



সি প্রোগ্রামিংয়ে অ্যারের গুরুত্ব সম্পর্কে গত পর্বে আলোচনা করা হয়েছে। অ্যারের ব্যবহার একদিকে যেমন জটিলতা কমায়ে আরেক দিকে কোডারের জন্য প্রোগ্রাম লেখা সহজও করে। এছাড়া অ্যারের ব্যবহার তুলনামূলক অনেক কম সময়ে করা যায়। সাধারণ ভেরিয়েবল ব্যবহার করলে যেখানে অনেক বেশি সময় লাগে সেখানে অ্যারে ব্যবহার করলে অল্প সময়ে কোড লেখা সম্ভব।

সাধারণ অ্যারে কিভাবে ডিক্লেয়ার করতে হয়, কিভাবে অ্যারে মেমরিতে জায়গা দখল করে এবং কিভাবে তা কাজ করে সে সম্পর্কে বিস্তারিত আলোচনা করা হয়েছে। এ লেখায় নতুন ধরনের অ্যারে সম্পর্কে ধারণা দেয়া হয়েছে। অ্যারে নিয়ে এর আগে যত আলোচনা করা হয়েছে তার একটি লক্ষণীয় বিষয় হলো, অ্যারে সেখানে একক ডাইমেনশনের ছিল। অ্যারের ক্ষেত্রে ডাইমেনশনের ধারণা কিভাবে আসে তা প্রথমে অদ্ভুত লাগতে পারে। তবে ইউজারের জেনে রাখা ভালো অ্যারের মাঝেও ডাইমেনশনের কিছু ব্যাপার আছে। ডাইমেনশনের ব্যবহার আর কিছুই নয়, অ্যারেকে আরও প্রসারিত করা। অর্থাৎ সাধারণ অ্যারের জন্য যেখানে একটি এলিমেন্ট একটি ভেরিয়েবলের সমতুল্য, সেখানে একটি টুডি অ্যারের ক্ষেত্রে একটি এলিমেন্ট দুটি ভেরিয়েবলের সমান। সহজ কথায়, অ্যারের ডাইমেনশন যত বেশি হবে, অ্যারের প্রতিটি এলিমেন্ট দিয়ে ততগুলো ভেরিয়েবল প্রকাশ করা যাবে।

টুডি অ্যারে

সাধারণত ম্যাট্রিক্সের সমস্যার জন্য টুডি অ্যারে ব্যবহার করা হয়। টুডি অ্যারেকে রো এবং কলাম অনুযায়ী ভাগ করা যায়। তবে মেমরিতে কিন্তু এভাবে রো কলাম হিসেবে অ্যারে থাকে না। মেমরিতে অ্যারের গঠন সবসময় লিনিয়ার হয়। প্রোগ্রামে টুডি অ্যারে ব্যবহারের নিয়ম হলো :

```
data_type array_name[row_size][column_size];
```

এখানে ডাটা টাইপ বলতে যেকোনো ডাটা টাইপকে বোঝানো হচ্ছে। কাস্টম বা বিল্ট ইন সব ডাটা টাইপই ব্যবহার করা যাবে। অ্যারের নাম যেকোনো ভেরিয়েবলের নামানুসারে রাখা যাবে। রো সাইজ এবং কলাম সাইজ বলতে অ্যারের ইনডেক্স বোঝানো হচ্ছে। অর্থাৎ টুডি অ্যারের ক্ষেত্রে ইনডেক্স দুটি। যেমন : `int ar[2][2]`; এখানে একটি টুডি অ্যারে ডিক্লেয়ার করা হয়েছে যার ইন্ডেক্স ২, ২। অর্থাৎ এর রো এবং কলাম দুটি করে। এখন আমরা একে একটি ম্যাট্রিক্স বা ছক আকারে কল্পনা করতে পারি। বোঝার সুবিধার্থে এভাবে কল্পনা করা হয়। কিন্তু মেমরিতে অ্যারেটির জন্য পরপর চারটি ভেরিয়েবল ডিক্লেয়ার হবে। চিত্র-১-এ দেখানো হয়েছে কিভাবে টুডি অ্যারেকে ম্যাট্রিক্স আকারে কল্পনা করা হয়। লক্ষণীয়, এখানেও অ্যাস্কেসিং ০ থেকে শুরু হয়েছে। চিত্রে প্রতিটি

