

প্রোগ্রামিংয়ের মূল লক্ষ্য হলো, ইউজার তার ইচ্ছেমতো কাজ কমপিউটার দিয়ে করিয়ে নেবে। সে কাজটি হতে পারে কোনো সমস্যার সমাধান, কোনো কিছু নিয়ন্ত্রণ অথবা শুধু বিনোদন দান করা। প্রতিটি কাজ ইউজারকে বিভিন্ন কন্ডিশন বা শর্ত মেনে নিয়ে করতে হয়। একটি কন্ডিশন সত্য হলে একটি বিশেষ কাজ সম্পাদন হবে, মিথ্যা হলে অন্য একটি কাজ সম্পাদন হবে।

প্রতিটি প্রোগ্রামই আসলে অনেকগুলো কন্ডিশনাল অপারেশনের সমষ্টি। বিভিন্ন ধরনের কন্ডিশনাল অপারেশন আছে। যেমন : if-else, switch case, while loop ইত্যাদি। বিভিন্ন কন্ডিশনাল অপারেশনের কাজ কী তা নিয়ে ইউজারের যাতে বিভ্রান্তি সৃষ্টি না হয়, তাই নিয়ে এ লেখার অবতারণা।

কন্ডিশনাল অপারেশনের কর্মপদ্ধতি

একটি কন্ডিশনাল স্টেটমেন্ট বা কোনো কন্ডিশনাল অপারেশনের সাধারণত দুটি অংশ থাকে। একটি কন্ডিশন অংশ এবং অপরটি স্টেটমেন্ট অংশ। কন্ডিশন অংশে একটি এক্সপ্রেশন ব্যবহার করা হয়, যার মান থেকে প্রোগ্রাম বুঝতে পারে যে কন্ডিশনটি সত্য না মিথ্যা। শূন্য ছাড়া অন্য যেকোনো মানকে প্রোগ্রাম সত্য বলে ধরে নেয়। যেমন : a=2; b=3; হলে যদি কন্ডিশনের এক্সপ্রেশন হিসেবে a<b এবং a+b ব্যবহার করা হয়। তবে কন্ডিশনের মান হবে 1 তথা সত্য। আর যদি a>b এবং (a-b+1) ব্যবহার করা হয় তাহলে কন্ডিশনের মান হবে 0 তথা মিথ্যা। এভাবে সরাসরি রিলেশনাল অপারেটর দিয়ে বা সাধারণ ম্যাথমেটিক্যাল এক্সপ্রেশন দিয়ে কোনো কন্ডিশনের মান নির্ধারণ করা যায়। আবার কন্ডিশনের মান হিসেবে যদি সরাসরি কোনো সংখ্যা বা মান বা ভেরিয়েবল দেয়া হয়, তাহলে সরাসরি সে মানের ওপর ভিত্তি করে কন্ডিশনের মান ঠিক করা হয়। যেমন কোনো কন্ডিশনের এক্সপ্রেশন হিসেবে শুধু 3 ব্যবহার করা হলে সেটি সত্য হবে, কারণ 3 হলো একটি অশূন্য (নন-জিরো) সংখ্যা।

এখন বিভিন্ন কন্ডিশনাল/লুপিং কিওয়ার্ড ব্যবহার করে কিভাবে কন্ডিশনাল স্টেটমেন্ট তৈরি করতে হয়, তা দেখা যাক। লক্ষণীয়, কোনো কিওয়ার্ডের সাথে ব্যবহার হওয়া স্টেটমেন্ট বা কোড ব্লককে ওই কিওয়ার্ড সংশ্লিষ্ট স্টেটমেন্ট বলে। যেমন : if statement, while statement ইত্যাদি।

