

جلسه یازدهم (جریان های ورودی خروجی)

مدیریت ورودی و خروجی ها در زبان ++C به عهده ی جریان های ورودی خروجی میباشد اشیا cin و cout در ارتباط با جریان ها هستند شیء cin دارای یک حافظه میانی است که تمام داده های ورودی از صفحه کلید را در این حافظه میانی ذخیره می نماید هنگامی که برنامه به اجرای یکی از توابع عضو یا عملگر های این شیء می رسد ابتدا بررسی می شود که چیزی در حافظه ی میانی برای خواندن وجود دارد اگر وجود داشته باشد داده های مورد نظر توابع یا عملگرها از این حافظه میانی استخراج می شود اما اگر این حافظه خالی بود برنامه منتظر تایپ داده ها توسط کاربر می ماند تا داده های جدید را در حافظه میانی قرار داده و سپس داده های مورد نظر را استخراج نماید.

اگر هر زمان که در حال وارد نمودن ورودی ها هستید کلید enter را فشار دهید یک کاراکتر اتهای خط به انتهای حافظه موقت اضافه می شود و cin به بررسی حافظه میانی می پردازد و عمل مناسبی انجام می دهد مثلا عددی را خوانده در متغیر مورد نظر کپی میکند یا مثلاً اگر چیز مناسبی برای خواندن پیدا نکرد دوباره به شما درخواست ورودی می دهد. مثل زمانی که بدون وارد نمودن داده ای enter را فشار دهید برای روشن شدن موضوع به مثال توجه کنید.

```
char x[100];
cin.getline(x, 100);
cin.getline(x, 100, '!');
cin.getline(x, 100, '!');
cin.getline(x, 100, '!');
```

اولین دستور `cin.getline(x, 100);` نسخه ی اول این تابع است که کاراکتر مرز بندی آن همان کاراکتر پایان خط یا `enter` است با اجرای این دستور هر چه تایپ کنید و بعد کلید `enter` را فشار دهید به حافظه منتقل می شود و چون این دستور از پایان خط برای کاراکتر مرز بندی استفاده می کند هر آنچه که وارد شده در رشته ی `x` کپی می شود. بعد از آن دومین دستور `cin` اجرا می شود این دستور حافظه میانی را بررسی میکند و چیزی برای خواندن پیدا نمی کند پس منظر وارد شدن داد ها توسط کاربر میماند بیا بعد از این دفعه با توجه این که کاراکتر مرز بندی نقطه تعریف شده یک عبارت حاوی دو نقطه وارد نماییم حالا کلید `enter` را فشار می دهیم دستور `cin` دوم حافظه میانی را بررسی میکند این بار تا پایان اولین کاراکتر نقطه خوانده میشود و تمام آنچه که ماقبل اولین نقطه است خوانده می شود و در رشته ی `x` کپی میشود حال آنچه در حافظه است دومین عبارت و نقطه دوم است. با اجرا سومین دستور `cin` حافظه میانی بررسی می شود چون در آن چیزهای مناسبی برای خواندن یک رشته وجود دارد دیگر از کاربر درخواست تایپ ورودی نمی شود و این دستور مانند دستور دوم تمام رشته موجود را بجز کاراکتر نقطه را در `x` کپی می نماید و دوباره دستور چهارم به بررسی حافظه می پردازد و چون این بار در حافظه چیزی باقی نمانده بز با درخواست تایپ ورودی رو برو می شویم.

نکته:

۱. بدلیل استفاده از کاراکتر نقطه برای کاراکتر مرز بندی فشردن کلید `enter` دیگر اتمام عملیات تایپ ورودی شناخته نمی شود و با فشردن این کلید `cin` به بررسی وجود کاراکتر مرز بندی می پردازد چون پیدایش نمی کند به درخواست تایپ ورودی ادامه می دهد. نتیجه این که رشته ای که خوانده می شود می تواند شامل کاراکتر انتهای خط باشد در واقع وقتی کلید `enter` را فشار می دهید فقط به خط بعد می روید و کاراکتر پایان خط را مانند تمام کاراکتر های دیگر به رشته ی ورودی اضافه می کنید. برای پایان عمیات تایپ باید کاراکتر نقطه را تایپ کنید و بعد `enter` را فشار دهید در این مرحله با فشردن کلید `enter` دوباره `cin` به بررسی وجود کاراکتر مرز بندی پرداخته و آنرا پیدا نموده لذا به عملیات تایپ پایان میدهد.

