# সহজ ভাষায় প্রোগ্রামিং সি/সি++

আহমদ ওয়াহিদ মাসুদ

সব প্রোগ্রামিং ল্যাঙ্গুয়েজের জন্যই স্ট্রিং হলো একটি অবিচ্ছেদ্য অংশ। বিভিন্ন ল্যাঙ্গুয়েজে যেকোনো ইনপুট সাধারণত স্ট্রিং হিসেবেই নেয়া হয়। এরপর তাকে নির্দিষ্ট ডাটা টাইপে কনভার্ট করা হয়। যেকোনো কিছু প্রিন্ট করতে হলে তা স্ট্রিং হিসেবে প্রিন্ট হয়। অর্থাৎ ইউজার প্রোগ্রামিংয়ে যাই করুক না কেনো, তা কোনো না কোনো সময়ে স্ট্রিংয়ে কনভার্ট হয়। এ লেখায় স্ট্রিং সম্পর্কিত বিভিন্ন কাজ নিয়ে আলোচনা করা হয়েছে।

## এক স্ট্রিং ভেরিয়েবলের ডাটা অন্য স্ট্রিংয়ের জন্য নির্ধারণ করা

প্রোগ্রামে অনেক সময় একটি স্ট্রিং ভেরিয়েবলের ডাটা তথা স্ট্রিংকে অন্য স্ট্রিং ভেরিয়েবলের জন্য নির্ধারণ করার দরকার হয়। সি ল্যাঙ্গুয়েজে এ কাজের জন্য strcpy() নামে একটি লাইব্রেরি ফাংশন আছে, যার প্রটোটাইপ stdio.h ফাইলে বর্ণিত আছে। এটিকে হেডার ফাইলে নিচের মতো করে ডিক্লেয়ার করা হয়েছে।

```
prototype : char *strcpy(* dest, const char *src)
```

এখানে যে স্ট্রিং ভেরিয়েবলের ডাটা কপি করা হবে তা src-এর জন্য এবং যে ভেরিয়েবলে কপি করা হবে তা dest-এ পাঠাতে হয়। উদাহরণস্বরূপ :

```
char n1[30]=”rahimn karim”;
char n2[30]=”“;
strcpy(n2,n1);
```

উপরের প্রোগ্রামে n1 ও n2 নামে দুটি স্ট্রিং ভেরিয়েবল ডিক্লেয়ার করা হয়েছে, যার প্রথমটিতে কিছু ডাটা অ্যাসাইন করা হয়েছে ও দ্বিতীয়টি খালি রয়েছে। পরে স্ট্রিং কপি করার ফাংশনটি ব্যবহার করে একটি স্ট্রিংয়ের ডাটা অন্য স্ট্রিংয়ে কপি করা হয়েছে। এখানে খেয়াল করলে দেখা যাবে, ফাংশনের ভেতরে n2-এর পর n1 লেখা হয়েছে, যা প্রটোটাইপের ফরম্যাট অনুযায়ী কাজ করবে।

