সহজ ভাষায় প্রোগ্রামিং সি/সি++

আহমদ ওয়াহিদ মাসুদ

সহজে সমস্যা সমাধানের জন্য প্রোগ্রামিং ল্যাঙ্গুয়েজের সৃষ্টি। আর ব্যবহারের ওপর ভিত্তি করে প্রোগ্রামিং ল্যাঙ্গুয়েজের আবার প্রকারভেদ আছে। সি হলো মিড লেভেল ল্যাঙ্গুয়েজ। মিড লেভেল বলার কারণ এই নয় যে, এই ল্যাঙ্গুয়েজের ক্ষমতা কম। সরাসরি হার্ডওয়্যার এবং সফটওয়্যার দুই লেভেলেই কাজ করা যায় দেখে সি-কে মিড লেভেল বলা হয়। অর্থাৎ সি ল্যাঙ্গুয়েজে হাই লেভেল এবং লো লেভেল দুই ধরনের বৈশিষ্ট্যই আছে। এ লেখায় সম্পূর্ণ নতুন একটি বিষয় নিয়ে আলোচনা করা হয়েছে। এটি মূলত হাই লেভেল সম্পর্কিত।

অ্যারে

অ্যারে মূলত হাই লেভেল ল্যাঙ্গুয়েজের বৈশিষ্ট্য এবং এটি সি প্রোগ্রামিংয়ে অন্যতম প্রধান ফিচারগুলোর একটি। অ্যারে কি তা বোঝার জন্য একটু পেছনে ফিরে যেতে হবে। আমরা জানি, সি-তে ভেরিয়েবল ডিক্লেয়ার করা যায়। অ্যারের মূল ধারণা হলো অনেকগুলো ভেরিয়েবল একসাথে ডিক্লেয়ার করার একটি পদ্ধতি। ধরুন, কোনো প্রোগ্রামে একই সাথে ৫টি ভেরিয়েবল ডিক্লেয়ার করার প্রয়োজন। এক্ষেত্রে ইউজার সাধারণ নিয়মে ৫টি ভেরিয়েবল ডিক্লেয়ার করতে পারেন। এজন্য ৫টি স্টেটমেন্ট লেখার প্রয়োজন হবে। কিন্তু অ্যারে ব্যবহার করে ৫টি ভেরিয়েবল একই সাথে অর্থাৎ একটি স্টেটমেন্ট দিয়েই ডিক্লেয়ার করা সম্ভব। মাত্র ৫টি ভেরিয়েবলের ক্ষেত্রে হয়ত এটি তেমন গুরুত্বপূর্ণ বিষয় নয় কিন্তু অনেক বড় প্রোগ্রামে একই সাথে যখন ১০০ বা ১০০০ ভেরিয়েবল ডিক্লেয়ার করার প্রয়োজন হয়, তখন অ্যারে ব্যবহার করলে কোডিং অনেক সহজ হয়।

অ্যারে ডিক্লেয়ার

অ্যারে হলো কতগুলো ভেরিয়েবলের সমষ্টি। সুতরাং ভেরিয়েবল ডিক্লেয়ারের মতো করেই অ্যারে ডিক্লেয়ার করতে হয়। ভেরিয়েবল ডিক্লেয়ারের আগে যেমন ডাটা টাইপ লেখার দরকার, অ্যারের জন্যও তেমনি দরকার। অ্যারে ডিক্লেয়ার করার সিন্টেক্স : data_type array_name[array_size]। এখানে ডাটা টাইপ হলো অ্যারের ডাটা টাইপ অর্থাৎ কোনো ধরনের ভেরিয়েবলের অ্যারে ডিক্লেয়ার করা হচ্ছে, তা বলে দেয়া। অ্যারের নাম হলো যেকোনো ভেরিয়েবলের নাম। এখানে নতুন বিষয় হলো অ্যারে সাইজ। কতগুলো ভেরিয়েবল নিয়ে অ্যারে ডিক্লেয়ার করা হচ্ছে, সেটা বলতে হবে। এটিই হলো অ্যারে সাইজ। সাইজ যত হবে, ততগুলো ভেরিয়েবল নিয়ে অ্যারে গঠিত হবে। অ্যারে ও ভেরিয়েবল ডিক্লেয়ারের মাঝে মূল পার্থক্য হলো সাইজ। যেমন int prime[10], valid[5]; ইত্যাদি। এখানে একই সাথে প্রাইম নামে দশটি, ভ্যালিড নামে পাঁচটি ইন্টেজার ডিক্লেয়ার করা হয়েছে।