۲. اگر بعد از تمام دستورت cin باز هم چیزی در حافظه میانی باقی بماند در اجرا اولین دستور cin مورد بررسی و استفاده قرار می گیرد.

۳. در هر لحظه تایپ برای فرار از تایپ ادامه عبارت کاربر میتواند با کلید ترکیبی Ctrl+z (معروف به کاراکتر انتهای فایل) و فشردن کلید enter عملیات تایپ را خاتمه دهد در این حالت مقدار false را برمی گرداند. مثلا در مثال بالا برای خاتمه تایپ بجای تایپ نقطه و بعد از آن enter می توان Ctrl+z و بعد enter استفاده نمود. این مکنیزم معمولا برای cancel نمودن ورود عبارت تایپ شده استفاده می شود. ک البته cin این کار را انجام نمی دهد و شما باید با توجه به مقدار بازگشت داده شده توسط cin این کار را انجام دهید.

۴. در صورتی که عملیات خواندن داده ها به هر علتی با شکست مواجه شود cin مقدار false یعنی همان صفر را بر میگردداند. یک نمونه ی شکست تلاش برای خواندن حروف برای ورود به یک متغیر عدد صحیح است.

بررسی تعدادی از توابع شیء cin:

قبلا تابع getline بررسی کردیم حال با تعدادی دیگر از توابع این شیء آشنا می شویم این توابع به منظور دسترسی کاراکتر به کاراکتر حافظه ی میانی ساخته شده اند.

تابع cin.get(char)

این تابع یک کاراکتر از حافظه میانی خوانده و آن را از حافظه میانی حذف میکند. این تابع یک مقدار بازگشتی ۰ یا ۱ هم دارد که بیانگر شکست یا موفقیت عملیات تابع است.

تابع cin.putback(char)

آخرین کاراکتر خوانده شده را با کاراکتر آرگمان خود رونویسی نموده اولین دستور cin پردازش را از این کاراکتر آغاز خواهد کرد. به بیان دیگر آخرین کاراکتر خوانده شده را به حافظه میانی برمی گرداند تا برای دستور cin بعدی مجدد قابل خواندن باشد که البته این امتیاز را هم میدهد که بجای کاراکتر خوانده شده کاراکتر دیگری در حافظه نوشت.

تابع cin.peek()

این تابع یک مقدار بازگشتی از نوع int دارد که کاراکتر بعدی را می خواند بدون این که کاراکتر مذکور از حافظه ی میانی حذف شود. در واقع ترکیبی از دو تابع قبل.

تابع cin.ignore(int n,int delim)

این تابع تعداد n کاراکتر را بدون پردازش نادیده میگیرد و اگر قبل از پایان n کاراکتر به کاراکتر delim برخورد کند عملیاتش را متوقف میکند. برای مثال دستور cin.ignore(۱۰, 'x'); سعی میکند ده کاراکتر بعدی را رد کند و اگر مثلا کاراکتر پنجم x بود همین جا کار خود را متوقف می کند. دقت کنید که خود کاراکتر delim که در مثال x است نیز در نظر گرفته نمی شود. این تابع مقادیر پیش فرض یک و Ctrl+z (یا eof) (انتهای فایل) برای آرگمان هایش در نظر گرفته شده به این معنی که مثلا دستور cin.ignore(); یک کاراکتر را در نظر نمی گیرد و یا دستور cin.ignore(۱۰); ده کاراکتر بعدی را در نظر نمی گیرد و کاراکتر انتهای فایل را برای delim در نظر نمی گیرد.

تمرین :

کاربر یک چند جمله ای وارد میکند (حد اکثر ۳ جمله) با استفاده از توابع فوق برنامه ای بنویسید که ضریب و توان جملات وارد شده را در یک آرایه ی ۶ عنصری قرار دهد. کار بر از هشتک یا علامت کنترل (^) برای نمایش توان استفاده می کند.
مثال: $3x^2+4x-5$

جواب تمرین در جلسه بعد بررسی می شود. جلسه آینده به بررسی ساختار ها می پردازیم.