

| Time | Narration |
|-------|--|
| 00:02 | خوش آمدید ++Logical operators in C and C به برنامه آموزشی |
| 00:08 | مثلاً logical AND logical AND یادمی گیریم. مثل logical در این برنامه در مورد اپراتورهای expression1 && expression2 |
| 00:16 | Logical OR مثل expression1 OR expression2 |
| 00:21 | Logical NOT مثل not (Expression1) |
| 00:25 | و این را با کمک مثال انجام می دهیم |
| 00:28 | و gcc نسخه 11/10 و کامپایلر Ubuntu Operating System برای ضبط این برنامه من از |
| 00:33 | استفاده کرده ام ubuntu نسخه 4/6/1 در |
| 00:39 | آغاز می کنیم logical operators با مقدمه ای در مورد |
| 00:43 | هر مقدار غیر از 0 می باشد true و ++C, و C در |
| 00:48 | true غیر از صفر یعنی |
| 00:50 | false و صفر یعنی |
| 00:53 | می دهند false و 0 را برای true را استفاده می کنند 1 را برای Logical operators عباراتی که |
| 00:58 | را با کمک مثالی توضیح می دهم Logical operators حالا |
| 01:03 | می باشد C در Logical operators در اینجا یک برنامه برای |
| 01:08 | main در قسمت |
| 01:10 | را بعنوان عدد صحیح اعلام می کند c و a,b این دستور, متغیرهای |
| 01:16 | را وارد کند c و a, b از کاربر می خواهد که مقدار printf دستور |
| 01:21 | را از کاربر می گیرد c و a, b ورودی های متغیرهای scanf دستور |
| 01:28 | را برای پیدا کردن بیشترین مقایسه می کنیم c و b و a در اینجا ما مقدار |
| 01:33 | استفاده می کنیم logical AND برای مقایسه همزمان از اپراتور |
| 01:38 | را بدهد true باید درست باشد تا logical AND در اینجا تمام حالتها برای |
| 01:43 | در صورت حالت اشتباه, عبارت ارزیابی نمی شود |
| 01:49 | درست باشد ارزیابی می شود (a > b) اگر (a > c) پس عبارت |
| 01:56 | باشد دیگر عبارت ارزیابی نمی شود b کمتر از a اگر |
| 02:02 | این دستور در صورتی که حالت قبلی درست باشد ارزیابی می شود |

| | |
|-------|---|
| 02:07 | ارزیابی میشود (b>c) سپس |
| 02:10 | روی صفحه نمایش داده می شود b is greatest اگر حالت درست باشد |
| 02:16 | روی صفحه نمایش داده می شود c is greatest در غیر این صورت |
| 02:21 | می رویم logical OR حالا به اپراتور |
| 02:24 | مقدار درست را بدهد logical OR اینجا یکی از حالتها باید درست باشد تا |
| 02:30 | در صورت حالت درست , عبارت بعداً ارزیابی نمی شود |
| 02:35 | پس دو عبارت دیگر ارزیابی نخواهند شد a==zero اگر |
| 02:43 | اجرا می شود printf 0 باشد دستور (صفر) c یا a, b اگر یکی از |
| 02:49 | و آکولاد بسته return 0 به پایان برنامه می آیم |
| 02:54 | حالا برنامه را ذخیره کنید |
| 02:57 | کنید save (ذخیره) c. آن را با امتداد |
| 03:00 | ذخیره کرده ام logical.c من فایل را با نام |
| 03:03 | باز کنید T و Ctrl,Alt پنجره ترمینال را با فشار دادن همزمان کلیدهای |
| 03:08 | Enter. را تایپ کنید gcc space logical dot c space minus o space log برای کامپایل کد را فشار دهید |
| 03:23 | را تایپ کنید ./log برای اجرا |
| 03:27 | را فشار دهید Enter |
| 03:29 | من مقادیر 0, 34, 567 را وارد می کنم |
| 03:39 | خروجی نمایش داده می شود |
| 03:42 | C is greatest |
| 03:45 | صفر می باشد c و a, b حاصل ضرب |
| 03:50 | شما این برنامه را با ورودی های دیگر اجرا کنید |
| 03:55 | می نویسیم ++C حالا همین برنامه را در |
| 03:59 | من از قبل برنامه را نوشته ام |
| 04:03 | می باشند ++C در اینجا کدها در |
| 04:06 | می نویسیم. تغییراتی می دهیم ++C حالا همین برنامه را در |
| 04:11 | می باشد header تغییر در فایل |