ঘর একেকটি ভেরিয়েবল। টুডি অ্যারের ধারণার পেছনে কিছু বিশেষ কারণ আছে। প্রোগ্রামে এমন অনেক সমস্যা হতে পারে যে বিভিন্ন ভেরিয়েবলকে গ্রুপিং করতে হবে। একটি ভেরিয়েবলের অন্তর্গত অনেকগুলো ভেরিয়েবল, আবার অন্য একটি ভেরিয়েবলের অন্তর্গত অনেকগুলো ভেরিয়েবল। এভাবে বিভিন্ন গ্রুপিংয়ের প্রয়োজন হতে পারে। এ ধারণা থেকে টুডি অ্যারের সূচনা। এখানে প্রতিটি রো-এর অন্তর্গত সব কলাম থাকে। চিত্র থেকে দেখলে বিষয়টি আরও পরিষ্কার হবে। চিত্রে রো ০-এর অন্তর্গত ০০ এবং ০১ ভেরিয়েবল, রো ১-এর অন্তর্গত ১০ এবং ১১ ভেরিয়েবল।

টুডি অ্যারে ব্যবহারের সুবিধা

টুডি অ্যারের মূল ব্যবহার হয় ম্যাট্রিক্সের সমস্যা সমাধান করার জন্য। কেননা ম্যাট্রিক্সে যেমন রো কলামের বিষয় থাকে, টুডি অ্যারেতেও সেভাবে কল্পনা করা যায়। ধরা যাক, কোনো একটি স্কুলে ক্লাস ১ থেকে ৫ পর্যন্ত প্রতিটি ক্লাসে ১০ জন করে ছাত্র আছে। এখন এ ৫০ জন ছাত্রের মার্ক কোনো কাজে ব্যবহার করতে হবে।

করে দেয়া হয়েছে। পরের উদাহরণেও ভেরিয়েবলগুলোর মান পরপর ডিফাইন করা হয়েছে। কিন্তু এখানে ডিফাইন করার সময় আলাদা রো হিসেবে করা হয়েছে। অর্থাৎ প্রথমে প্রথম রো এবং পরে দ্বিতীয় রো-এর ভেরিয়েবলগুলোর মান ডিফাইন করা হয়েছে। টুডি অ্যারের ক্ষেত্রে এটি মান ডিফাইন করার একটি বিশেষ নিয়ম। এর সুবিধা হলো, যদি ইউজার এমন চান যে প্রথম রো-এর একটি ভেরিয়েবলের মান এবং দ্বিতীয় রো-এর সব ভেরিয়েবলের মান সরাসরি ডিফাইন করা হোক, তাহলে এ নিয়ম অনুসরণ করা যায়। এখানে প্রতিটি রো আলাদাভাবে ডিফাইন করা হয়েছে। তার মানে হলো প্রথমে যে দ্বিতীয় বন্ধনীর ভেতরে ২ তার মান ডিফাইন করা হয়েছে সেগুলো হলো ০০ এবং ০১ ভেরিয়েবলের মান। এখন ০১ ভেরিয়েবলের মান যদি ইউজার না দেন, বাকি মানগুলো যদি দেন তাহলে প্রোগ্রাম সেভাবে মান ডিফাইন করবে। কিন্তু আলাদা রো অনুসারে মান ডিফাইন না করে সাধারণ নিয়মে সব ডিফাইন করা হলে এমনটি করা সম্ভব হতো না। আবার এই নিয়ম শুধু টুডি অ্যারের ক্ষেত্রেই

সহজ ভাষায় প্রোগ্রামিং সি/সি++

আহমদ ওয়াহিদ মাসুদ

তাহলে `int mark[5][10]`; এই অ্যারেটি ব্যবহার করা যায়। এখানে মোট পাঁচটি রো-এর অন্তর্গত ১০টি করে কলাম আছে। শুধু এ ধরনের সমস্যাই নয়, ম্যাট্রিক্সের সব সমস্যাই টুডি অ্যারে দিয়ে সমাধান করা যায়। যেমন ম্যাট্রিক্সের মান বের করা, ম্যাট্রিক্সের গুণ ইত্যাদি কঠিন সমস্যা টুডি অ্যারে দিয়ে সহজেই করা সম্ভব।