## একটি স্ট্রিংয়ে মোট ক্যারেক্টার সংখ্যা বের করা

কোনো প্রোগ্রামে স্ট্রিংয়ের অনেক ধরনের কাজের মাঝে একটি হলো স্ট্রিংয়ে মোট কতগুলো ক্যারেক্টার আছে তা বের করা। যারা প্রোগ্রামিং কন্টেস্ট করেন তাদের জন্য এটি খুবই সাধারণ একটি সমস্যা। আর এ ধরনের সমস্যার বাস্তব উদাহরণ হলো, কোনো ওয়ার্ড প্রসেসর যেমন মাইক্রোসফট ওয়ার্ডে যখন কোনো টেক্সট ফাইল ওপেন করা হয়, তখন একদম নিচে ওই ফাইলের মোট ক্যারেক্টার সংখ্যা দেখায়। শুধু তাই নয়, ইউজার যদি একটি নির্দিষ্ট প্যারা সিলেক্ট করে, তাহলে সিলেক্ট করা প্যারায় কতগুলো ক্যারেক্টার আছে তাও দেখানো হয়। যাই হোক, এ ধরনের সমস্যা সমাধানের অনেক উপায় আছে। তবে এখানে সবচেয়ে সহজ উপায়টি দেখানো হয়েছে। সি-তে strlen() নামে একটি লাইব্রেরি ফাংশন আছে। এর কাজ হলো কোনো স্ট্রিংয়ের মোট লেংথ বের করা। এটি কীভাবে ব্যবহার করা হয় ও কীভাবে এটি ব্যাবহার করে স্ট্রিংয়ের মোট ক্যারেক্টার সংখ্যা বের করা যায়, তা নিচে একটি ছোট উদাহরণের মাধ্যমে দেখানো হলো।

```
char _address[]=”141,142 love road, tejgaon”
int length=strlen(_address);
```

প্রথমে একটি স্ট্রিং ভেরিয়েবল ডিক্লেয়ার করে তাতে ভ্যালু হিসেবে একটি অ্যাড্রেস রাখা হয়েছে। ওই স্ট্রিংয়ে কতগুলো ক্যারেক্টার আছে। তা গণনা করার জন্য strlen() ফাংশনটি ব্যবহার করা হয়েছে। মনে রাখতে হবে strlen() ফাংশন সবসময় প্যারামিটার হিসেবে একটি স্ট্রিং নেয় এবং সবসময় একটি ইন্টিজার ভ্যালু রিটার্ন করে, এই ভ্যালুটি হলো ওই স্ট্রিংয়ের মোট ক্যারেক্টার সংখ্যা। তাই উপরের অ্যাড্রেসের মোট ক্যারেক্টার সংখ্যা রাখার জন্য length নামে একটি ইন্টিজার ডিক্লেয়ার করা হয়েছে এবং এর ভ্যালু হিসেবে strlen() ফাংশনের রিটার্ন করা ভ্যালু নির্ধারণ করা হয়েছে। তাহলে length ভেরিয়েবলে যে ভ্যালু পাওয়া যাবে তা-ই হলো অ্যাড্রেসের মোট ক্যারেক্টার সংখ্যা। এভাবে লাইব্রেরি ফাংশন ব্যবহার করে যেকোনো স্ট্রিংয়ের মোট ক্যারেক্টার সংখ্যা বের করা যায়। তবে আগেই বলা হয়েছে, এটি সবচেয়ে সহজ উপায়। তাই এভাবে গণনা করলে স্ট্রিংয়ের সব ক্যারেক্টারকেই বিবেচনা করা হবে। অর্থাৎ একটি স্ট্রিংয়ে যদি অনেক স্পেস থাকে, তাহলে তাকেও ক্যারেক্টার হিসেবে দেখা হবে। এখন ইউজার যদি চায় যে স্পেসকে গণনা করা হবে না, অথবা শুধু অক্ষরকে ধরা হবে কোনো সংখ্যাকে বিবেচনা করা হবে না, তখন এভাবে ক্যারেক্টার গণনা করলে হবে না। সেজন্য অন্য উপায়ে ক্যারেক্টার সংখ্যা বের করতে হবে। যার একটি উপায় হলো, কোনো স্ট্রিংয়ের সব ক্যারেক্টারের মাঝে একটি একটি করে ক্যারেক্টার নিয়ে প্রথমে দেখা হবে সেটি গণনা করা যায় কি না। এরপর তাকে বিবেচনা করা হবে। একটি ছোট উদাহরণ নিচে দেয়া হলো।

```
char _address[]=”141,142 love road, tejgaon”
int count=0;
char c=’’;
for(int i=0;i<strlen(_address);i++)
{
    c=_address[i];
if((c>=65&&c<=90)||(c>=97&&c<=122))
    count++;
}
```