অ্যারের এলিমেন্ট বলতে বুঝায় এর সাইজ কত। যদি অ্যারের সাইজ ৫ হয়, তাহলে অ্যারের এলিমেন্ট সংখ্যা হলো ৫। অর্থাৎ ৫টি ভেরিয়েবল নিয়ে অ্যারেটি গঠিত। এলিমেন্টের কথা আলাদাভাবে বলার কারণ এলিমেন্ট নিয়ে অনেক ধরনের কাজ করা যায়।

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 পড়ে তখন কনস্ট্যান্ট ব্যবহার করে অ্যারে ডিক্লেয়ার করাই ভালো। কারণ, যদি প্রোগ্রামের কোনো ভুল হয় অথবা কোনো কিছু পরিবর্তনের জন্য অ্যারেগুলোর সাইজ পরিবর্তনের দরকার পড়ে, 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];
…
```

এখানে প্রোগ্রামের শুরুতেই array_size নামে একটি কনস্ট্যান্ট ডিক্লেয়ার করা হয়েছে। প্রতিবার অ্যারে ডিক্লেয়ার করার সময় সাইজ হিসেবে এই কনস্ট্যান্ট ব্যবহার করা হয়েছে। এখন যদি তেমন কোনো প্রয়োজন হয় যে অ্যারেগুলোকে ৫টি নয় বরং ৬টি এলিমেন্ট সহকারে ডিক্লেয়ার করতে হবে, তাহলে কষ্ট করে প্রতিটি ডিক্লারেশনের লাইন এডিট না করে শুধু array_size-এর মান ৬ করে দিলেই হবে। একইভাবে কনস্ট্যান্টের বদলে কোনো ভেরিয়েবল ব্যবহার করেও ডিক্লেয়ার করা সম্ভব। শুধু খেয়াল রাখতে হবে, যে ভেরিয়েবল ব্যবহার করা হচ্ছে তার মান যেনো আগে থেকে নির্ধারণ করা থাকে। অন্যথায় এরর দেখাবে। টার্বো সি-তে ভেরিয়েবল দিয়ে অ্যারের সাইজ নির্ধারণ করতে গেলে অনেক সময় এরর দেখায়। তাই টার্বো সি-তে এভাবে ডিক্লেয়ার না করাই ভালো। তবে অন্যান্য ৩২ বিটের কম্পাইলার যেমন ভিজ্যুয়াল সি++ ইত্যাদি ব্যবহার করলে এরর দেয়ার সম্ভাবনা থাকে না।

মেমরিতে অ্যারের nVb

আমরা জানি, কোনো ভেরিয়েবল ডিক্লেয়ার করলে তা মেমরিতে নিয়মানুসারে কিছু নির্দিষ্ট পরিমাণে জায়গা দখল করে। যেমন ডাটা টাইপ, কম্পাইলার ইত্যাদি। এখন প্রশ্ন হলো অ্যারে কি সাধারণ ভেরিয়েবলের মতো মেমরিতে জায়গা দখল করে কি না। মেমরি নিয়ে আলোচনা করা হচ্ছে, কারণ অ্যারের বিভিন্ন এলিমেন্টের সাথে মেমরির বিশেষ সম্পর্ক আছে। সাধারণ ভেরিয়েবলের মতোই অ্যারের জন্য মেমরি নির্ধারণ করা হয়, তবে অ্যারের এলিমেন্ট যখন একের অধিক হয় তখন কিছু বিশেষ নিয়ম অনুসরণ করা হয়। যেমন প্রোগ্রামে ১০ এলিমেন্টবিশিষ্ট কোনো ইন্টিজার অ্যারে ডিক্লেয়ার করা হলে মেমরিতে পরপর ১০টি ইন্টিজার ভেরিয়েবল স্থান দখল করবে। মনে রাখতে হবে, অ্যারের এলিমেন্টগুলো একেকটি আলাদা ভেরিয়েবল। এগুলো সবসময় পরপর অবস্থান করে। মেমরির শুরু থেকে যেখানে ফাঁকা জায়গা পাবে, প্রোগ্রাম সেখানেই অ্যারের স্থান নির্ধারণ করবে। যদি মেমরির এক জায়গায় ৮ বাইট জায়গা পরপর খালি থাকে, তাহলে ১০ এলিমেন্টবিশিষ্ট অ্যারের সেখানে অ্যালোকেট করা হবে না। কারণ যে অ্যারেটি ডিক্লেয়ার করা হয়েছে তার এলিমেন্ট সংখ্যা ১০। তাই মেমরির শুরু থেকে সার্চ করে প্রথম যেখানে পরপর ১০ বাইট জায়গা ফাঁকা পাওয়া যাবে প্রোগ্রাম সেখানেই অ্যারে অ্যালোকেট করবে। সেটি একটি সাধারণ উদাহরণ মাত্র। ডাটা টাইপ এবং কম্পাইলারের ভিত্তিতে একক ভেরিয়েবলের জন্য নির্ধারিত জায়গা পরিবর্তন হয় এবং অ্যারে কত বাইট করে জায়গা দখল করবে তাও পরিবর্তন হয়। যদি ইন্টিজার টাইপের একটি অ্যারে ডিক্লেয়ার করা হয় যার এলিমেন্ট সংখ্যা ৫, তাহলে তা মেমরিতে পরপর ১০ বাইট জায়গা দখল করবে। কারণ ইন্টিজার ২ বাইট করে জায়গা দখল করে।