টুডি অ্যারের এলিমেন্ট ব্যবহারের নিয়ম

সাধারণ অ্যারে ব্যবহারের সময় দেখানো হয়েছে ইনডেক্সিং ০ থেকে শুরু করা হয়। এক্ষেত্রেও তার ব্যতিক্রম নয়। তবে এখানে ডান দিকের বন্ধনীর মান প্রথমে বাড়ে, পরে বাম দিকের বন্ধনীর মান বাড়ে। অর্থাৎ প্রথমে কলাম তারপর রো-এর মান বাড়ে। এখন অ্যারের কোনো নির্দিষ্ট এলিমেন্ট ব্যবহার করতে হলে সাধারণ অ্যারের নিয়ম মেনে চললেই হবে। শুধু পার্থক্য হলো এখানে একই সাথে দুটি ইনডেক্স ব্যবহার করতে হবে। যেমন `ar[1][0]` বলতে চিত্রের অ্যারেটির নিচের বাম দিকের ভেরিয়েবলকে বোঝায়। এটি দ্বিতীয় রো-এর প্রথম ভেরিয়েবল।

টুডি অ্যারের ভেরিয়েবলের মান বিভিন্নভাবে নির্ধারণ করা যায়। ডিক্লেয়ার করার সময়ই মান নির্ধারণ করা সম্ভব। `int a1[2][2]={{1,2,3,4}, int a2[2][2]={{1,2},{3,4}}}`। এখানে প্রথমে সরাসরি সব ভেরিয়েবলের মান পরপর ডিফাইন

প্রয়োজ্য। অন্য কোনো ধরনের অ্যারের জন্য যদি এ নিয়ম অনুসরণ করা হয় তাহলে প্রোগ্রাম এরর দেখাবে। একটি বিষয় খেয়াল রাখতে হবে। রোগুলোর মান আলাদাভাবে ডিফাইন করলে প্রতিটা রো-এর বন্ধনীর পর একটি করে কমা দিতে হবে। এখানে যদি সেমিকোলন দেয়া হয়, তাহলে প্রোগ্রাম এরর দেখাবে। সাধারণ নিয়ম অনুযায়ী সবার শেষে সেমিকোলন দিতে হবে।

টুডি অ্যারের ইনপুট নিতে হলে লুপ ব্যবহার করতে হয়। সাধারণত ফর ব্যবহার করেই ইনপুট নেয়া হয়। কিন্তু একটি লুপ ব্যবহার করে সাধারণ অ্যারের মান ইনপুট নেয়া যায়। টুডি অ্যারের ইনপুটের জন্য দুটি লুপের প্রয়োজন। এক্ষেত্রে আউটার লুপ দিয়ে রো এবং ইনার লুপ দিয়ে প্রতিটি কলাম বোঝানো হয়। নিচে শুধু ইনপুট নেয়ার একটি উদাহরণ দেয়া হলো :

```
for(row=0;row<limit;row++)
for(col=0;col<limit;col++)
scanf("%d",&arr[row][col]);
```

এখানে `arr[row][col]` নামে একটি অ্যারের ইনপুট নেয়া হয়েছে। ফর লুপে লিমিট নামে একটি ভেরিয়েবল ব্যবহার করা হয়েছে। এটি হলো অ্যারের ইনডেক্সের লিমিট। প্রথম ফর লুপ অ্যারেটির রো কাউন্ট করবে এবং ভেতরের লুপটি প্রতিটি রো-এর অন্তর্গত যে কলামগুলো আছে সেগুলোর আলাদাভাবে ইনপুট নেবে। এভাবে লুপের মাধ্যমে একটি টুডি অ্যারের এলিমেন্টগুলোর মান ইনপুট নেয়া সম্ভব। আবার

