

প্রতিটি প্রোগ্রামিং ল্যাপ্সুয়েজেরই কিছু সাধারণ বৈশিষ্ট্য থাকে। তবে সি ল্যাপ্সুয়েজেই বেশিরভাগ বৈশিষ্ট্যের সূচনা হয়। এ ধরনের একটি বৈশিষ্ট্য হলো স্ট্রিং। শুধু সি ল্যাপ্সুয়েজেই নয়, আধুনিক সব ল্যাপ্সুয়েজে স্ট্রিংয়ের ব্যবহার দেখা যায়। স্ট্রিং কী এবং কীভাবে ব্যবহার করা হয়, তা নিয়ে বিস্তারিত আলোচনা করা হয়েছে এ লেখায়।

## স্ট্রিং কী?

সি ল্যাপ্সুয়েজে মূল চারটি ডাটা টাইপ আছে, যা নিয়ে আগে আলোচনা করা হয়েছে। স্ট্রিং হলো একটি বিশেষ ধরনের ডাটা টাইপ। যেকোনো প্রোগ্রামিং ল্যাপ্সুয়েজ দিয়ে যেকোনো প্রোগ্রাম ডেভেলপ করার সময় স্ট্রিং ব্যবহার করা হয়। স্ট্রিংয়ের সাথে ক্যারেক্টার টাইপের ডাটার

সীমাবদ্ধতা নেই। ইউজার যেকোনো ক্যারেক্টার যত ইচ্ছে ব্যবহার করতে পারেন। তবে প্রথম দিকে টার্বো সি-তে স্ট্রিংয়ের ধারণক্ষমতা ১০২৪ ক্যারেক্টার পর্যন্ত ছিল। আসলে কম্পাইলার অনুযায়ী স্ট্রিংয়ের ধারণক্ষমতা এত বেশি থাকে যে তা প্রায় অসীম বলা যায়। কিন্তু সমস্যা হয় ইউজার যখন একই স্ট্রিং একাধিক লাইনে ডিক্লেয়ার করতে চান। একটি লাইনে অনেক বড় স্ট্রিং ডিক্লেয়ার করতে গেলে অনেক সময় তা স্ক্রিনে নাও ধরতে পারে। সে ক্ষেত্রে একাধিক লাইনে স্ট্রিং ডিক্লেয়ার করার প্রয়োজন দেখা দেয়। কিন্তু সাধারণভাবে একাধিক লাইনে স্ট্রিং ডিক্লেয়ার করা যায় না। সে ক্ষেত্রে একটি ব্যাক স্ল্যাশ (\) ব্যবহার করতে হবে। যেমন :

```
“string in
```

লেখা হয়। এটি স্ট্রিং ব্যবহারের সময় ব্যবহার করা হয়। একটি স্ট্রিংয়ের একদম শেষে নাল ক্যারেক্টার প্রোগ্রাম নিজে থেকেই বসিয়ে দেয়।

প্রোগ্রামে printf() ফাংশন দিয়ে কোনো কিছু প্রিন্ট করতে হলে ফাংশনের ভেতরে যা লেখা হয় সেটাও একটি স্ট্রিং হিসেবেই প্রোগ্রাম গণনা করে। এখন এ স্ট্রিংটি প্রিন্ট করা শুরু করার পর কোথায় থামতে হবে সেটি প্রোগ্রাম নাল ক্যারেক্টারের মাধ্যমে বুঝতে পারে। নাল ক্যারেক্টার না থাকলে প্রিন্ট করা আর থামবে না। প্রোগ্রাম প্রিন্ট করেই যাবে। অর্থাৎ নির্দিষ্ট লাইন প্রিন্ট করার পর প্রোগ্রাম নাল ক্যারেক্টার না পায়, তাহলে গারবেজ ভ্যালু প্রিন্ট করতে থাকবে। তাই যেকোনো স্ট্রিংয়ের শেষ বোঝাতে স্ট্রিংয়ের শেষে নাল ক্যারেক্টার ব্যবহার করা হয়।