এখানে স্ট্রিং হিসেবে আগের উদাহরণের অ্যাড্রেসকে ব্যবহার করা হয়েছে। কিন্তু আগের উদাহরণে সব ক্যারেক্টারকে গণনা করা হলেও এবারে শুধু A-Z এবং a-z কে গণনা করা হবে, অন্য কোনো ক্যারেক্টার থাকলে তাকে ক্যারেক্টার হিসেবে বিবেচনা করা হবে না। প্রথমে কাউন্ট নামে একটি ইন্টিজার ডিক্লেয়ার করা হয়েছে ক্যারেক্টার গণনা করার জন্য। তারপর সি নামে একটি টেম্পরারি ক্যারেক্টার ভেরিয়েবল ডিক্লেয়ার করা হয়েছে। এবার একটি ফর লুপে অ্যাড্রেসের সব ক্যারেক্টার চেক করা হয়েছে। এখানে ফর লুপের কন্ডিশন হলো strlen(_address); অর্থাৎ অ্যাড্রেসে যতগুলো ক্যারেক্টার আছে ফর লুপটি ততবার চলবে। তারপর প্রতিবার অ্যাড্রেস স্ট্রিংয়ের ক্যারেক্টারকে সি ভেরিয়েবলের মাঝে নিয়ে তা চেক করা হয়েছে। চেকিং if-else দিয়ে করা হয়েছে এবং এখানে কন্ডিশন হিসেবে সরাসরি আস্কি ভ্যালু ব্যবহার হয়েছে। আমরা জানি A-Z এর আস্কি ভ্যালু হলো ৬৫-৯০ এবং a-z এর আস্কি ভ্যালু হলো ৯৭-১২২। তাই ক্যারেক্টার সি-এর আস্কি ভ্যালু যদি কন্ডিশনের মাঝে পরে শুধু তাহলেই কাউন্ট ভেরিয়েবলের মান ১ করে বাড়বে। আর ফর লুপ শেষে কাউন্ট ভেরিয়েবলের মানই হলো মোট ক্যারেক্টার সংখ্যা। এটি হলো ক্যারেক্টার গণনার একটি কাস্টম কন্ডিশন। ইউজার যদি চান যে সব নিউমেরিক অক্ষরকেও ক্যারেক্টার হিসেবে কাউন্ট করতে হবে, তাহলে if-else-এর কন্ডিশন সেভাবে ঠিক করে দিতে হবে। এখানে যদিও আস্কি ভ্যালু দিয়ে কন্ডিশন চেক করা হয়েছে। ইউজার চাইলে সরাসরি ক্যারেক্টার দিয়েও চেক করতে পারেন। সে ক্ষেত্রে কন্ডিশন এরকম করে লিখতে হবে : if((c>=’a’&&c<=’z’)||(c>=’A’&&c<=’Z’)। ইউজারের বোঝার সুবিধার্থে প্রথমে আস্কি ভ্যালু দিয়ে কন্ডিশন দেয়া হয়েছে। কারণ প্রথমেই ক্যারেক্টার দিয়ে কন্ডিশন দেয়া হলে মনে প্রশ্ন জাগতে পারত যে >=a এবং <=z দিয়ে কী বোঝানো হচ্ছে। আসলে আমরা জানি সব ক্যারেক্টারকে প্রোগ্রাম আস্কি ভ্যালু দিয়ে চেনে। তাই ইউজার যদি সরাসরি ক্যারেক্টার দেন তাহলেও প্রোগ্রাম তাকে আসলে আস্কি ভ্যালুতে কনভার্ট করে নেবে। আর সরাসরি আস্কি ভ্যালু দিয়ে সেখানে কনভার্টের কোনো প্রশ্ন নেই। তবে খেয়াল রাখতে হবে, সিঙ্গেল কোটের (‘’) মাঝে ক্যারেক্টার লিখতে হবে। তাই লক্ষ করলে দেখা যাবে উপরের কন্ডিশনে ক্যারেক্টারকে সিঙ্গেল কোটের মাঝে লেখা হয়েছে, কিন্তু আস্কি ভ্যালুকে কোনো কোটের মাঝে লেখা হয়নি। কারণ সিঙ্গেল কোটেশনের মাঝে যাই লেখা হোক না কেনো তা ক্যারেক্টার হিসেবে বিবেচনা করা হবে, আর ডাবল কোটের মাঝে লিখলে তাকে স্ট্রিং হিসেবে দেখা হবে। সাধারণ এনকোডিং মেথডে ২৫৫টি ক্যারেক্টার আছে। অর্থাৎ তাদের আস্কি ভ্যালু 0-255। এ সাধারণ ক্যারেক্টারের মাঝে ইংরেজি লেটারসহ নিউমেরিক ▶