একটি টুডি অ্যারের মানগুলো প্রিন্ট করার জন্যও একইভাবে লুপ ব্যবহার করতে হবে। আর লুপ হিসেবে ফর লুপ ছাড়াও অন্য যেকোনো লুপ ব্যবহার করা যেতে পারে। শুধু কন্ডিশন ঠিক থাকলেই হবে।

লুপটি ভালোভাবে লক্ষ করলে দেখা যাবে এখানে নেস্টেড লুপ ব্যবহার করা হয়েছে। নেস্টেড লুপ সম্পর্কে আগেও বলা হয়েছে। একটি লুপের মাঝে আরেকটি লুপ ব্যবহার করলে তাকে নেস্টেড লুপ বলা হয়। কিভাবে ইনপুটগুলো নেয়া হচ্ছে, কেন আউটার লুপ রো এবং ইনার লুপ কলামের ইনপুট নেয় তা বোঝা কোডারের জন্য গুরুত্বপূর্ণ। আউটার লুপের ভেতরে ঢোকানোর পর প্রোগ্রাম ইনার লুপে ঢুকবে। ইনার লুপের কন্ডিশন যতক্ষণ না পর্যন্ত মিথ্যা হচ্ছে ততক্ষণ পর্যন্ত ইনপুট নেয়া হবে এবং এখানে ইনার লুপের ইনক্রিমেন্ট হচ্ছে, আউটার লুপের নয়। তার মানে চিত্র-১-এর মতো কোনো চিত্র কল্পনা করলে বোঝা যাবে। আউটার লুপের মান যখন ১, তখন ইনার লুপের সব মানের জন্য ইনপুট নেয়া হচ্ছে। অর্থাৎ প্রথম রো-তে প্রোগ্রাম ঢোকানোর পর নিচের দিকে না নেমে ডান দিকে প্রোগ্রাম যেতে থাকবে এবং ইনপুট নিতে থাকবে। ইউজার ইচ্ছে করলে নিচের দিকে পরপর ইনপুট নিতে পারেন, কিন্তু তা সঠিক নিয়ম হবে না। ডান দিকে যেতে যেতে যখন একটি রো সম্পূর্ণ হয়ে যাবে তখন ইনার লুপের কন্ডিশন মিথ্যা হয়ে যাবে। তাই আউটার লুপের ১ ইনক্রিমেন্ট হবে। তারপর আবার ইনার লুপ ০ থেকে চলা শুরু হবে এবং দ্বিতীয় রো-এর জন্য প্রোগ্রাম ডান দিকে যেতে থাকবে এবং ইনপুট নিতে থাকবে। এভাবে সব রো-এর ইনপুট নেয়া হবে। আউটার লুপের কন্ডিশন একসময় মিথ্যা হলে প্রোগ্রাম লুপ থেকে বের হয়ে আসবে এবং ইনপুট নেয়া বন্ধ হয়ে যাবে। তবে লুপের কন্ডিশন লেখার সময় খেয়াল রাখা উচিত যেমনো অ্যারের ইনডেক্স ওভারফ্লো না করে। অর্থাৎ লুপের কন্ডিশনে দেয়া আছে যে পাঁচটি রো পর্যন্ত মান নেয়া হবে, কিন্তু অ্যারেতে ডিক্লেয়ার করা আছে তিনটি রো। এক্ষেত্রে প্রোগ্রাম কোনো এরর দেখাবে না, কিন্তু ইনপুট নেয়ার সিকোয়েন্স ঠিক থাকবে না। তাই একটি ভেরিয়েবলের মান অন্য ভেরিয়েবলে চলে যাবে।

ম্যাট্রিক্স মাল্টিপ্লিকেশন

গণিতের খুবই প্রচলিত এবং কঠিন একটি সমস্যা হলো ম্যাট্রিক্সের গুণ করা। প্রোগ্রামে এ ধরনের কাজ টুডি অ্যারে দিয়ে করা হয়। এ ধরনের গুণ করার জন্য একটি নির্দিষ্ট নিয়ম মেনে চলা হয়। কারণ যদি এটি অনেক কঠিন মনে হয় তাহলে সরাসরি মুখস্ত করা ভালো। কারণ টুডি অ্যারের জন্য আর কোনো নিয়ম নেই। আর অ্যারে দিয়ে এই গুণনের কাজ খুব সহজেই করা যায়।