## স্ট্রিং ভেরিয়েবল

যদিও ইউজারের বোঝার সুবিধার্থে বলা হয়েছে স্ট্রিং একটি বিশেষ ধরনের ডাটা টাইপ। কিন্তু সি-তে স্ট্রিং নামে আসলে কোনো ডাটা টাইপ নেই। অন্যান্য প্রোগ্রামিং ল্যাপ্সুয়েজে স্ট্রিংকে আলাদা ডাটা টাইপ হিসেবে দেখা হলেও সি-তে আসলে স্ট্রিংকে ক্যারেক্টার অ্যারে দিয়ে তৈরি করা হয়। তাই সি-তে স্ট্রিং ভেরিয়েবল ডিক্লেয়ার করার মূল নিয়ম হলো প্রথমে একটি ক্যারেক্টার অ্যারে ডিক্লেয়ার করে তার মাঝে ক্যারেক্টারগুলো রাখা এবং সবশেষে একটি নাল ক্যারেক্টার বসিয়ে দেয়া। তাহলেই তা একটি স্ট্রিং হিসেবে ব্যবহার হবে। যেমন, ইউজার একটি স্ট্রিং ইনপুট নিতে চাচ্ছেন। স্ট্রিংটি হবে “What is your name?”। এখানে ডাবল কোটেশনের ভেতরে স্পেসসহ মোট ১৮টি ক্যারেক্টার আছে। তাই ইউজারকে ১৯টি ক্যারেক্টারের একটি অ্যারে ডিক্লেয়ার করতে হবে। অ্যারেতে ১৮টি ক্যারেক্টার বসানোর পর শেষ এলিমেন্টে নাল ক্যারেক্টার বসবে।

## char[] এবং স্ট্রিং ভেরিয়েবল

একটি ক্যারেক্টার অ্যারের বিভিন্ন এলিমেন্টের জন্য যেকোনো ক্যারেক্টার নির্ধারণ করা যায়। আবার ওপরে দেখানো হয়েছে স্ট্রিং হলো এমন একটি ভেরিয়েবল, যেখানে অনেকগুলো ক্যারেক্টারকে ডাবল কোটেশনের মাঝে পাশাপাশি রাখা হয়। তাই স্ট্রিং ডিক্লেয়ার করার জন্য সাধারণত এ ধরনের ক্যারেক্টার অ্যারে ব্যবহার করা হয়। যেমন :

```
char qstn[] = “What is your name?”
```

এখানে ক্যারেক্টার অ্যারে হিসেবে qstn ব্যবহার করা হয়েছে। লক্ষ করলে দেখা যাবে, এখানে একই সাথে অ্যারেটি ডিক্লেয়ার এবং ডিফাইন করা হয়েছে। আবার ডিক্লেয়ার করার সময় এর ইনডেক্স বা এলিমেন্ট সংখ্যা বলা হয়নি। এভাবে অ্যারে ডিক্লেয়ার করলে এর সাইজ প্রোগ্রাম নিজে থেকেই নির্ধারণ করে নেবে। যেমন, এখানে মোট ১৮টি ক্যারেক্টার আছে, নালসহ ১৯টি ক্যারেক্টারের প্রয়োজন হবে। তাই প্রোগ্রামে qstn-এর এলিমেন্ট নাম্বার নিজে থেকেই ১৯ করে নেবে। তবে ইউজার চাইলে নিজের পছন্দমতো এলিমেন্ট নাম্বার

# সহজ ভাষায় প্রোগ্রামিং সি/সি++

আহমদ ওয়াহিদ মাসুদ

অনেক মিল রয়েছে। স্ট্রিং আসলে অনেকগুলো ক্যারেক্টারের সমষ্টি। তাই একে মূল ডাটা টাইপের মাঝে রাখা হয়নি। ক্যারেক্টার টাইপের ভেরিয়েবলের মাঝে শুধু একটি ক্যারেক্টার রাখা যায়। কিন্তু একটি স্ট্রিংয়ের মাঝে একটি সম্পূর্ণ বাক্য রাখা সম্ভব। আবার একটি ক্যারেক্টারকেও স্ট্রিং হিসেবে ব্যবহার করা যায়। সুতরাং দেখা যাচ্ছে, ক্যারেক্টারকে স্ট্রিং হিসেবে ব্যবহার করা যায়, কিন্তু স্ট্রিংকে ক্যারেক্টার হিসেবে ব্যবহার করা যায় না।

## স্ট্রিং ডিক্লেয়ার

এখন দেখা যাক কীভাবে স্ট্রিং ডিক্লেয়ার করা যায়। সি-তে (অন্যান্য ল্যাপ্সুয়েজেও) ডাবল কোটেশনের মাঝে কোনো কিছু ডিক্লেয়ার করলেই তা স্ট্রিং হিসেবে ডিক্লেয়ার করা হয়। যেমন :

```
string x = “this is a string.”;
```