সংখ্যাও আছে। ইউজার চাইলে এ আস্কি ভ্যালুগুলো ইন্টারনেট থেকে দেখতে পারেন। তবে ইংরেজি ল্যাঙ্গুয়েজ ছাড়া যদি অন্য কোনো ল্যাঙ্গুয়েজ ব্যবহার করা হয়, সে ক্ষেত্রে আর সাধারণ এনকোডিং থাকে না। তখন সেটি ইউনিকোডে এনকোড করা হয় যেখানে আস্কি ক্যারেক্টার 0-232টি।

উপরের উদাহরণগুলোতে স্ট্রিং হিসেবে অ্যারে ব্যবহার করা হয়েছে। ইচ্ছে করলে এখানে পয়েন্টারও ব্যবহার করা যেত। কিন্তু এখানে অ্যারের ব্যবহার ঠিকমতো না করলে প্রোগ্রামের লজিক ঠিক থাকলেও ফলাফল ঠিকমতো আসবে না। তাই ইউজারের সুবিধার্থে এখানে অ্যারের যে নিয়মগুলো নিয়ে ভুল হওয়ার সম্ভাবনা আছে, তা তুলে ধরা হলো। প্রথমে অ্যারে নিয়ে সংক্ষিপ্ত বর্ণনা দেয়া হলো।

ভেরিয়েবল কী, তা ডিক্লেয়ার করলে কী হয় এবং তা কীভাবে কাজ করে তা আমরা জানি। অ্যারের মূল ধারণা নতুন কিছুই নয়। অ্যারে হলো অনেকগুলো ভেরিয়েবল একসাথে ডিক্লেয়ার করার একটি পদ্ধতি। ধরুন, কোনো প্রোগ্রামে একইসাথে পাঁচটি ভেরিয়েবল ডিক্লেয়ার করার প্রয়োজন হলো। তাহলে ইউজার সাধারণ নিয়মে পাঁচটি ভেরিয়েবল ডিক্লেয়ার করতে পারেন। এজন্য পাঁচটি স্টেটমেন্ট লেখার প্রয়োজন হবে। কিন্তু অ্যারে ব্যবহার করে পাঁচটি ভেরিয়েবল একই সাথে অর্থাৎ একটি স্টেটমেন্ট দিয়েই ডিক্লেয়ার করা সম্ভব। মাত্র পাঁচটি ভেরিয়েবলের ক্ষেত্রে হয়তো এটি তেমন গুরুত্বপূর্ণ বিষয় নয়, কিন্তু অনেক বড় বড় প্রোগ্রামে একই সাথে যখন ১০০ বা ১০০০ ভেরিয়েবল ডিক্লেয়ার করার প্রয়োজন হয়, তখন অ্যারে ব্যবহার করলে কোডিং অনেক সহজ হয়ে পড়ে। অ্যারে ডিক্লেয়ার করার সিনটেক্স : data_type array_name[array_size]। এখানে ডাটা টাইপ হলো অ্যারের ডাটা টাইপ অর্থাৎ কোন ধরনের ভেরিয়েবলের অ্যারে ডিক্লেয়ার করা হচ্ছে তা বলে দেয়া। অ্যারের নাম হলো যেকোনো ভেরিয়েবলের নাম। এখানে নতুন বিষয় হলো অ্যারে সাইজ। আমরা জানি অ্যারে হলো অনেকগুলো ভেরিয়েবলের সমষ্টি। সুতরাং কতগুলো ভেরিয়েবল নিয়ে অ্যারে ডিক্লেয়ার করা হচ্ছে সেটা বলতে হবে। সাইজ যত হবে ততগুলো ভেরিয়েবল নিয়ে অ্যারে গঠিত হবে। যেমন : int prime[10], valid[5]; ইত্যাদি। এখানে একই সাথে প্রাইম নামে দশটি, ভ্যালিড নামে পাঁচটি ইন্টেজার ডিক্লেয়ার করা হয়েছে।