If statement

কন্ডিশন নিয়ে কাজ করার জন্য বহুল ব্যবহৃত পদ্ধতি হলো if স্টেটমেন্ট। প্রথমে একটি বাক্য লক্ষ করা যাক; যদি বৃষ্টি হয়, তাহলে আমরা যাব না। এখানে একটি কাজ হলে আরেকটি কাজ করা হবে। অর্থাৎ প্রথম কাজের ওপর দ্বিতীয় কাজ হবে কি হবে না তা নির্ভর করছে। সি ল্যাঙ্গুয়েজে এ ধরনের কন্ডিশন নিয়ে কাজ করার জন্য সাধারণত if স্টেটমেন্ট ব্যবহার হয়। আর

if স্টেটমেন্টের সাথে কন্ডিশন হিসেবে একটি এক্সপ্রেশন থাকে, যা নির্ধারণ করে যে কোন কাজটি সম্পন্ন করা হবে। প্রোগ্রামে if-কে তিনভাবে ব্যবহার করা যায়। যেমন সাধারণ if হিসেবে, if-else স্টেটমেন্ট হিসেবে অথবা else if চেইন হিসেবে।

সাধারণ if-এর কাজ খুবই সহজ। একটি মাত্র কন্ডিশন দেয়া থাকে এবং সেটি সত্য হলে কোনো কাজ সম্পন্ন হবে, না হলে সম্পন্ন হবে না। যেমন :

```
if(age>=18)
    printf("You are mature.");
if(age<=18)
    printf("You are immature.");
```

এখানে প্রতিটি if-এর নিচে যে প্রিন্ট করার কমান্ড দেয়া হয়েছে তা হলো কাজ, আর if-এর ডান পাশে প্রথম বন্ধনীর মাঝে যা আছে তা হলো কন্ডিশন। এই কন্ডিশনটিই নির্ধারণ করে যে সংশ্লিষ্ট if স্টেটমেন্টটি কার্যকর হবে কি না।

উপরের প্রোগ্রামটি if-else দিয়েও করা যায়।

যেমন :

```
if(age>=18)
    printf("You are mature.");
else
    printf("You are immature.");
```

সহজ ভাষায় প্রোগ্রামিং সি/সি++

আহমদ ওয়াহিদ মাসুদ

এখানে age-এর মান 1৮ বা তার বেশি হয় হলে প্রথম লাইনটি প্রিন্ট, অন্যথায় দ্বিতীয় লাইনটি প্রিন্ট হবে।

অনেকগুলো কন্ডিশন থাকলে else-if চেইন ব্যবহার করা যায়। যেমন :

```
if(age>=50)
    printf("You are old.");
else if((age>=25)&&(age<50))
    printf("You are young");
else if((age>=18)&&(age<25))
    printf("You are mature");
else if((age>=10)&&(age<18))
    printf("You are a boy");
else
    printf("You are a child");
```

উপরের else-if চেইনে কয়েকটি বিষয় লক্ষণীয়। if-এর সাথে যেমন কোনো কন্ডিশন ব্যবহার হয়, তেমনি else-if-এর সাথেও কন্ডিশন থাকে। তবে শুধু else-এর সাথে কোনো কন্ডিশন থাকে না। এটি হলো ডিফল্ট অপারেশন। অর্থাৎ যখন কোনো কন্ডিশনই সত্য হবে না, তখন এই ডিফল্ট অপারেশনটি কার্যকর হবে। if বা else বা if-else কিওয়ার্ডের পর কোনো কন্ডিশন থাকুক বা না থাকুক কোনো সেমিকোলন দেয়া যাবে না। তবে তাদের অধীনে যেসব স্টেটমেন্ট থাকে তাদের শেষে সাধারণ নিয়মানুযায়ী সেমিকোলন ব্যবহার করতে হবে। উপরের প্রতিটি if-else-এর কন্ডিশনগুলো একটু জটিল মনে হবে, আসলে তা নয় বরং এখানে দুটি কন্ডিশন একত্রে ব্যবহার করা হয়েছে। পরে দুটি কন্ডিশনকে && (অ্যান্ড) দিয়ে যুক্ত করা

