেমন রিকার্সনের কন্ডিশনে যদি ভুলবশত কোনো কোনো শর্ত দেয়া হয় যাতে ফাংশন কখনো পৌঁছায় না, তাহলে ফাংশন তার শেষ খুঁজে পাবে না। অর্থাৎ রিকার্সনটি কখনই শেষ হবে না। যেটি একটি ইনফাইনিট ফাংশনের বা রিকার্সনের সৃষ্টি করবে। এরকম অবস্থায় আসলে প্রোগ্রাম খেঁচো থাকবে এবং যাই করতে বলা হোক না কেন কিছুই হবে না। আপাতা দুষ্টিতে দেখলে মনে হবে প্রোগ্রাম খেঁচো থাকে, কিন্তু আসলে এ অবস্থায় প্রোগ্রাম চলতে থাকে। যেহেতু প্রোগ্রাম কোনো কমান্ড শেষ না করা পর্যন্ত পরের কমান্ডে যায় না এবং যেহেতু ইনফাইনিট হওয়া মানে কোনো কমান্ডের এক্সিকিউশনের শেষ না হওয়া, তাই প্রোগ্রামের রিকার্সন আর কখনো শেষ হয় না অর্থাৎ প্রোগ্রাম পরবর্তী কমান্ডে আর যেতে পারে না। তাই প্রোগ্রামটি ইনফাইনিট হয়ে যায়।

প্রোগ্রামিং সি/সি++

(৭৪ পৃষ্ঠার পর)

উদাহরণস্বরূপ- দেয়া ফিফনাল্ডির প্রোগ্রামটির কথাই কলা যায়। এখানে fib() ফাংশনের ভেতরে প্রথম কন্ডিশনে কেউ যদি ভুলবশত if(m>20) লেখে এবং প্রোগ্রাম রান করার পর ইনপুট ৫ দেয়, তাহলে প্রোগ্রামটি ইনফাইনিটি রিকার্সিভ হয়ে যাবে এবং বিভিন্ন সংখ্যা অনবরত প্রিন্ট করেই যাবে। কেননা ইনপুট যদি ৫ হয়, তাহলে প্রোগ্রাম প্রথম কন্ডিশনে পড়বে না, দ্বিতীয় কন্ডিশনে যাবে। কিন্তু দ্বিতীয় কন্ডিশনে রিকার্সন করা হয়েছে এবং প্যারামিটার হিসেবে আগের মান কমিয়ে দেয়া হয়েছে। তাই রিকার্সনের পরবর্তী পর্যায়ে আরেকটি fib() ফাংশন তৈরি হবে, যেখানে প্যারামিটার হিসেবে যাবে ৪ এবং ৩, এরপর আবার রিকার্সন হবে এবং প্যারামিটারের মান কমে যাবে। অর্থাৎ প্রোগ্রাম কখনই প্রথম কন্ডিশনে যাবে না, সবসময় দ্বিতীয় কন্ডিশনেই যেতে থাকবে। আর প্রথম কন্ডিশনে না গেলে সরাসরি কোনো মান রিটার্নও হবে না, রিকার্সন থাকবেও না। তাই প্রোগ্রাম ইনফাইনিট হয়ে যাবে। প্রোগ্রাম যদি একবার ইনফাইনিট হয়ে যায়, তখন তা বন্ধ করা হাড়া আর কোনো উপায় থাকে না। এ কারণে ইউজার যদি এডিটর হিসেবে টার্বো সি ব্যবহার করেন, তাহলে কিছুকণ পরপরই সেভ করা উচিত। টার্বো সি-তে সেভ করার সহজ উপায় হলো F2 চাপা। আর কেউ যদি আধুনিক ৩২ বিটের কোনো এডিটর যেমন ভিজুয়াল সি, ডেভ