অ্যারের এলিমেন্ট ব্যবহার

অ্যারের এলিমেন্ট মেমরিতে কিভাবে থাকে তা জানা থাকলে এলিমেন্টের ব্যবহার সম্পর্কেও ভালোভাবে ধারণা পাওয়া যায়। অ্যারের একটি সুবিধা হলো, এটি একটি একক ভেরিয়েবলের মতো ডিক্লেয়ার করা হয়, কিন্তু এর প্রতিটা এলিমেন্ট আলাদাভাবে নিয়ন্ত্রণ করা যায়। মূলত এ কারণেই অ্যারের উৎপত্তি। যেমন int piyal[5]; এই অ্যারেটি ডিক্লেয়ার করা হলে ৫টি ইন্টিজার ভেরিয়েবল পরপর ডিক্লেয়ার হবে। যাদের সাধারণ নাম থাকবে piyal। কিন্তু চাইলে তাদেরকে আলাদাভাবে ব্যবহার করা যাবে। আমরা জানি, অ্যারে ডিক্লেয়ার করার সময় বন্ধনীর মাঝে অ্যারের সাইজ উল্লেখ করতে 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 প্রতিটা এলিমেন্টকে আলাদাভাবে শনাক্তকরণ। উপরে piyal[5] ভেরিয়েবলটি ডিক্লেয়ার করা হলে মেমরিতে পরপর piyal[0], piyal[1], piyal[2], piyal[3], piyal[4] নামের পাঁচটি ভেরিয়েবল অ্যালোকেট করা হবে। এখন বন্ধনী ব্যবহার করে এই 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 এক্ষেত্রে দ্বিতীয় এলিমেন্টের মান যেহেতু গারবেজ, তাই গুণফলও গারবেজ আসবে। অর্থাৎ ইন্ডেক্সিংয়ের মাধ্যমে প্রতিটি এলিমেন্টকে আলাদা ভেরিয়েবলের মতো ব্যবহার করা যায়।

অ্যারের এলিমেন্ট ব্যবহারের নিয়ম

অ্যারে ব্যবহারে সবচেয়ে বেশি যে ভুলটি হয়, তা হলো সাইজ ও ইন্ডেক্সের সমস্যা। এটি সবসময় খেয়াল রাখতে হবে যে, সাইজ ও ইন্ডেক্সের ব্যবহারের সিন্টেক্স একই হলেও এরা ভিন্ন জিনিস। উভয়ই [] বন্ধনীর মাধ্যমের ব্যবহার হয়। কিন্তু অ্যারে সাইজ বলতে বোঝায় 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] এবং শেষ এলিমেন্টের নাম a[4]। অর্থাৎ a[5] নামের কোনো এলিমেন্ট নেই। কারণ, যেহেতু প্রোগ্রামিংয়ে ইন্ডেক্সিং ০ থেকে শুরু হয় এবং ০ থেকে ৪ পর্যন্ত মোট ৫টি সংখ্যা আছে তাই সর্বশেষ এলিমেন্ট হলো a[4]।

একটি সাধারণ ভেরিয়েবল ব্যবহারের সব নিয়ম একটি একক এলিমেন্টের বেলায় সত্য। সাধারণ ভেরিয়েবল এবং একটি একক অ্যারে এলিমেন্টের মাঝে নামের পার্থক্য ছাড়া আর কোনো পার্থক্য নেই।

শুধু অ্যারের নামকে ভেরিয়েবল হিসেবে ব্যবহার করা যায় না। অর্থাৎ int a[5] অ্যারে ডিক্লেয়ার করা হলে শুধু a নামে কোনো ভেরিয়েবল থাকবে না। তবে ফাংশনের প্যারামিটার হিসেবে কোনো অ্যারে পাঠাতে হলে শুধু অ্যারের নাম ব্যবহার করতে হয়। তাও আবার লোকাল প্যারামিটারের বেলায় প্রযোজ্য।

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 এক্ষেত্রে অ্যারের চতুর্থ এলিমেন্টের মান পরিবর্তন হচ্ছে। তবে i 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। তাই ইন্ডেক্সিংয়ের ব্যাপারে সতর্ক থাকা দরকার।

অ্যারের মান নির্ধারণ