হয়েছে। এখানে & ব্যবহার না করে && ব্যবহার করা হয়েছে, কারণ & হলো বিটওয়াইজ অপারেটর এবং && হলো বাইনারি অপারেটর। কন্ডিশন && দিয়ে যুক্ত করার মানে হলো দুটো কন্ডিশনই যদি সত্য হয়, তাহলে পুরো কন্ডিশনটি সত্য হবে। যেমন উপরের উদাহরণটিতে তৃতীয় কন্ডিশনের দিকে খেয়াল করলে বোঝা যাবে, ওই কন্ডিশনটি তখনই সত্য হবে যখন age-এর মান 1৮-এর সমান অথবা বড় হবে এবং age-এর মান ২৫-এর কম হবে, অর্থাৎ দুটো কন্ডিশনই সত্য হবে। কিন্তু এমন যদি হতো যে (age>=18) এই কন্ডিশনটি শুধু সত্য হতো, কিন্তু (age<25) এই কন্ডিশনটি মিথ্যা হতো, তখন ((age>=18)&&(age<25)) এই পুরো কন্ডিশনটি মিথ্যা হবে। যদি যেকোনো একটি কন্ডিশনের সত্য হওয়ার জন্য পুরো কন্ডিশনটি সত্য করতে হয় তাহলে || (অথবা) এই অপারেটরটি ব্যবহার করতে হবে। সুতরাং তখন পুরো কন্ডিশনটি হবে ((age>=18)||age<25))।

কখনো কখনো এমন কিছু অবস্থার সৃষ্টি হতে পারে যে একটি শর্ত সত্য হলে কোনো কাজ অপর কোনো শর্তের ওপর নির্ভর করবে এবং

আগের শর্তটি মিথ্যা হলে কাজটি অন্য কতগুলো শর্তের ওপর নির্ভর করবে। উদাহরণস্বরূপ, রহিম যদি SSC পরীক্ষায় পাস করে থাকে, তাহলে সে কলেজে পড়তে পারবে, আর যদি ফেল করে, তাহলে সে কলেজে পড়তে পারবে না। আবার রহিম যদি কলেজে পড়তে পারে, তাহলে তার জিপিএ চেক করে দেখা হবে, যদি সে ৩.৫-এর ওপরে পায় তাহলে সে সায়েন্সে পড়তে পারবে, অন্যথায় পারবে না। এ ধরনের সমস্যা সমাধানের জন্য nested if else ব্যবহার করা হয়। অর্থাৎ একটি if-else-এর মধ্যে আরেকটি if-else থাকতে পারে, একাধিকও থাকতে পারে। একটি প্রোগ্রাম উদাহরণ হিসেবে দেয়া হলো।

```
if(score>=90)
{
    if(grade=='A')
        printf("grade is A");
    else
        printf("grade is not A");
}
else
    printf("disqualified!");
```

এখানে ন্যেস্টেড if-else ব্যবহার করা হয়েছে। প্রথমে আউটার if-এর কন্ডিশন যদি সত্য হয় অর্থাৎ স্কোরের মান যদি ৯০-এর সমান অথবা বেশি হয় তাহলে প্রোগ্রাম ইনার if-এ ঢুকবে, আর যদি মিথ্যা হয় তাহলে প্রিন্ট করবে "disqualified!"। প্রোগ্রাম যদি ইনার if-এ ঢুকে তাহলে চেক করবে যে গ্রেডের মান A কিনা, যদি A হয় তাহলে প্রিন্ট করবে "grade is A" আর

যদি না হয় তাহলে প্রিন্ট করবে “grade is not A”।

Switch case

এটি if-এর মতো একটি কন্ডিশনাল অপারেটর। এর ব্যবহারও if-এর মতো। তাই বিষদ আলোচনা না করে সরাসরি একটি উদাহরণ এবং তার ব্যাখ্যা দেয়া হলো।

```
int grade=8;
switch(grade)
{
    case 10:
    case 9:
        printf("grade=9");
        break;
    case 8:
        printf("grade=8");
        break;
    case 7:
        printf("grade=7");
        break;
    default:
        printf("no score!");
        break;
}
```

