



شرط

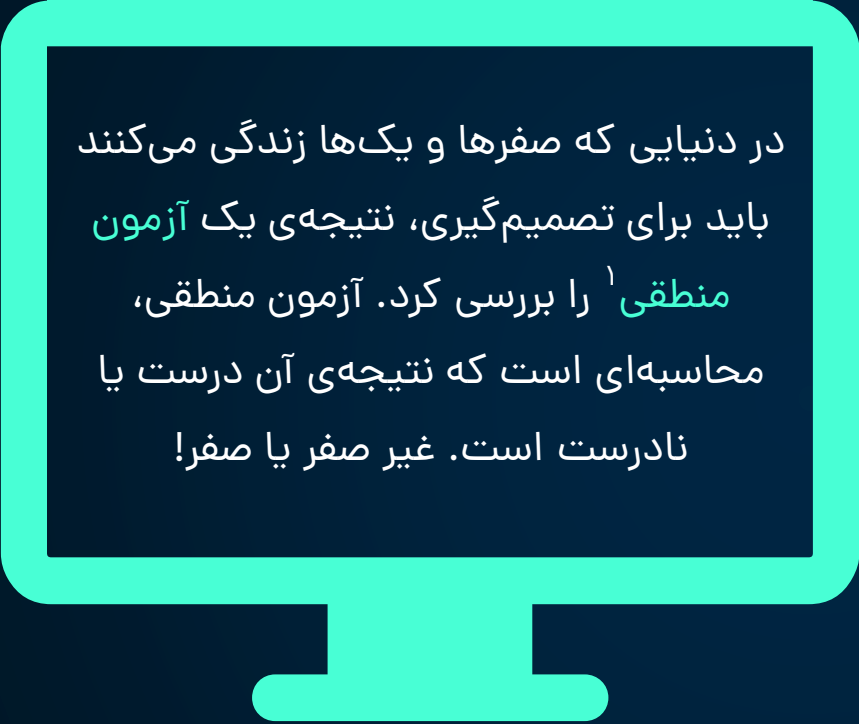
جلسه ششم

بسم الله الرحمن الرحيم

Condition



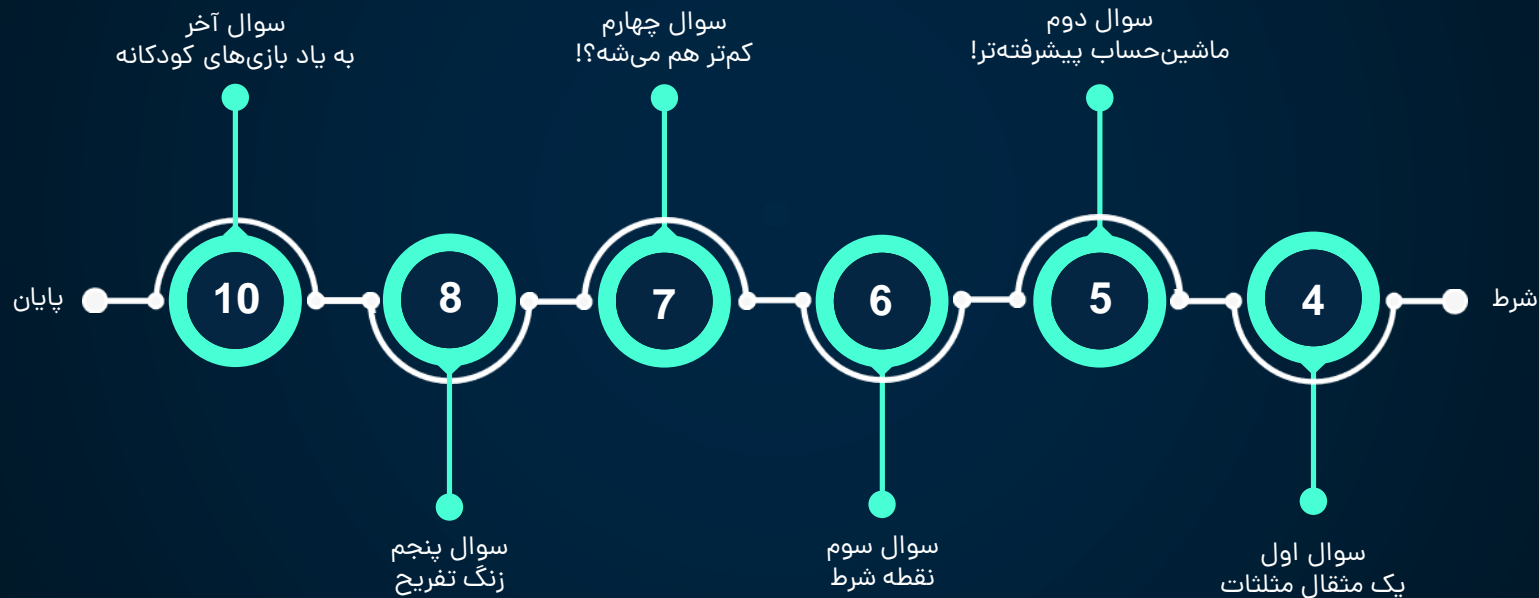
کارگاه مبانی برنامه‌نویسی - دانشکده مهندسی کامپیوتر دانشگاه هوایی شهید ستاری