| | |
|-------|--|
| 04:14 | را استفاده کرده ام <code>using</code> دستور |
| 04:18 | همچنین در دستوره‌های ورودی و خروجی تغییراتی می باشد |
| 04:21 | عمل می کنند C اپراتورها با همان روش در |
| 04:25 | را کلیک کنید <code>Save</code> |
| 04:27 | ذخیره کنید <code>cpp</code> فایل را با امتداد |
| 04:31 | باز کنید <code>T</code> و <code>Ctrl,Alt</code> ترمینال را با فشار دادن همزمان کلیدهای |
| 04:36 | را فشار <code>Enter</code> را تایپ کنید <code>log1 space minus o space logical.cpp ++g</code> برای کامپایل دهید |
| 04:49 | را تایپ کنید <code>log1 /</code> برای اجرا |
| 04:53 | را فشار دهید <code>Enter</code> |
| 04:56 | من مقادیر <code>0, 34, 567</code> را وارد می کنم |
| 05:02 | می باشد C می بینیم که خروجی مثل همان در برنامه |
| 05:05 | برنامه را با ورودی های دیگر اجرا کنید |
| 05:10 | حالا یک اشتباه متداول را بررسی می کنیم |
| 05:12 | باز می گردیم <code>editor</code> به |
| 05:16 | فرض کنید در اینجا براکتها را فراموش کرده ام |
| 05:20 | این و این را حذف کنید |
| 05:26 | حالا ببینیم چه می شود. برنامه را ذخیره کنید |
| 05:30 | به ترمینال بازگردید |
| 05:32 | مثل قبل کامپایل و اجرا کنید |
| 05:38 | این اشتباه را می بینیم |
| 05:41 | Expected identifier before '(' token. |
| 05:45 | زیرا در اینجا دو عبارت متفاوت داریم |
| 05:48 | ارزیابی کنیم <code>AND</code> ما باید آن‌ها را بعنوان یک عبارت با استفاده از اپراتور |
| 05:53 | حالا به برنامه برمی گردیم و اشتباه را تصحیح می کنیم |
| 05:57 | براکت را اینجا و اینجا وارد کنید |

| | |
|-------|--|
| 06:04 | را کلیک کنید Save |
| 06:06 | به ترمینال باز گردید |
| 06:09 | مثل قبل کامپایل و اجرا کنید |
| 06:14 | و عمل می کند |
| 06:22 | حالا برنامه را خلاصه می کنیم |
| 06:24 | logical AND مثل $((a > c) \&\& (a > b))$ در این برنامه این موارد را یاد گرفتیم |
| 06:32 | Logical OR مثل $(a == 0 \parallel b == 0 \parallel c == 0)$ |
| 06:39 | ارائه |
| 06:41 | یک برنامه که دو عدد را بعنوان ورودی از کاربر می گیرد بنویسید |
| 06:44 | را استفاده کنید NOT بررسی کنید که آیا دو عدد مساوی هستند. اپراتور $(a != b)$ راهنمایی |
| 06:54 | ویدیو را در لینک زیر مشاهده کنید |
| 06:57 | می باشد spoken tutorial این خلاصه پروژه |
| 06:59 | اگر پهنای باند خوبی ندارید، ابتدا دانلود و سپس مشاهده کنید |
| 07:03 | ارائه میدهد spoken tutorial کارگاه آموزشی استفاده از Spoken Tutorial تیم پروژه |
| 07:07 | و به کسانی که آزمون آنلاین را قبول شوند گواهینامه میدهد |
| 07:11 | ایمیل بفرستید "contact@spoken-tutorial.org" برای جزئیات بیشتر لطفا به |
| 07:18 | می باشد Talk To a Teacher بخشی از پروژه Spoken tutorial |
| 07:21 | که تحت پشتیبانی National Mission on Education دولت هند می باشد MHRD توسط ICT, از طریق |
| 07:27 | موجود spoken hyphen tutorial dot org slash NMEICT hyphen intro اطلاعات بیشتر |
| 07:30 | می باشد |
| 07:37 | ترجمه و صداگذاری توسط شبنم اقبال خان... با تشکر از شما |