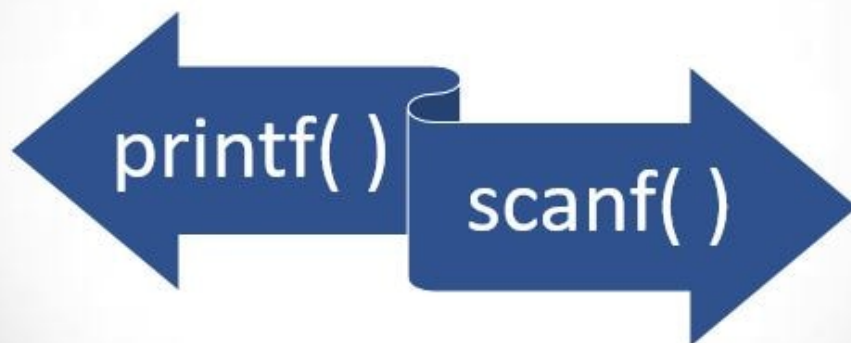


## کد نامه

ویژه دانش‌جویان مبانی برنامه‌سازی نیم‌سال اول ۱۴۰۰-۱۳۹۹ دانشکدهی مهندسی کامپیوتر دانشگاه صنعتی شریف

### Formatted I/O functions



در این شماره از  
کدنامه، می‌خوانید:



اسرار printf و scanf



آشنایی با خطاهای متداول  
در کوئرا

### اسرار printf و scanf

#### نکات نهان این دو تابع را بیش‌تر بشناسید

توابع printf و scanf، علی‌رغم ظاهر ساده‌شان، امکانات بسیار گوناگونی را در اختیار برنامه‌نویس می‌گذارند، به طوری که تنها با استفاده از همین دو تابع به نحوی خاص و استفاده نکردن از هیچ ساختار دیگری هم‌چون حلقه و یا شرط، می‌توان بسیاری از برنامه‌ها را نوشت. با توجه به کاربرد اسرار این توابع در تمرین‌های درس، لازم است با این توابع بیش‌تر آشنا شوید. با «کدنامه» همراه باشید.

28 آبان

دستورات شرطی

آشنایی با ساختارهای شرطی  
(مشابه لوزی فلوچارت) در زبان  
C (مانند دستور if)

26 آبان

فرمت‌بندی ورودی/خروجی

آشنایی با توابع مهم printf و  
scanf در زبان C

24 آبان

مقدمات برنامه‌سازی

آشنایی با ساختار یک برنامه به  
زبان برنامه‌نویسی C

مطالب تدریسی شده

در کلاس درس

توسط استاد در

هفته‌ی گذشته

## اسرار printf و scanf

علی حاتمی

specifier[length][precision].[width][flags]%

حال می‌خواهیم با بخش‌های مختلف این کلمات، بیش‌تر آشنا شویم. توصیه می‌کنیم خودتان با نوشتن چند برنامه، بر این موارد مسلط شوید.

## Specifier

تنها بخش اجباری یک کلمه‌ی درصد دار، نوع داده‌ی است که در خروجی چاپ خواهد شد؛ برای مثال از **C** برای چاپ کاراکتر و از **d** برای چاپ اعداد دهدهی علامت دار استفاده می‌شود. این بخش، همان کاراکتری است که در کلاس خوانده‌اید. تعدادی دیگر از پرکاربردترین این حروف که برخی از آن‌ها را در کلاس نیز دیده‌اید، به شرح زیر است:

- **f**: برای چاپ اعداد اعشاری دهدهی

- **O**: برای چاپ اعداد در مبنای ۸

- **X** یا **x**: برای چاپ اعداد در مبنای ۱۶

- **E** یا **e**: برای چاپ اعداد به صورت فرم علمی

**سوال:** به نظر شما، چگونه می‌توان خود کاراکتر % را در خروجی چاپ کرد؟

## Flags

- - یعنی همترازی چپ (تراز کردن سمت چپ اعداد چاپ‌شده با یک‌دیگر) که میزان آن (تعداد فاصله‌هایی که اول عدد زده می‌شود)، در **width** مشخص خواهد شد. همترازی راست به صورت پیشفرض و بدون علامت انجام خواهد شد.