در دنیایی که صفرها و یک‌ها زندگی می‌کنند
باید برای تصمیم‌گیری، نتیجه‌ی یک **آزمون**
منطقی^۱ را بررسی کرد. آزمون منطقی،
محاسبه‌ای است که نتیجه‌ی آن درست یا
نادرست است. غیر صفر یا صفر!


ما هر روز برای انجام هر یک از کارهایمان در حال تصمیم‌گیری هستیم. مثل این که انتخاب کنیم چه غذایی بخوریم؟ آیا لازم هست برای خروج از خانه لباس گرم بپوشیم یا نه. همه‌ی این تصمیم‌گیری‌ها مبتنی بر شرایط خاصی هستند و ممکن است عوض شدن اوضاع، منجر به تغییراتی در تصمیمات ما شود. اهمیت شرط در زندگی کاملاً مشهود است و نیازی به بازگو کردن آن نیست؛ پس باید ببینیم چه‌طور می‌توانیم شرط‌های مورد نظرمان را در دنیای کامپیوتر اعمال کنیم.

فهرست



سوال اول: یک مثال مثلثات

برنامه‌ای بنویسید که سینوس و علامت کسینوس یک زاویه را دریافت کند، کسینوس آن را محاسبه کرده و سپس بگوید زاویه در کدام ربع دایره‌ی مثلثاتی واقع شده است. 

اگر به خاطر داشته باشید، در جلسه‌ی قبل گفتیم فرض می‌کنیم همه‌ی ورودی‌های برنامه کاملاً منطقی و درست هستند؛ چراکه هنوز توانایی لازم برای محدود کردن ورودی‌ها را نداریم. 

به نظر شما، ممکن است این کد با چه خطاهایی مواجه شود؟ 

برای مثال، آیا سینوس می‌تواند مقدار ۵ را داشته باشد؟ اگر این عدد را وارد کنید، برنامه با چه مشکلی روبه‌رو می‌شود؟

سعی کنید ورودی‌های نادرست را به کمک شرط‌ها محدود کنید. 

سوال دوم: ماشین حساب پیشرفته‌تر!



در این بخش به سراغ کد ماشین حساب جلسه‌ی قبل بروید که آن را با کمک کد خدا و Botfather دیباگ کرده بودید. حال سعی کنید که ماشین حساب خود را ارتقا دهید و خواسته‌ی سوال را به عنوان یکی از بخش‌های جدید ماشین حساب اضافه کنید.

همچنین کد را با کمک switch case بازنویسی کنید. یعنی بخشی که لازم است ماشین حساب شما بداند چه عملیاتی را باید انجام دهد، به جای استفاده از if، با switch case تشخیص دهد.