দেখা যাচ্ছে, এখানে ডাটা টাইপ হিসেবে string কিওয়ার্ড ব্যবহার করা হয়েছে। এটি অবশ্য বিভিন্ন ল্যাপ্সুয়েজ অনুযায়ী পরিবর্তন হতে পারে। আর স্ট্রিং ভেরিয়েবল হিসেবে বেছে নেয়া হয়েছে x-কে। এরপর ডাবল কোটেশনের মাঝে যাই লেখা হোক, তাই ভেরিয়েবল x-এর মাঝে স্ট্রিং হিসেবে স্টোর হয়ে যাবে। ওপরের উদাহরণে শুধু লেটার ব্যবহার করা হয়েছে। ইউজার ইচ্ছে করলে যেকোনো নিউমেরিক্যাল ক্যারেক্টার অথবা যেকোনো স্পেশাল ক্যারেক্টারও ব্যবহার করতে পারেন। এমনকি ইউজার যদি শুধু ডাবল কোটেশন অর্থাৎ x = “” লেখেন, তাহলে এটিও একটি স্ট্রিং হবে, তবে একে বলা হবে এমটি স্ট্রিং। সুতরাং অন্যান্য ভেরিয়েবলের যেমন বিভিন্ন ধরনের সীমাবদ্ধতা আছে (যেমন সাধারণ ইন্টিজারের জন্য সর্বোচ্চ মান ৩২৭৬৭), ইন্টিজারের সে ধরনের তেমন কোনো

```
two line”;
```

```
“string in\
```

```
two line.”;
```

প্রথম স্ট্রিংটি ডিক্লেয়ার করা হয়েছে দুটি লাইনে। এখানে প্রোগ্রাম এরর দেখাবে। পরের স্ট্রিংও ডিক্লেয়ার করা হয়েছে দুই লাইনে। কিন্তু প্রথম লাইনের শেষে একটি ব্যাক স্ল্যাশ দেয়া হয়েছে। এ ক্ষেত্রে প্রোগ্রাম কোনো এরর দেখাবে না। এভাবে একটি স্ট্রিং ডিক্লেয়ার করার সময় যত ইচ্ছে লাইন ব্যবহার করা যাবে। কিন্তু একটি করে ব্যাক স্ল্যাশ ব্যবহার করতে হবে।

## স্ট্রিং ডাটা এবং মেমরি

একটি প্রোগ্রামের সব ডাটাই মেমরিতে রাখা হয়। একেকটি ডাটা মেমরিতে একেক নিয়ম অনুসারে জায়গা দখল করে। যেমন ইন্টিজার ২ বাইট করে, ডাবল ৮ বাইট করে ইত্যাদি। কিন্তু একটি স্ট্রিংয়ের এ ধরনের কোনো নির্দিষ্ট সীমা নেই। স্ট্রিংয়ে যত ক্যারেক্টার থাকে তা মেমরিতে রাখার জন্য তত ক্যারেক্টার থেকে ১ বাইট জায়গা বেশি দখল করে। এ ১ বাইট জায়গা দিয়ে বোঝানো হয় স্ট্রিংটি শেষ হয়েছে। অন্যভাবে বলা যায় এ অতিরিক্ত ১ বাইটে নাল ক্যারেক্টার বসে।

## নাল ক্যারেক্টার (\0) এবং নাল

নাল কী তা নিয়ে আগেও আলোচনা করা হয়েছে। নাল মানে কিছুই নয়। কোনো পয়েন্টার দিয়ে যদি নালকে পয়েন্ট করা হয়, তার মানে হলো আসলে পয়েন্টারটি কোনো ভ্যালুকে পয়েন্ট করছে না। কোনো ভ্যালুর অনুপস্থিতি বোঝাতেই নাল ব্যবহার করা হয়। তবে সি-তে নাল এবং নাল ক্যারেক্টার এক নয়। শুধু নাল বলতে বোঝায় ডাটার অনুপস্থিতি। আর নাল ক্যারেক্টার একটি বিশেষ ক্যারেক্টার, যার আস্কি (ASCCI) ভ্যালু 0। এই ক্যারেক্টিকে ‘\0’ দিয়ে

ডিক্লেয়ার করে দিতে পারেন। কিন্তু তাহলে একটি একটি করে ক্যারেক্টার অ্যারেতে ইনপুট দিতে হবে।