প্রোগ্রামে ভেরিয়েবলের মতো অ্যারেকেও ডাটা হিসেবে নিয়ে কাজ করা যায় এবং সে ডাটা বিভিন্নভাবে ব্যবহার করা যায়। অ্যারের মান নির্ধারণ বলতে বিভিন্ন এলিমেন্টের ডাটা নির্ধারণই বোঝায়। ধরা যাক, প্রোগ্রামে একটি অ্যারে ডিক্লেয়ার করা হলো int a[5]; নামে। এখন মেমরিতে ৫টি ইন্টিজার ভেরিয়েবল অ্যালোকেট হলো, কিন্তু এই ৫টি ভিন্ন ভিন্ন ভেরিয়েবলের মান নির্ধারণ অনেকভাবে করা যায়। প্রথমত সাধারণ ভেরিয়েবলের মান নির্ধারণের মতো করে এলিমেন্টগুলোর মান নির্ধারণ করা যায়। যেমন a[0]=10; a[1]=20; ইত্যাদি। এটি হলো অ্যাসাইন অপারেটরের সাহায্যে ডাটা নির্ধারণ করা। মান নির্ধারণের আরেকটি মাধ্যম হলো ডিক্লেয়ারেশনের সময় বন্ধনী ব্যবহার করে সরাসরি ডাটা অ্যাসাইন করা। যেমন int a[5]={10,20,30,40,50}; এখানে একটি স্টেটমেন্টের মাধ্যমে অ্যারের সব এলিমেন্টের মান একসাথে নির্ধারিত হলো। যদি অনেকগুলো এলিমেন্টবিশিষ্ট অ্যারে ব্যবহার করা হয়, তাহলে আলাদা স্টেটমেন্ট ব্যবহার করে মান নির্ধারণ করা কষ্টসাধ্য ব্যাপার। তখন এভাবে সব মান একসাথে ডিক্লেয়ার করা যায়, তবে বন্ধনীর মাধ্যমে মান নির্ধারণে কিছুটা সাবধানতা অবলম্বন করা উচিত। কারণ, বন্ধনীর ভেতরে যতগুলো মান দেয়া হবে অ্যারের ততগুলো এলিমেন্টের মান নির্ধারিত হবে। ধরা যাক, অ্যারের এলিমেন্ট আছে ৫টি কিন্তু বন্ধনীর ভেতরে ভুলে ৪ বা ৬টি মান দেয়া হলো। ৪টি মান দিলে অ্যারের প্রথম ৪টি এলিমেন্টের মান ঠিকই নির্ধারিত হবে, কিন্তু ৫ম এলিমেন্টটি অনির্ধারিত থেকে যাবে, অর্থাৎ গারবেজ ভ্যালু থাকবে। আর ৬টি মান দিলে অ্যারের ৫টি এলিমেন্টের মান ঠিকই নির্ধারিত হবে, কিন্তু একটি মান অতিরিক্ত থেকে যাবে বলে প্রোগ্রাম এরর দেখাবে। তাই অ্যারের এলিমেন্ট সংখ্যায় অল্প হলে এভাবে মান নির্ধারণ করা যেতে পারে, কিন্তু এলিমেন্ট সংখ্যায় বেশি হলে এ পদ্ধতি অবলম্বন না করাই ভালো।

বন্ধনীর মাধ্যমে অ্যারের মান নির্ধারণের আরও কিছু উদাহরণ দেয়া হলো। আমরা জানি, অ্যারের ডাটাটাইপ যেকোনো ভেরিয়েবলের অর্থাৎ যেকোনো বিল্টইন ডাটাটাইপ হতে পারে। কাস্টম ডাটাটাইপও হতে পারে, তবে সে বিষয়ে এখন আলোচনা করা হয়নি। ধরা যাক, প্রোগ্রামে একটি ক্যারেক্টার অ্যারে ডিক্লেয়ার করা হলো এবং তার মানগুলো বন্ধনীর মাধ্যমে দেয়া দরকার। অ্যারেতে ক্যারেক্টার হিসেবে মান দিতে হলে সিঙ্গেল কোটেশন দিতে হয়। যেমন char A[5]={‘a’,’e’,’i’,’o’,’’}; এখানে অ্যারের এলিমেন্টেগুলোর মান ডিক্লারেশনের সময়ই দেয়া হলো এবং এভাবে ক্যারেক্টার মান দেয়ার জন্য কোটেশন ব্যবহার করতে হবে। যদি স্ট্রিংয়ের অ্যারে হয় তাহলে ডাবল কোটেশন দিতে হবে। প্রাথমিকভাবে বলা যায় স্ট্রিং এক ধরনের ভেরিয়েবল, যেখানে ডাটা হিসেবে কোনো একক ক্যারেক্টার নয় বরং এক বা একাধিক ক্যারেক্টারের লাইন স্টোর করা হয়।

অ্যারের মান নির্ধারণের সবচেয়ে সহজ এবং নিরাপদ মাধ্যম হলো কোনো লুপ ব্যবহার করা। কারণ লুপ ব্যবহার করলে বন্ধনীর মতো ভুল হওয়ার সম্ভাবনা থাকে না, আবার অনেকগুলো স্টেটমেন্ট লেখার প্রয়োজনও পড়ে না। লুপ ব্যবহারে মান নির্ধারণের একটি উদাহরণ হলো :