ম্যাট্রিক্স গুণ করার আগে এর সূত্রটি জানা প্রয়োজন। দুটি ম্যাট্রিক্স তখনই গুণ করা সম্ভব যখন প্রথম ম্যাট্রিক্সের কলাম সংখ্যা এবং দ্বিতীয় ম্যাট্রিক্সের রো সংখ্যা সমান হবে। এটি সব


ম্যাট্রিক্সের জন্য সত্য। আর গুণের ক্ষেত্রে প্রথম ম্যাট্রিক্সের রো এবং দ্বিতীয় ম্যাট্রিক্সের কলামের সাথে গুণ করে মানগুলো যোগ করে তার রেজাল্ট ম্যাট্রিক্সের প্রথম রো হিসেবে নির্ধারণ করতে হয়। নিচে শুধু গুণনের মূল লজিকটি কোডে দেয়া হলো :

```
for(i=0;i<row_a;i++)
{
    for(j=0;j<row_b;j++)
    {
        sum=0;
        for(k=0;k<col_b;k++)
        sum=sum+a[i][k]*b[k][j];
        c[i][j]=sum;
    }
}
```

এখানে তিনটি লুপ ব্যবহার করা হয়েছে, যার মাঝে দুটি হলো ইনডেক্স বাড়ানোর জন্য এবং একটি হলো গুণের জন্য। ম্যাট্রিক্সের গুণের এই কোডটি সবসময় একই হবে, তাই মুখস্ত করাটাই সবচেয়ে সহজ। তারপর কোডারের সুবিধার্থে লুপগুলো ব্যাখ্যা করা হলো। প্রথম লুপ দিয়ে প্রথম ম্যাট্রিক্সের রো নিয়ন্ত্রণ করা হচ্ছে। দ্বিতীয় লুপ দিয়ে দ্বিতীয় ম্যাট্রিক্সের কলাম নিয়ন্ত্রণ করা হচ্ছে। লক্ষ করলে দেখা যাবে, দ্বিতীয় লুপের কন্ডিশন হিসেবেই ম্যাট্রিক্সের রো-এর লিমিট দেয়া হয়েছে। এখানে ইচ্ছে করলে প্রথম ম্যাট্রিক্সের কলাম অর্থাৎ col_Aও দেয়া যায়। যেহেতু এই ম্যাট্রিক্স দুটি গুণ করা সম্ভব, তার মানে হলো প্রথম ম্যাট্রিক্সের কলাম এবং দ্বিতীয় ম্যাট্রিক্সের রো-এর সংখ্যা সমান।

এখানে sum নামে একটি ভেরিয়েবল নেয়া হয়েছে, যার কাজ ক্ষণস্থায়ী। আমরা জানি

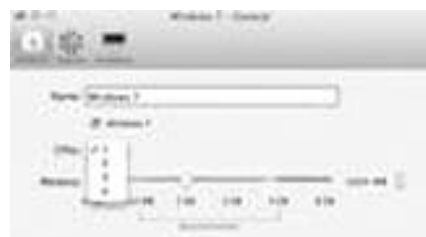
ম্যাট্রিক্সের গুণনে রো আর কলামগুলো গুণ করে সব যোগ করে তারপর রেজাল্ট ম্যাট্রিক্সের একটি এলিমেন্ট হিসেবে রাখা হয়। এখানে ভেরিয়েবলটি সেই কাজটিই করছে। মূল গুণের কাজটি হচ্ছে sum-এর মান নির্ধারণের লিনটিতে। এটি কোডার নিজে থেকে সিমুলেশন করলে বুঝতে পারবেন কিভাবে কাজ করছে। সিমুলেশন মানে হলো কোনো প্রোগ্রাম কমপিউটারে রান না করে নিজের হাতে কাগজে কলমে একটি একটি স্টেপ লেখা। এখানে sum-এর মান নির্ধারণের জন্য যে সূত্র লেখা হয়েছে, সিমুলেশন করলে দেখা যাবে গুণ করার পদ্ধতির সাথে তা মিলে যাচ্ছে। সবশেষে sum-এ যে মান জমা হচ্ছে তার রেজাল্ট ম্যাট্রিক্সে রাখা হচ্ছে। এই কোডের একটি খুবই গুরুত্বপূর্ণ অংশ হলো দ্বিতীয় লুপের পরে প্রত্যেকবার sum-এর মান ০ করে দিতে হবে। কারণ ভেরিয়েবলটির মান রিসেট না করলে পরবর্তী এলিমেন্টের মান নির্ধারণের সময় ভুল ফল দেবে।