توجه: به نوع ورودی‌های قبلی ماشین حساب دقت کنید. برنامه در بخش‌های قبلی همیشه دو عدد اعشاری یا صحیح را دریافت می‌کرد، اما در این سوال ورودی دوم شما از نوع کاراکتر است. از چه راه‌هایی می‌توانید این مشکل را برطرف کنید؟




راهنمایی: برای این بخش یک علامت مشخص را انتخاب کنید بعد تنها کافیست دو بخش کد خود را با هم ترکیب کنید.

سوال سوم: نقطه شرط

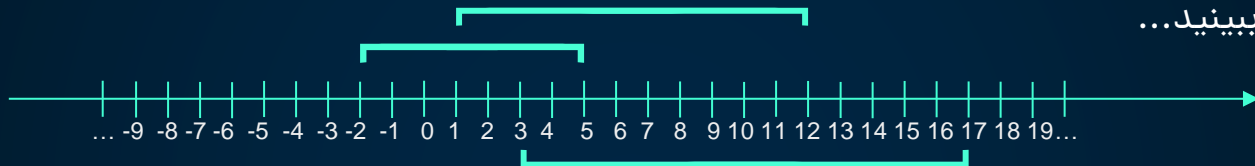
در ذهن خود یک مرور سریع بر درس هندسه داشته باشید! 

در نگاهی گذرا می‌توانید ببینید که در این درس از شرطهای زیادی استفاده می‌کنیم؛ به عنوان مثال: اگر شیب دو خط برابر باشد، می‌گوییم این دو خط موازی هستند. ما می‌توانیم با برنامه‌نویسی، بسیاری از همین شرطها را با سرعت و دقت بیشتری چک کنیم تا به نتیجه برسیم.

برنامه‌ای بنویسید که مختصات طول و عرض چهار نقطه را به عنوان ورودی بگیرد و اگر همه‌ی این نقاط روی یک خط باشند، عبارت "All in one line"، اگر سه نقطه روی یک خط باشند، عبارت "Three in one line"، اگر این چهار نقطه یک لوزی را تشکیل می‌دهند، عبارت "Diamond" و اگر هیچ یک از شرطهای بالا درست نبود، عبارت "None" را در خروجی چاپ کند. 

سوال چهارم: کم تر هم می شه؟!

شکل زیر را ببینید...



بازه‌ای که بین هر سه بخش هم‌پوشانی دارد را بیابید. چه‌طور این کار را کردید؟ حال همان شرط‌هایی که در ذهن خود مرور کردید تا به بخش مشترک بین سه بازه برسید را به کامپیوترتان نیز آموزش دهید! برای این کار، برنامه‌ی شما باید ابتدا سه بازه از اعداد را دریافت کند. این سه بازه در خطوطی متوالی به برنامه داده می‌شود که در هر خط، عدد اول و آخر هر بازه نوشته می‌شود. حال شما می‌خواهید با کمک شرط‌ها و مطالب قبلی‌ای که آموخته‌اید، ببینید که اشتراک این بازه‌ها چه اعدادی هستند.

نکته‌ی مهم برنامه‌ی شما این است: این کار را با کم‌ترین تعداد شرط ممکن انجام دهید.

آیا برنامه‌ای که نوشتید با تعداد شرط کمتر هم برقرار می‌شود؟

سوال پنجم: زنگ تفریح

قطعه کد زیر را اجرا کنید و در مورد علت هر خروجی‌ای که مشاهده می‌شود، بحث کنید. (به جز خط آخر. خط آخر را فقط ببینید 😊 در آینده با این کاراکتر بیشتر آشنا خواهید شد)



```
printf("true = %d\tfalse = %d\n", true, false);
if (3 * 4)
    printf("Result of 3 * 4 is: %d\n", 3 * 4);
if (3 * 4 == 12)
    printf("Result of 3 * 4 == 12 is: %d\n", 3 * 4 == 12);
if ('b')
    printf("ASCII code of 'b' is: %d\n", 'b');
if (' ')
    printf("What is space ASCII code?\n");
if ('\0')
    printf("****the ASCII code of \0 is 0****\n");
printf("You will learn more about \0 in \"String\". BOOOM!");
```


گذاشتن `stdbool.h` در ابتدای برنامه برای شناختن `true` و `false` فراموش نشود.