সি, কোড ব-ক ইত্যাদি ব্যবহার করেন, তাহলে সেভ না করলেও তেমন সমস্যা হয় না।

রিকার্সনের দ্বিতীয় সমস্যা হলো লজিকে ভুল। এমনও হয় যে ইউজার ঠিকমতো লজিক ডিজাইন করলেন, লজিক অনুযায়ী কোডও করলেন, প্রোগ্রাম ঠিকমতো রানও করল, কিন্তু ফল ঠিক এলো না। এ ধরনের সমস্যা খুবই কমন এবং এক্ষেত্রে সাধারণত লজিকে ভুল থাকে না, কোডে ভুল থাকার সম্ভাবনাই বেশি থাকে। আর রিকার্সনের কোড সাধারণত খুব ছোট হয়ে থাকে। তাই এ ধরনের রিকার্সনের কোডে ভুল হলে তা বের করা খায় কঠিন ব্যাপার হয়ে দাঁড়ায়। কোড যত ছোট তা ইউজারের জন্য বোঝা তত কঠিন। তাই এসব ক্ষেত্রে সবচেয়ে ভালো উপায় হলো ছবি ঠেকে দেয়া। তারপর ছবি ধরে ধরে কোড মিলিয়ে দেখা যে কোথাও ভুল হলো কি না।

আরেকটি ছোট সমস্যা হলো রিকার্সনের কোড করার সময় মাঝে মাঝে ডেরিয়েবলের স্কেপ নিয়ে ঝামেলা দেখা যায়। রিকার্সনের মাঝে ভুলে এমন কোনো কোড লেখা হয়, যার ডেরিয়েবলের স্কেপ সেই ফাংশনের বাইরে। সেক্ষেত্রে এরর দেখা দেবে। তবে এটি তেমন বড় কোনো সমস্যা নয়, কারণ এ ধরনের ভুল হলে কম্পাইলার এরর দেবে এবং ইউজার আগে থেকেই বুঝতে পারবেন। সে ধরনের ভুলই সবচেয়ে বেশি বিপজ্জনক যেগুলোর জন্য কম্পাইলার কোনো এরর দেয় না, কিন্তু প্রোগ্রামের ফল ভুল আসে।

রিকার্সন প্রোগ্রামিং জগতের খুবই গুরুত্বপূর্ণ

একটি বিষয়। এটি ব্যবহার করা যেমন সহজ, ডিজাইন করা তেমন কঠিন। অনুশীলনের মাধ্যমেই এ ব্যাপারে পারদর্শী হওয়া সম্ভব।

ফিডব্যাক : wahid_cseaut@yahoo.com

কারুকাজ বিভাগে লেখা আহ্বান

কারুকাজ বিভাগের জন্য প্রোগ্রাম, সফটওয়্যার টিপস আহ্বান করা হচ্ছে। লেখা এক কলামের মধ্যে হলে ভালো হয়। সফট কপি সহ প্রোগ্রামের সোর্স কোডের হার্ড কপি প্রতি মাসের ২৫ তারিখের মধ্যে পাঠাতে হবে।

সেরা ৩টি প্রোগ্রাম/টিপস-এর লেখককে যথাক্রমে ১,০০০ টাকা, ৮৫০ টাকা ও ৭০০ টাকা পুরস্কার দেয়া হয়। এ ছাড়াও প্রোগ্রাম/টিপস মানসম্মত বিবেচিত হলে, তা প্রকাশ করে প্রচলিত হারে সন্মানী দেয়া হয়।

প্রোগ্রাম/টিপস-এর লেখকদের নাম কর্মপত্রটির জগৎ এর বিসিএস কর্মপত্রটির সিটি অফিস থেকেও জানা যাবে। পুরস্কারের টাকা কর্মপত্রটির জগৎ-এর বিসিএস কর্মপত্রটির সিটি অফিস থেকে সংগ্রহ করতে হবে। সংগ্রহের সময় অবশ্যই পরিচয়পত্র দেখাতে হবে। এক পুরস্কার চলতি মাসের ৩০ তারিখের মধ্যে সংগ্রহ করতে হবে।