به نام خدا

دانشکده‌ی مهندسی برق و کامپیوتر دانشکده فنی دانشگاه تهران

# مبانی کامپیوتر و برنامه‌نویسی

| | | |
|---|---|---|
| نیمسال اول 1401-1402 | دستور کار آزمایشگاه شماره 3<br>آشنایی با ساختارهای کنترلی | استاد : دکتر مرادی، دکتر هاشمی |

## 1. انجام دهید! (پیاده سازی دستورات شرطی به کمک if و else)

1. برنامه ی زیر را بخوانید. به نظر شما چه اتفاقی خواهد افتاد؟
2. حال آن را در محیط VS code نوشته و آن را کامپایل کرده و اجرا نمایید. آیا این برنامه همان طور که انتظار داشتید عمل می کند؟ چه اتفاقی افتاد؟

```
#include<stdio.h>
int main() {
	int input;
	scanf("%d", &input);
	if (input = 5){
		printf("Your number was 5\n", input);
	}
	else if (input > 5) {
		printf("Your number was greater than 5\n");
	}
	else {
		printf("Your number was less than 5!\n");
	}
	return 0;
}
```

3. در برنامه فوق از دستور scanf مقدار ورودی خوانده شده(متغیر input) را، با دستور printf نمایش دهید، ببینید که چرا این برنامه مخالف انتظارتان پاسخ می دهد؟

4. برنامه فوق را تصحیح کنید به صورتی که در ازای ورودی‌های مختلف خروجی مناسب را چاپ نماید.

**توجه:** گذاشتن '{' و '}' برای نوشتن قطعه کد مربوط به دستور if و else در برنامه بالا اجباری نیست. ولی در حالتی که بخواهیم بیش از یک دستور در بخش if و یا else اجرا شوند، این مورد اجباری می شود.

قسمت1 : علت را برای دستیاران آموزشی توضیح دهید.

پیش از انجام قسمت بعدی، ابتدا ساختار کلی حلقه های for و while را مرور می کنیم:

1. ساختار کلی while به صورت زیر است:

```
while (/*condition*/){
        /*loop_body*/
}
```

2. ساختار کلی for به صورت زیر است:

```
for (/*initialization;*/ /* condition;*/ /*update*/){
        // loop_body
}
```

همچنین هر حلقه ی for را می توان با یک حلقه ی while پیاده سازی کرد و برعکس. به عبارت دیگر قطعه کد های زیر معادل یکدیگر هستند:

```
for (/*initialization;*/ /* condition;*/ /*update*/){
        // loop_body
}
```

```
// initialization;
while (/*condition*/) {
        // loop_body
        // update;
}
```

## 2. انجام دهید!

می خواهیم برنامه ای بنویسیم که عدد n را از کاربر گرفته جدول ضربی به اندازه‌ی n را در خروجی چاپ کند.

1) ضمن دادن یک پیغام مناسب به کاربر عدد n را از او دریافت کنید.
2) برنامه را باید به کمک حلقه پیاده سازی کنید.

**راهنمایی:** از شبه کد زیر کمک بگیرید:

```
MAIN ()
1     let n and m be dimensions of the output
2     Get n and m from the user
3     let i and j be two distinct integers
4     for i = 1 to n
5           for j = 1 to m
6                 print(i * j)
7                 print(“ ”)
```

3) برای مثال خروجی شما به ازای n = 5 و m = 6 به صورت زیر خواهد بود:

```
5 6
1 2 3 4 5 6
2 4 6 8 10 12
3 6 9 12 15 18
4 8 12 16 20 24
5 10 15 20 25 30
```

قسمت 2 : حال نتیجه را به دستیاران آموزشی نشان دهید.

## 3. انجام دهید! (شابلون)

حال سعی کنید برنامه ی قسمت قبل را به گونه ای تغییر دهید که یک مثلث متساوی الساقین مطابق شکل صفحه بعد رسم کند.