উপরের উদাহরণ দেখেই ধারণা করা যায় কিভাবে সুইচ কেস কাজ করে। সুইচের পাশে প্রথম বন্ধনীর ভেতরে এক্সপ্রেশন থাকে। সেই এক্সপ্রেশনের মান নন-জিরো (অশূন্য) হলে প্রোগ্রাম কেসটির ভেতরে ঢুকবে, অন্যথায় ঢুকবে না। এখন সেই এক্সপ্রেশনটি যেই মান রিটার্ন করবে প্রোগ্রাম তত নম্বর কেসে প্রবেশ করবে। যেমন এখানে এক্সপ্রেশন হিসেবে শুধু grade ব্যবহার করা হয়েছে। এর অর্থ গ্রেডের মান যত হবে প্রোগ্রাম তত নম্বর কেসে প্রবেশ করবে। আর গ্রেডের মান শূন্য হলে প্রোগ্রাম এই সুইচ কেসে প্রবেশই করবে না। এখানে গ্রেডের মান হলো ৮। তাই প্রোগ্রাম ৮ নম্বর কেসে প্রবেশ করবে। তারপর উল্লিখিত লাইনটি প্রিন্ট করবে। তারপর ব্রেক করবে অর্থাৎ প্রোগ্রাম এই কেস থেকে বের হয়ে যাবে, কিন্তু বন্ধ হয়ে যাবে না, কেসের পরবর্তী কাজ সম্পাদন করবে। সুতরাং ব্রেকের কাজ হলো প্রোগ্রাম যেই লুপ/কেসে আছে সেখান থেকে বের হয়ে যাওয়া। তবে if-else কোনো লুপ/কেস নয়, সে ব্যাপারে সবসময় খেয়াল রাখতে হবে। এখন কথা হচ্ছে, উপরের কোডে প্রতিটি কেসের পর যদি ব্রেক দেয়া না থাকত তাহলে কি হতো? এর উত্তর হলো,

কেসের ভেতরে যদি ব্রেক না থাকত তাহলে প্রোগ্রাম সংশ্লিষ্ট কেসের কাজ সম্পাদন করত এবং তারপর পরের কেসগুলোর কাজও সম্পাদন করত। এখানে কেস ১০-এ কিছু লেখা নেই। সুতরাং প্রোগ্রাম যদি কেস ১০-এ প্রবেশ করে, তাহলে তা কোনো কমান্ড পাবে না। তাই প্রোগ্রাম পরবর্তী কেসে প্রবেশ করবে। প্রতিটি সুইচ কেসেই একটি করে ডিফল্ট কেস দেয়া থাকে, উপরের কোডটিতেও দেয়া আছে। এর কারণ, যদি কখনো এমন কোনো অবস্থার সৃষ্টি হয় যে গ্রেডের মান ৭ থেকে ১০-এর মধ্যে নেই অর্থাৎ তা কোনো কেসেই পড়ছে না, তখন প্রোগ্রাম ডিফল্ট কেসে প্রবেশ করবে। আরেকটি জিনিস সবসময় খেয়াল রাখতে হবে- যেকোনো কেসের ইনডেক্সিংয়ের জন্য (অর্থাৎ কেস ১০, কেস ৯, কেস ৮ ইত্যাদি নামকরণ) সবসময় কোন মান ব্যবহার করতে হবে। যেমন ৯, ১০, ৮ ইত্যাদি মান, বিভিন্ন ক্যারেক্টার যেমন a, A, b, c, D, d ইত্যাদি ব্যবহার করা যাবে। আমরা জানি, প্রতিটি ক্যারেক্টারের একটি করে আক্ষি মান (Ascii value) আছে যা দিয়ে প্রোগ্রাম ওই ক্যারেক্টারকে শনাক্ত করতে পারে। তাই যদি কেস ‘a’, কেস ‘b’ ইত্যাদি নাম দেয়া হয় তাহলে প্রকৃতপক্ষে ওই ক্যারেক্টারের আক্ষি মানগুলো ব্যবহার হয়। তবে ইউজারের সেটি নিয়ে চিন্তা করার কোনো দরকার নেই, কারণ তিনি ওইসব ক্যারেক্টার দিয়েই প্রোগ্রামিং করতে পারবেন এবং তাতে কোনো সমস্যা হবে না। সুইচ কেসের মধ্যে নেস্টেড সুইচ কেসও ব্যবহার করা যায়।