```
....
int array[5]={0},i=0;
for(i=0;i<5;i++)
        scanf("%d",&array[i]);
....
```

এখানে প্রথমে array নামের ৫ এলিমেন্টবিশিষ্ট একটি অ্যারে ডিক্লেয়ার করা হলো। ডিক্লেয়ার করার সময় বন্ধনীর মাধ্যমে সব এলিমেন্টের মান ০ করে দেয়া হয়েছে, যাতে কোনো গারবেজ ভ্যালু না থাকে। এখানে লক্ষণীয়, অ্যারের সব এলিমেন্টের মান একই নির্ধারণ করার দরকার হলে বন্ধনীর মাঝে নির্দিষ্ট মানটি দিয়ে দিলেই হয়। এখানে যদি বন্ধনীর ভেতরে ২ দেয়া হতো তাহলে অ্যারের সব এলিমেন্টের মান শুরুতে ২ নির্ধারিত হতো। এরপর আরও একটি ভেরিয়েবল ডিক্লেয়ার করা হয়েছে লুপ চালানোর জন্য। এখানে অ্যারের ইন্ডেক্সিংয়ের জন্য ভেরিয়েবল i ব্যবহার করা হয়েছে। লক্ষ করলে দেখা যাবে, লুপের ভেতরে i-এর মান ১ করে বাড়ছে, যার ফলে প্রতিবার অ্যারের ইন্ডেক্সের মানও ১ করে বাড়ছে। এভাবে লুপ ব্যবহার করে অ্যারের ইনপুট নিলে ইন্ডেক্সিংয়ের জন্য ভেরিয়েবল ব্যবহার করতে হয়, তা না হলে ইনপুট নেয়া যায় না।

অ্যারে প্রোগ্রামিংয়ের অত্যন্ত গুরুত্বপূর্ণ একটি অংশ। অ্যারে ব্যবহারের মাধ্যমে শত শত ভেরিয়েবলের কাজ নিমেষেই করা সম্ভব। প্রোগ্রামিংয়ের সবক্ষেত্রেই অ্যারের ব্যবহার লক্ষ করা যায়। ডাটাবেজের কাজ পুরোটাই অ্যারে দিয়ে করা হয়। কারণ ডাটাবেজে অসংখ্য ভেরিয়েবলের দরকার হয় যেগুলো এককভাবে ডিক্লেয়ার করতে গেলে কোডিং করা অসম্ভব হয়ে যাবে। এমনকি গ্রাফিক্সের কাজেও অ্যারের বহুল ব্যবহার হয়। গ্রাফিক্সের কাজেও প্রতিটা পিক্সেলের জন্য আলাদা ভেরিয়েবলের দরকার হয় যা অ্যারে দিয়ে করা যায়। কজ

ফিডব্যাক : wahid_cseaust@yahoo.com