1) ضمن دادن یک پیغام مناسب به کاربر عدد m را از او دریافت کنید.
2) برنامه را باید به کمک حلقه پیاده سازی کنید.
3) برای مثال خروجی شما به ازای m های متفاوت به صورت زیر است:

```
3
-*-
***

5
--*--
-***-
*****
```

```
7
---*---
--***--
-*****-
*******

11
-----*-----
----***----
---*****---
--*******--
-*********-
***********

13
------*------
-----***-----
----*****----
---*******---
--*********--
-***********-
*************
```

قسمت 3 : حال نتیجه را به دستیاران آموزشی نشان دهید.

## 4. انجام دهید! (ترکیب دو برنامه ی بالا)

حال می خواهیم با ترکیب دو برنامه ی بالا یک برنامه ی بزرگ تر بسازیم. این برنامه قرار است یک منو داشته باشد. بدین گونه که کاربر بتواند بین دو شکل مربع و مثلث یکی را انتخاب کرده و برنامه ما شکل مورد نظر را برای او رسم کند و مجدداً با دادن پیغامی مناسب منتظر درخواست جدیدی از کاربر بماند. لازم به ذکر است که می بایست بجای استفاده از دستورات if و else، از switch استفاده کنید.

برنامه ی شما باید کارهای زیر را انجام دهد:

1) ابتدا پیغام مناسبی به کاربر بدهید که از بین جدول‌ضرب و مثلث یکی را انتخاب کند و یا برنامه را خاتمه دهد. فرض کنید کاربر برای جدول‌ضرب عدد 1 و برای مثلث عدد 2 وارد می‌کند. در صورتی هم که بخواهد برنامه خاتمه یابد، عدد 0 را تایپ می‌کند. هر عددی که غیر از این سه عدد وارد شد، باید به کاربر پیغام خطا داده و مجدداً از او درخواست ورودی کنید. (یعنی مجدداً مرحله 1 را تکرار کنید.)
2) ضمن دادن پیغام مناسبی به کاربر، گزینه‌ی را که انتخاب کرده اعلام کنید و از او در خواست کنید که ورودی مناسب هر یک را وارد نماید.

3) به ازای انتخاب جدول ضرب، هیچ یک از اعداد m و n نباید کوچکتر از 1 باشند. اگر چنین بود، ضمن دادن پیغام خطا به کاربر مجدداً به مرحله ی 1 بر گردید.
4) به ازای انتخاب مثلث، عدد m از کاربر دریافت شود. چنانچه m کوچکتر از 3 باشد، لازم است تا پیام خطا بدهید و به مرحله‌ی 1 باز گردید.

**راهنمایی 1:** برای این که برنامه شما تا زمانی که کابر درخواست خاتمه آن را نکند، ادامه یابد می توانید از عبارت while (true) استفاده کنید و کل برنامه تان را درون بدنه ی حلقه ی while بنویسید. و اگر کاربر درخواست خاتمه داد با استفاده از روش زیر از حلقه while خارج شوید. (در این صورت با فرض این که کل برنامه تان درون while باشد، تنها کدی که پس از بیرون آمدن از حلقه ی while اجرا خواهد شد بازگشت از تابع main خواهد بود.)

```
int quit = 1, x = -1;
while(quit){
    scanf("%d", &x);
    switch(x){
            case 0:
                    quit = 0;
                    break;
    }
}
```

**راهنمایی 2:** تا به حال فرض ما بر این بوده است که کاربر هیچ کدام از ورودی ها را اشتباه وارد نمی کند. در این برنامه می خواهیم ورودی های اشتباه را نیز در نظر بگیریم. برای این کار ضمن دادن پیغامی مناسب به کاربر، در ساختاری مشابه ساختار بالا، در بدنه case مورد نظرمان، جز عبارت break چیزی نمی‌گذاریم. در این صورت سیستم کاری انجام نخواهد داد و مجدد به ابتدای حلقه‌ی while باز می‌گردد.

قسمت 4 : حال نتیجه را به دستیاران آموزشی نشان دهید.