### ‘A’ এবং ‘A’-এর মাঝে পার্থক্য

সিঙ্গেল কোটেশনের মাঝে কিছু দিলে তা প্রোগ্রাম ক্যারেক্টার হিসেবে গ্রহণ করবে। আর ডাবল কোটেশনের মাঝে কিছু দিলে তা হবে একটি স্ট্রিং। এমনকি একটি ক্যারেক্টারও যদি ডাবল কোটেশনের মাঝে রাখা হয় তাহলে ওই একটি ক্যারেক্টারই স্ট্রিং হিসেবে গণনা করা হবে। সুতরাং প্রোগ্রামে লিখতে হবে :

```
char ch='a';
char ch[]="a";
```

প্রথম ডিক্লেয়ারেশনের সময় মেমরিতে একটি সেলে A ক্যারেক্টারটি রাখা হবে। আর দ্বিতীয় ডিক্লেয়ারেশনের সময় মেমরির সেলে প্রথমে A ক্যারেক্টার রেখে তারপর একটি নাল ক্যারেক্টার রাখা হবে। এমনকি শুধু দুটি ডাবল কোটেশন দিয়েও একটি স্ট্রিং ডিক্লেয়ার করা যায়। সে ক্ষেত্রে স্ট্রিংটিতে শুধু একটি নাল থাকবে। এবার আরেকটি উদাহরণ দেয়া হলো :

```
char ch[]="";
char ch[]=NULL;
```

ওপরের দুটি ডিক্লেয়ারেশন এক নয়। প্রথমটিতে কোনো ডাটা নেই। h« মানে শুধু একটি \0 থাকবে। লক্ষণীয়, একটি নাল ক্যারেক্টার থাকা মানেও কিন্তু স্ট্রিংটির একটি ভ্যালু আছে। কিন্তু ডিক্লেয়ারের সময় যদি ভ্যালু হিসেবে NULL দেয়া হয়, তার মানে স্ট্রিংটির কোনো ভ্যালুই নেই। এ কারণেই বলা হয়েছে নাল এবং নাল ক্যারেক্টার এক নয়।

### স্ট্রিং ইনপুট এবং আউটপুট

স্ট্রিং ইনপুট এবং আউটপুট নেয়ার জন্য সাধারণত দুই ধরনের পদ্ধতি ব্যবহার করা হয়। একটি ফরম্যাট স্পেসিফায়ার ব্যবহার করে ইনপুট নেয়া, আরেকটি ব্যবহার না করে। প্রথমে কীভাবে ফরম্যাট স্পেসিফায়ার ব্যবহার করে ইনপুট নেয়া হয় তা দেখানো হবে।

```
char ch[50]="";
scanf("%s",&ch);
```

ইউজার ইচ্ছে করলে ফরম্যাট স্পেসিফায়ার ছাড়াও ইনপুট নিতে পারেন। যেমন :

```
char ch[50]="";
gets(ch);
```

এখানে gets() নামের যে ফাংশনটি ব্যবহার করা হয়েছে তা শুধু স্ট্রিং ইনপুটের জন্যই ব্যবহার করা হয়। সাধারণত দুটি পদ্ধতিই ঠিকমতো কাজ করে। কিন্তু প্রোগ্রাম যদি অনেক বড় হয় এবং বিশেষত একটি লুপের মাঝে যদি স্ট্রিং ইনপুট নিয়ে লুপটিকে বারবার চালানো হয়, তাহলে gets() ফাংশন ব্যবহার করলে অনেক সময় ভুল ইনপুট যেতে পারে। এর কারণ, একটি স্ট্রিংয়ের ইনপুট নেয়া তখনই শেষ হয় যখন এন্টার বাটনটি প্রেস করা হয়। gets() ফাংশন এভাবেই বানানো হয়েছে। যখন এন্টার প্রেস করা হবে তখন যেনো তা ইনপুট নেয়া বন্ধ করে দেবে এবং সবশেষে একটি নাল ক্যারেক্টার বসিয়ে দেবে। আসলে স্ট্রিং ইনপুট নেয়ার পদ্ধতি হলো ইউজার যখন একটি স্ট্রিং টাইপ করেন তখন প্রতিটি বাটন প্রেস করার সাথে সাথে সেই সিম্বলটি মেমরির স্ট্যাকে বসে যায়। পরে তা স্ট্রিংয়ের জন্য ডিক্লেয়ার করা অ্যারেতে পাঠানো হয়। এখন ইউজার একটি লাইন টাইপ করে এন্টার বাটন প্রেস করলেন। ফলে যা হবে তা হলো, প্রথমে ইনপুট নেয়া বন্ধ হবে। তারপর এন্টারের সিম্বলের (এন্টার বাটনের সিম্বল \n এবং আক্ষি মান ১০) আগ পর্যন্ত অ্যারেতে বসবে। এরপর লুপের অন্যান্য কাজ সম্পন্ন হয়ে লুপ শেষ হবে। পরবর্তী লুপে যখন প্রোগ্রাম আসবে তখন আবার gets() ফাংশন ইনপুট নেয়ার জন্য কাজ করবে। কিন্তু এর আগের \n এখনও স্ট্যাকে রয়ে গেছে। তাই প্রোগ্রাম মনে করবে আরেকবার এন্টার প্রেস করা হয়েছে। এভাবে লুপের ভেতরে স্ট্রিং ইনপুট নেয়ার সময় সমস্যা তৈরি হতে পারে। এর জন্য ফরম্যাট scanf() ফাংশন ব্যবহার করা যেতে পারে। তাহলে এ ধরনের সমস্যা হওয়ার সম্ভাবনা থাকে না।