অ্যারের এলিমেন্ট বলতে বোঝায় তার সাইজ কত। যদি অ্যারের সাইজ ৫ হয়, তার মানে অ্যারের এলিমেন্ট সংখ্যা হলো ৫। অর্থাৎ পাঁচটি ভেরিয়েবল নিয়ে অ্যারেটি গঠিত। অ্যারে ডিক্লেয়ার করার সময় সাইজ হিসেবে যে শুধু সংখ্যাই ব্যবহার করা যাবে, এমন কোনো কথা নেই। ইউজার ইচ্ছে করলে কোনো ভেরিয়েবল অথবা কনস্ট্যান্টও ব্যবহার করতে পারেন। বড় প্রোগ্রামের ক্ষেত্রে যখন একই সাইজবিশিষ্ট একাধিক অ্যারে ব্যবহারের প্রয়োজন হয়, তখন কনস্ট্যান্ট ব্যবহার করে অ্যারে ডিক্লেয়ার করাই ভালো। কারণ যদি প্রোগ্রামের কোনো ভুল হয় অথবা কোনো কিছু পরিবর্তনের জন্য অ্যারেগুলোর সাইজ পরিবর্তনের দরকার হয়, তাহলে কষ্ট করে প্রতিটি অ্যারের ডিক্লারেশনে এডিট না করে শুধু কনস্ট্যান্টের মান পরিবর্তন করলেই হয়। যেমন :

```
....
#define array_size 5;
....
int a1[array_size];
int a2[array_size];
...
```

অ্যারের প্রতিটি এলিমেন্ট আলাদাভাবে নিয়ন্ত্রণ করা যায়। যেমন : int piyal[5]; এ অ্যারেটি ডিক্লেয়ার করা হলে পাঁচটি ইন্টিজার ভেরিয়েবল পর পর ডিক্লেয়ার হবে। যাদের সাধারণ নাম থাকবে piyal। কিন্তু চাইলে তাদেরকে আলাদা আলাদাভাবে ব্যবহার করা যাবে। উপরের অ্যাড্রেস অ্যারেকেও একই নিয়ম অনুসারে আলাদা আলাদাভাবে ব্যবহার করা হয়েছে। আমরা জানি অ্যারে ডিক্লেয়ার করার সময় বন্ধনীর মাঝে অ্যারের সাইজ উল্লেখ করে দিতে হয়। এই বন্ধনী অনেকভাবে ব্যবহার করা যেতে পারে। মনে রাখতে হবে অ্যারে যখন ডিক্লেয়ার করা হয় শুধু তখনই বন্ধনীটি অ্যারের সাইজ নির্ধারণের জন্য ব্যবহার হয়। অন্য সময় অ্যারের ইন্ডেক্সিংয়ের জন্য এই বন্ধনী ব্যবহার করা হয়। ইন্ডেক্সিং বলতে বোঝায় প্রতিটি এলিমেন্টকে আলাদাভাবে শনাক্তকরণ। উপরে piyal[5] ভেরিয়েবলটি ডিক্লেয়ার করা হলে মেমরিতে পরপর piyal[0], piyal[1], piyal[2], piyal[3], piyal[4] নামে পাঁচটি ভেরিয়েবল অ্যালোকেট করা হবে। এখন বন্ধনী ব্যবহার করে এ পাঁচটি ভেরিয়েবলকে আলাদাভাবে নিয়ন্ত্রণ করা সম্ভব। যেমন : এখন যদি লেখা হয় piyal[0]=10; তাহলে অ্যারের প্রথম এলিমেন্টটির মান ১০ নির্ধারিত হবে। piyal[0]=piyal[0]*piyal[1]; এই স্টেটমেন্টটি লিখলে প্রথম ও দ্বিতীয় এলিমেন্টের মান গুণ করে প্রথম এলিমেন্টে রাখা হবে। তবে এ ক্ষেত্রে দ্বিতীয় এলিমেন্টের মান যেহেতু গারবেজ, তাই গুণফলও গারবেজ আসবে।