## 5. بازی حدس عدد(امتیازی)

با استفاده از ساختار if و else که در قسمت‌های پیشین فرا گرفتید، می‌خواهیم تا در این قسمت با اقتباس از الگوریتم مشهوری به نام جستجوی دودویی(binary search) عددی تصادفی را که بین 1 تا 100 انتخاب شده است، پیدا کنیم. روش تولید یک عدد تصادفی بین 1 تا 100 در زبان c به شکل زیر است که در آن، متغیّر seed را با شماره‌ی دانشجویی خود مقدار بدهید:

```
#include <stdlib.h>

int seed = 810193570;
```

شما می توانید هر مقدار دلخواهی را به متغیّر تعریف شده بدهید. اینجا به دلیل اینکه نتیجه هر فرد حتی‌الامکان متفاوت */ بشود، بهتر است که هر دانشجو شماره‌ی دانشجویی خودش را به عوان مقدار به این متغیر بدهد. /*

```
srand(seed);

int random_number = rand() % 100;
```

در ادامه، در یک حلقه‌ی while(true)، ساختارهای کنترلی‌ای پیاده سازی کنید که رفتارهای زیر را داشته باشد:

- ابتدا از کاربر بخواهد که یک عدد بین 1 تا 100 حدس بزند.
- چنانچه عدد وارد شده برابر با عدد تصادفی تولید شده بود، حلقه‌ی while(true) به اتمام برسد.
- چنانچه عدد وارد شده از عدد تصادفی تولید شده کوجکتر بود، پیامی کوتاه به کاربر بدهد که عدد او از عدد تصادفی کوچکتر است.
- در صورت بزرگتر بودن عدد وارد شده از عدد تصادفی تولید شده، پیامی دیگر داده شود تا کاربر از این موضوع آگاه گردد.

چگونه کاربر می تواند در سریعترین زمان عدد تصادفی تولید شده را حدس بزند؟

قسمت 5 : نتیجه را به دستیاران آموزشی نشان دهید.

❋ **توضیحات بیشتر:** تولید اعداد تصادفی از مسائل مهم حوزه‌ی علوم کامپیوتر است؛ و مهمترین کاربرد آنها در حوزه‌ی امنیّت شبکه به چشم می‌خورد. از آنجا که در کامپیوتر تمام کارها باید دقیقا مشخص باشند، لذا مجبوریم تا بجای انجام کارهایی مانند پرتاب سکه برای تولید کاملا تصادفی هر بیت از یک رشته دودویی، از توابعی استفاده کنیم که رفتارهایی شبه تصادفی دارند. چنانچه از تعریف یک تابع مشخص است، لازم است تا ضابطه‌ای داشته باشد و داشتن چنین خاصیتی، یعنی تولید اعداد به صورت کاملا تصادفی صورت نمی‌گیرد. چراکه در هر حال پس از گذشت زمان و تولید اعداد شبه تصادفی، به دلیل داشتن حالت‌های محدود(هر چقدر هم زیاد باشند، بی‌نهایت نخواهند بود) تابع تولید کننده آنها شروع به تولید اعداد تکراری خواهد کرد. به عبارت دیگر، هر تابع تولید کننده اعداد شبه تصادفی، یک دوره‌ی تناوب دارد. لذا یک راهکار برای این مشکل، افزایش دوره‌ی تناوب تابع تولید کننده‌ی اعداد تصادفی است. در کد شما، تابعی از زبان c استفاده شده است که با کمک گرفتن از عددی به نام seed در محاسبات خود، رشته‌ای از اعداد تصادفی را تولید می‌کند. حال اگر تعداد بیت‌های seed زیاد باشد، جستجو در فضای آن مشکل‌تر خواهد بود و لذا نمی‌توان با دانستن ضابطه‌ی تابع تولید کننده‌ی اعداد شبه‌تصادفی، seed ای را یافت که رشته‌ای از اعداد را تولید کند که نهایتا به عدد شبه تصادفی در دست ما برسد.[1]

**موفق باشید**

[1] برگرفته از اسلایدهای درس امنیت شبکه جناب آقای دکتر بهنام بهرک، پاییز 96