به یاد دارید در ابتدای دستورکار گفتیم شرطها نتیجه‌ی یک آزمون منطقی را چک می‌کنند و این نتیجه می‌تواند صفر باشد یا غیر صفر؟



اینجا دیدید با وجود این‌که تنها `*` باید معادل نادرست و `1` معادل درست باشد، این‌طور نیست و `if` هر عددی غیر از صفر را درست در نظر می‌گیرد.



حالا در ادامه خطوط زیر را بررسی کنید...

```
int a = 5;
int b = 0;
int c = printf("Condition\n");

if (a && b || c) {
    printf("Line 1 - %d is true\n", c);
} else {
    printf("Line 1 - Condition is not true\n");
}

if (printf("") && printf("I am true")) {
    printf("Line 2 - Condition is true\n");
}
```

و اما سوال آخر: به یاد بازی‌های کودکانه



بابا می‌خوایم بازی کنیم نمی‌شه که خشک صحبت کنیم.

همون دستورکار قبلی کافی بود. من نمی‌تونم کتابی حرف بزنم.



باشه باشه قبول کردم دیگه همون عادی صحبت کنیم.



بریم بچه‌ها منتظرن... برو شروع کن.

سلام و خدا قوت به شماهایی که تا این‌جای دستورکار رو انجام دادین. حالا سوال آخر رو با هم پیش می‌بریم که نشون بدیم حتی بازی‌های بچگانه رو هم می‌تونیم به کامپیوترمون یاد بدیم.



از شما می‌خوایم قبل از هر کاری، کد ارسالی رو بررسی کنین. فکر می‌کنین برنامه‌ای که نوشته شده (با این‌که الان ناقصه و قراره که کاملش کنیم)، داره قوانین کدوم بازی رو می‌گه؟





همون طور که احتمالا حدس زدین، این پروژه دقیقا همون بازی سنگ کاغذ قیچی... پس می‌خوایم با کامپیوتر سنگ کاغذ قیچی بازی کنیم :)

چه‌طوری؟

برای شروع بازی، اول باید با هم به توافق برسین که برنده‌ی بازی باید به چند امتیاز برسه؟ یعنی لازمه که کامپیوتر شرط خاتمه‌ی بازی رو بدونه، وگرنه تا جان در بدن داشته باشه باید باهاش بازی کنین (=)



قدم دوم چیه؟ یه دستتون رو ببرین پشتتون و اون یکی دست رو کنار صورتتون نگه دارین تا تعداد دفعاتی که برنده می‌شین رو بشمرین.

حالا کامپیوتر وارد بازی می‌شه. متاسفانه یا خوش‌بختانه کامپیوتر دستی برای این سبک بازی نداره و بازی براش یه مدل دیگه تعریف شده :

پس دست و بالتونو جمع کنین و بذارین رو کیبورد :دی



اذیت نکن ملت رو :|



خب حالا دوباره به کد برگردین و سعی کنین توضیح بدین که کامپیوتر قراره با شما چه طور بازی کنه. یه نکته ی خوب اینه که دیگه لازم نیست تعداد دفعات برنده شدن تون رو بشمرین؛ چراکه حریف هوشمندتون بدون اشتباه این کار رو براتون انجام می ده. فقط کافیه سنگ کاغذ یا قیچی رو با انتخاب یک عدد، مشخص کنین.

حریف هم از تابع رندوم برای این انتخاب استفاده می کنه.



قبل از بازی کد رو کامل کنید تا کامپیوتر (این رقیب و داور شرافتمند) با هر دور بازی، امتیاز برنده رو افزایش بده.



امیدواریم که از این بازی لذت برده باشین. تا کارگاه بعدی خدا نگهدار همگی *_*

;