আমাদের সবারই জানা, টাইপ করার সময় এন্টার বাটন প্রেস করলে নতুন লাইনে কার্সর চলে যায়। এটি একেবারেই সাধারণ একটি অপারেশন এবং এর জন্য কোনো ধরনের জটিলতার মশোমুখি হতে হয় না। কিন্তু স্ট্রিং নিয়ে কাজ করার সময় প্রোগ্রামারকে এ বিষয়টি নিয়ে সতর্ক থাকতে হয়। কারণ ইউজার যখন এন্টার বাটন প্রেস করেন, তখন আসলে একই সাথে দুটি কাজ সম্পন্ন হয়। প্রথমে টাইপিংয়ের কার্সরটি যেখানে আছে ঠিক তার নিচের স্থানে চলে যায়। এর নাম নিউ লাইন এবং এর সিম্বল \n। আর পরের কাজ হলো- কার্সর যেখানে যে লাইনে আছে, সেখান থেকে সে লাইনের শুরুতে চলে যাওয়া। যার নাম ক্যারিয়েজ রিটার্ন এবং এর সিম্বল \r। তাহলে দেখা যাচ্ছে এন্টার প্রেস করলে আসলে \rn সিম্বল কাজ করে।

এখন ধরা যাক, প্রোগ্রামার একটি স্ট্রিংয়ের ওপর কন্ডিশন নিয়ে প্রোগ্রাম লিখছেন। কন্ডিশনে যদি এ ধরনের থাকে যে ইউজার টাইপ করতে করতে এন্টার দিলে কোনো বিশেষ কাজ হবে, তাহলে এই ক্যারিয়েজ রিটার্ন সিম্বলের দিকে খেয়াল রাখতে হবে। কারণ সেখানে কন্ডিশন হিসেবে শুধু \n চেক করলেই হবে না, \r-কেও চেক করতে হবে। তবে কম্পাইলার ভিত্তিতে নিউ লাইন এবং ক্যারিয়েজ রিটার্নকে একসাথে বা আলাদাভাবে লেখা যায়।

স্ট্রিং ইনপুট নেয়া যত কঠিন, আউটপুট দেয়া ততটাই সহজ। সাধারণভাবে %s ব্যবহার করে printf() ফাংশন ব্যবহার করলেই স্ট্রিং প্রিন্ট হয়ে যাবে। অথবা ইউজার puts() ফাংশন ব্যবহার করতে পারেন। সে ক্ষেত্রে ফাংশনের আর্গুমেন্ট হিসেবে শুধু স্ট্রিং ভেরিয়েবলের নাম অর্থাৎ অ্যারের নাম দিয়ে দিলেই কাজ হয়ে যাবে।

স্ট্রিং নিয়ে কাজ করা প্রোগ্রামিংয়ের খুব গুরুত্বপূর্ণ অংশে পরে। কারণ, ইউজার কিবোর্ডে যাই টাইপ করুন না কেনো, আসলে তা স্ট্রিং হিসেবে মেমরিতে যায়। পরে তাকে বিভিন্নভাবে প্রসেস করে অন্যান্য ভেরিয়েবলে রূপান্তরিত করা হয় [১৬]

ফিডব্যাক : wahid\_cseast@yahoo.com