প্রোগ্রামিংয়ে অ্যারের ব্যবহার অত্যন্ত গুরুত্বপূর্ণ এবং প্রায় সব প্রোগ্রামেই এর ব্যবহার দেখা যায়। প্রোগ্রামে প্রয়োজনমতো ইউজার সিঙ্গেল বা ডাবল ডাইমেনশন ব্যবহার করতে পারেন। এখানেই অ্যারের ব্যবহার সীমাবদ্ধ নয়, মাঝে মাঝে থ্রিডি অ্যারে ব্যবহারেরও প্রয়োজন দেখা দিতে পারে। এটি ব্যবহারের নিয়ম একই, শুধু ইনডেক্সের সংখ্যা একটি বেশি হবে। মাল্টি ডাইমেনশন অ্যারে হয়তো সব প্রোগ্রামে ব্যবহার হয় না, কিন্তু অনেক প্রোগ্রামেই আছে যেগুলো মাল্টি ডাইমেনশন অ্যারে ছাড়া করা খুবই কঠিন হয়ে যায়। তাই মাল্টি ডাইমেনশন অ্যারে সম্পর্কে ভালো ধারণা থাকা প্রয়োজন 


ফিডব্যাক : wahid_cseust@yahoo.com

(৭৪ পৃষ্ঠার পর)

ম্যাক কমপিউটারের ওএস এক্সে উইন্ডোজ চালু করা



চিত্র-১২

ধাপ-১৩ : স্বাভাবিক পিসির মতো ভার্চুয়াল মেশিন বন্ধ করতে পারবেন অথবা সাময়িকভাবে বন্ধ অর্থাৎ pause করলে অস্থায়ীভাবে কাজের মাঝে থেমে যাবে, যাতে আপনি দ্রুতগতিতে আবার কাজে ফিরে যেতে পারেন। প্যারালালের মাধ্যমে আপনি দ্রুতগতিতে মাল্টিপল ফাইল তৈরি এবং স্ল্যাপশট সেভ করতে পারবেন। একটি স্ল্যাপশট তৈরি করে ভার্চুয়াল মেশিনের কপি এবং সেভ করে যাতে যেকোনো সময় আপনি ফিরে পেতে পারেন 

ফিডব্যাক : mahmood_sw@yahoo.com

কারুকাজ বিভাগে লিখুন

কারুকাজ বিভাগের জন্য প্রোগ্রাম ও সফটওয়্যার টিপস বা টুকটাকি লিখে পাঠান। লেখা এক কলামের মধ্যে হলে ভালো হয়। সফট কপি সহ প্রোগ্রামের সোর্স কোডের হার্ড কপি প্রতি মাসের ২০ তারিখের মধ্যে পাঠাতে হবে।

সেরা ৩টি প্রোগ্রাম/টিপসের লেখককে যথাক্রমে ১,০০০, ৮৫০ ও ৭০০ টাকা পুরস্কার দেয়া হয়। সেরা ৩ টিপস ছাড়াও মানসম্মত প্রোগ্রাম/টিপস ছাপা হলে তার জন্য প্রচলিত হারে সম্মানী দেয়া হয়। প্রোগ্রাম/টিপসের লেখকদের নাম কমপিউটার জগৎ-এর বিসিএস কমপিউটার সিটি অফিস থেকেও জানা যাবে। পুরস্কার কমপিউটার জগৎ-এর বিসিএস কমপিউটার সিটি অফিস থেকে সংগ্রহ করতে হবে। সংগ্রহের সময় অবশ্যই পরিচয়পত্র দেখাতে হবে এবং পুরস্কার চলতি মাসের ৩০ তারিখের মধ্যে সংগ্রহ করতে হবে।