অ্যারে ব্যবহারে সবচেয়ে বেশি যে ভুলটি হয় তা হলো সাইজ ও ইন্ডেক্সের সমস্যা। এটি সবসময় খেয়াল রাখতে হবে সাইজ ও ইন্ডেক্সের ব্যবহারের সিনটেক্স একই হলেও এরা ভিন্ন জিনিস। উভয়ই [] বন্ধনীর মাধ্যমের ব্যবহার হয়। কিন্তু অ্যারে সাইজ বলতে বোঝায় যে অ্যারেতে কতগুলো এলিমেন্ট থাকবে। আর ইন্ডেক্স দিয়ে অ্যারের এলিমেন্টগুলোকে অ্যাড্রেস করা হয়। প্রোগ্রামিং ল্যাঙ্গুয়েজে যেকোনো কিছুর অ্যাড্রেসিং ০ থেকে আরম্ভ হয়। তাই অ্যারের সাইজ ৫ হলেও পঞ্চম এলিমেন্টের ইন্ডেক্স হবে ৪। যেমন : int a[5]; এখানে পাঁচটি এলিমেন্টবিশিষ্ট একটি অ্যারে ডিক্লেয়ার করা হয়েছে। কিন্তু অ্যারেটির প্রথম এলিমেন্টের নাম a[0] ও শেষ এলিমেন্টের নাম a[4], a[5] নামে কোনো এলিমেন্ট নেই।

অ্যারের এলিমেন্টকে কোনো আইডেন্টিফায়ার দিয়েও ইন্ডেক্সিং করা যায়। তবে খেয়াল রাখতে হবে ইন্ডেক্সিং করার আগে আইডেন্টিফায়ারের মান যেনো নির্ধারণ করা হয় এবং সংশ্লিষ্ট মানের এলিমেন্ট যেনো অ্যারেতে উপস্থিত থাকে। যেমন : i=3; a[i]=30; এখানে অ্যারের ইন্ডেক্সিং করা হয়েছে একটি আইডেন্টিফায়ার দিয়ে। আর এ ক্ষেত্রে অ্যারের চতুর্থ এলিমেন্টের মান পরিবর্তন হচ্ছে। তবে এর মান ০ থেকে ৪-এর বাইরে হলে এরর দেখাবে। কারণ অ্যারেতে ০ থেকে ৫ পর্যন্ত ইন্ডেক্সের এলিমেন্ট আছে। এররটি আবার লজিক্যাল এরর হবে। অর্থাৎ সেটি কম্পাইলের সময় ধরা নাও পড়তে পারে। অ্যাড্রেস ভেরিয়েবলের ক্ষেত্রেও আইডেন্টিফায়ার হিসেবে ব্যবহার করা হয়েছে। কিন্তু সেখানে কখনও লজিক্যাল এরর দেখাবে না, কারণ এর মান 0 থেকে strlen(_address) পর্যন্ত।

স্ট্রিংয়ের ব্যবহার পরিধি অনেক বিস্তৃত বলে এর অনেক ফাংশন বানানো হয়েছে। এসব ফাংশন ব্যবহারের মাধ্যমে স্ট্রিং দিয়ে অনেক জটিল কাজ সহজে সমাধান করা যায়। কজ

*ফিডব্যাক : wahid_cseaust@yahoo.com*