- + یعنی اعداد در هر صورتی علامت خواهد داشت. (علامت مثبت برای اعداد مثبت و علامت منفی برای اعداد منفی) در صورت عدم وجود این نماد، اعداد مثبت بدون علامت مثبت نوشته خواهند شد.

- **(space)**: اگر عدد علامت نداشته باشد، یک فاصله به جای علامت آن چاپ خواهد شد.

- **(صفر):** در هنگام چاپ، به تناسب **width**، در سمت چپ عدد، رقم صفر چاپ خواهد شد.

## Width

**Width** کمترین تعداد ارقامی است که در خروجی چاپ خواهد شد. برای مثال اگر مقدار آن پنج باشد و عددی که قرار است به وسیله آن چاپ شود

## ساختار نوشته‌ها در printf

تاکنون کاربردهای زیادی از تابع پردردسر (!) **printf** در کلاس درس دیده‌اید. در این مطلب، می‌خواهیم تعدادی از اسرار نهفته‌ی **printf** را برای شما آشکار سازیم.

## علائم مهم و پراستفاده در عبارتی که قرار است چاپ شود

- **\n**: از این کاراکتر برای رفتن به خط بعد استفاده می‌شود.

- **\t**: این کاراکتر به اندازه یک **Tab** فاصله می‌اندازد که معمولاً به اندازه‌ی هشت فاصله (**space**) است.

- **'**: برای نوشتن تک کوتیشن از آن استفاده می‌شود.

- **"**: برای نوشتن دابل کوتیشن از آن استفاده می‌شود.

- **\\**: برای نوشتن یک اسلش استفاده می‌شود. (زمانی که از یک اسلش استفاده می‌کنید، زبان **C**، انتظار یک کاراکتر کنترلی بعد از آن را دارد. ممکن است نوشتن یک بک اسلش به تنهایی مشکلی ایجاد نکند، ولی بهتر است برای نوشتن بک اسلش از **\\** استفاده کنید)

- **\0**: به معنی کاراکتر **Null** است. اگر از این کاراکتر در میان یک رشته استفاده کنید، در هنگام چاپ تنها هر آن‌چه تا قبل از این کاراکتر نوشته شده است، چاپ خواهد شد و باقی آن چاپ نخواهد شد، اگرچه در کد نوشته شده‌اند.

## کلمات درصد دار

بسیاری از اوقات در برنامه‌نویسی نیازمندیم تا مقدار یک متغیر را چاپ کنیم و همان‌طور که می‌دانید، این کار به کمک کلمات درصد دار، هم‌چون **%d** و یا **%C** انجام می‌شود؛ اما جدای از این عبارات ساده، امکان افزودن پیچیدگی به این عبارات نیز وجود دارد. مثلاً، فرض کنید می‌خواهیم عددی را در مبنایی خاص و یا با تعداد رقم اعشار مشخصی چاپ کنیم. به این منظور، لازم است با ساختار دقیق این کلمات درصد دار، بیش‌تر آشنا شویم. فرمت کلی این کلمات به شکل زیر است (کروشه‌ها تایپ نمی‌شوند):

## آشنایی با خطاهای متداول در کوئرا

### سید پارسا نشایی

از تمرین یک به بعد، تمرین‌ها پس از ارسال به صورت خودکار توسط کوئرا داوری می‌شوند و نمره‌ی سوال به شما اعلام می‌شود. پس از آپلود هر فایل پاسخ، وضعیت پاسخ «در حال داوری» شده و اگر پس از چند ثانیه، صفحه را refresh کنید، نمره‌ی سوال (غالباً از ۱۰۰) در جلوی اسم سوال در لیست «ارسال‌ها» به شما نمایش داده خواهد شد. با کلیک روی نمره، می‌توانید لیستی از test case ها را به همراه نمره‌ی شما در هریک، مشاهده کنید. هر test case، یک ورودی نمونه است که به کد شما داده شده است و خروجی کد شما با خروجی مورد انتظار، تطبیق داده شده و در صورت عیناً یکسان بودن، نمره‌ی آن را دریافت خواهید کرد. اخذ نکردن نمره‌ی هر تست کیس، می‌تواند به یکی از دلایل زیر باشد:

- خطای Wrong Answer - کد شما خروجی درست را تولید نکرده است و احتمالاً یکی از نکات تمرین را در نظر نگرفته‌اید.
- خطای Compilation Error - حتماً از زبان C به جای زبان C++ استفاده کنید، وگرنه ممکن است به این خطا برخوردید.
- خطای Syntax Error - دستوری را در زبان C اشتباه نوشته‌اید. با کامپایل کردن دوباره روی دستگاه خود، اشکال را ببینید.
- خطای Time Limit - اجرای هر کد، باید در زمانی که بالای سوال مشخص شده، به اتمام برسد، در غیر این صورت، لازم است کد خود را بهینه کنید تا در زمان کوتاه‌تری اجرا شود.
- خطای Memory Limit - اجرای هر کد، باید حافظه‌ای کمتر از محدودیت اعلام شده در بالای سوال را اشغال کند.
- خطای Runtime - سایر خطاهای ممکن، هم‌چون تقسیم عدد بر صفر و یا - همان‌گونه که بعداً خواهید خواند - دسترسی به عناصری از آرایه که موجود نیستند، که در حالت عادی باعث کرش کردن برنامه و خروج ناگهانی از آن می‌شوند، در این دسته جای می‌گیرند.

سه رقمی بود، دو فضای خالی پشت عدد قرار خواهد گرفت. اگر در محل width عدد بگذارید آن مقدار به عنوان عرض در نظر گرفته خواهد شد و اگر \* بگذارید مقدار عرض را به عنوان پارامتر ورودی (همان گونه که یک عدد int را برای چاپ شدن به تابع می‌فرستید)، باید به تابع بفرستید.

### Precision

این مقدار برای مقادیر صحیح، کمترین تعداد ارقامیست که نوشته خواهد شد، برای مقادیر علمی و اعشاری، بیشترین تعداد ارقام در اعشار است و در مقادیر کاراکتری، بیشترین تعداد کاراکتری است که چاپ خواهد شد. این مقدار به دو صورت (number). و \* نوشته می‌شود. در حالت اول، مقدار آن همان عددی است که آمده و در حالت دوم مقدار آن به عنوان پارامتر ورودی، مشابه قسمت قبل، به تابع printf داده خواهد شد.

### Length

این قسمت بلندی نوع داده را مشخص می‌کند که می‌توانید با جست‌وجو در اینترنت با مقادیر مجاز برای آن، آشنا شوید.

### ساختار نوشته‌ها در scanf

همان‌طور که می‌دانید، قالب‌های خاص در scanf نیز همانند همانند printf باید از ساختار خاصی پیروی کنند:

specifier[length][width][\*]%

### specifier

این بخش، همان کاراکتری است که در کلاس خوانده‌اید و برای ورودی گرفتن به انواع مختلف، از آن استفاده می‌کنید.

### Length و Width

کاربرد این دو مورد در scanf، مشابه کاربردشان در printf است.

### نماد ستاره

ستاره، هنگامی کاربرد دارد که مقداری از ورودی خوانده شده، اما در جایی ذخیره نشود؛ مثلاً اگر در scanf از عبارت %s استفاده کنیم، وقتی برنامه به این قسمت می‌رسد، هر چه رشته دید نادیده می‌گیرد (scan نمی‌کند) و پس از اتمام رشته‌ها، به سراغ بقیه‌ی آن‌چه در تابع باقی مانده است، می‌رود.