While loop

ছোট একটি প্রোগ্রাম উদাহরণ হিসেবে দেয়া হলো এবং তার ভিত্তিতে এই লুপটির বিষদ আলোচনা করা হয়েছে।

```
int i=1;
while(1)
{
    printf("%d",i);
    i++;
    if(i==100)
        break;
}
```

While লুপের সিনটেক্স বেশ সহজ। while কিওয়ার্ডটি লিখে তার পাশে এক্সপ্রেশন দিতে

হয়। এক্সপ্রেশনের মান নন-জিরো হলে লুপটির অধীনে যে কোড ব্লকটি আছে তা সম্পাদন করা হয়। তবে কন্ডিশন বা এক্সপ্রেশনের মান যতক্ষণ সত্য হবে লুপটি ততক্ষণ চলতে থাকবে। উপরের উদাহরণটি দেখলে বোঝা যাবে লুপটি একটি অসীম লুপ। কারণ, এই লুপের এক্সপ্রেশন হিসেবে সরাসরি একটি নন-জিরো মান দেয়া হয়েছে এবং এই মানটির পরিবর্তনের কোনো নির্দেশ দেয়া নেই। অর্থাৎ যতবার এই এক্সপ্রেশনের মান চেক করা হবে, ততবার একটি নন-জিরো মান পাওয়া যাবে অর্থাৎ লুপের কন্ডিশন সত্য বলে বিবেচিত হবে। প্রথমবার প্রোগ্রাম যখন লুপটিতে আসবে তখন দেখবে যে কন্ডিশন সত্য, সুতরাং প্রোগ্রাম লুপে প্রবেশ করবে। তারপরে i-এর মান (১) প্রিন্ট করবে। এরপর i-এর মান ১ বাড়াবে। তারপর চেক করবে i-এর মান ১০০ কি না। যদি ১০০ হয়, তাহলে লুপটি ব্রেক করবে অর্থাৎ প্রোগ্রাম লুপ থেকে বের হবে। আর যদি ১০০ না হয় তাহলে প্রোগ্রাম আবার লুপের কন্ডিশনে ফিরে যাবে এবং দেখবে যে কন্ডিশন কি সত্য না মিথ্যা। এবারও কন্ডিশন সত্য হবে, কারণ এক্সপ্রেশন হিসেবে সরাসরি একটি নন-জিরো মান দেয়ায় লুপটি অসীম হয়ে গেছে। দ্বিতীয়বার প্রোগ্রাম লুপটিতে প্রবেশ করার পর আবার i-এর মান প্রিন্ট করবে। এবার কিন্তু i-এর মান হিসেবে ২ প্রিন্ট করবে, কারণ আগের লুপে i-এর মান ১ ইনক্রিমেন্টে অর্থাৎ বাড়ানো হয়েছে। এভাবে লুপটি চলতে থাকবে। অর্থাৎ এই প্রোগ্রামটি চালালে ১ থেকে ১০০ পর্যন্ত প্রিন্ট হবে। ১০০ প্রিন্ট করার পর যখন i-এর মান ১০১ হয়ে যাবে তখন প্রোগ্রাম if স্টেটমেন্টে প্রবেশ করবে এবং তখন if কন্ডিশন সত্য হওয়ার জন্য প্রোগ্রাম লুপটি ব্রেক করবে। উল্লেখ্য, প্রোগ্রাম যখন একবার কোনো লুপ বা কেস ব্রেক করে তখন তাকে নির্দেশ না দেয়া পর্যন্ত নিজে থেকে কখনও ওই লুপ/কেসে প্রবেশ করে না।

প্রোগ্রামিংয়ের সবচেয়ে গুরুত্বপূর্ণ বিষয়গুলোর একটি হলো কন্ডিশনাল অপারেশন। তাই একজন প্রোগ্রামারের কন্ডিশন নিয়ে কাজ করার ব্যাপারে খুব ভালো ধারণা থাকতে হবে **১৩**

ফিডব্যাক : wahid_cseast@yahoo.com