



« به نام راستگوی بی همتا »

مبانی کامپیوتر و برنامه سازی

دستور کار آزمایشگاه شماره 2



زمستان 98

آشنایی با ساختارهای کنترلی

دکتر مرادی

امروز قرار است استفاده از ساختارهای مختلف کنترلی در زبان C و تحلیل روند های برنامه با استفاده از حلقه ها و شرط ها را بررسی کنیم.

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 Visual Studio 2012 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 بر روی دستور scanf یک breakpoint بگذارید و از آن جا به بعد برنامه را خط به خط اجرا نمایید. (این کار را توسط امکان step over نرم افزار VS2012 می توانید انجام دهید که در خودآموز این نرم افزار مفصلاً توضیح داده شده است). 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.

یافته های خود را در کادر زیر بنویسید.

به علت این که در if اول عبارت به صورت logical نیامده است برابری با 5 چک نمی شود و هر بار عدد 5 در input ریخته شده و چون موفقیت آمیز است وارد همان if شده و وارد بقیه if ها نمی شود در واقع انگار اصلا input وارد شده به کار نمی آید.

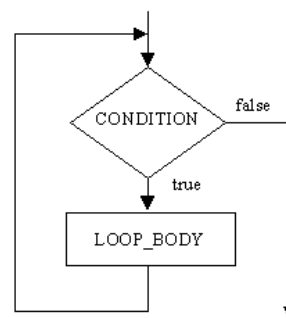
```
#include <stdio.h>
int main() {
    int input;
    scanf("%d", &input);
    if (input == 5){
        printf("Your number was 5\n", input);
    }
    else if (input > 5) {
        printf("Your number was greater than 5\n");
    }
    else {
        printf("Your number was less than 5!\n");
    }
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

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.

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:

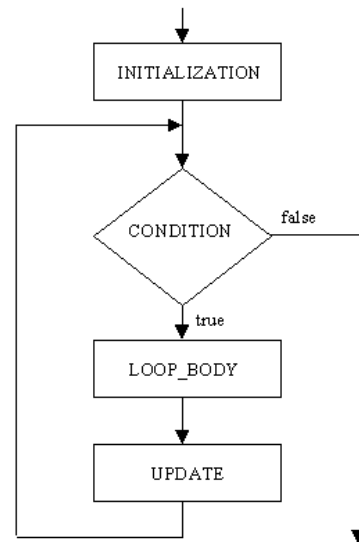
while (condition)
loop_body



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! (شابلون)

می خواهیم برنامه ای بنویسیم که عدد n را از کاربر گرفته و مربعی به ضلع 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 استفاده کنید:

```
READ n
FOR i = 0 to n
  FOR j = 0 to n
    PRINT '*'
    PRINT ' '
  ENDFOR
  PRINT '\n'
ENDFOR
```

3) برای مثال خروجی شما به ازای $n = 7$ به صورت زیر خواهد بود:



The screenshot shows a Windows command prompt window titled "C:\windows\system32\cmd.exe". The prompt is at the top left, followed by a question mark. Below the question mark is a 7x7 grid of asterisks. At the bottom of the window, it says "Press any key to continue . . .".

حال نتیجه را در کادر زیر بنویسید.

```
#include <stdio.h>
int main() {
    int num = 0;
    printf("Please Enter N Here: ");
    scanf("%d", &num);
    for (int i = 0; i < num; i++){
        for (int j = 0; j < num; j++){
            printf("* ");
        }
        printf("\n");
    }
}
```

3. انجام دهید! (یک شابلون دیگر)

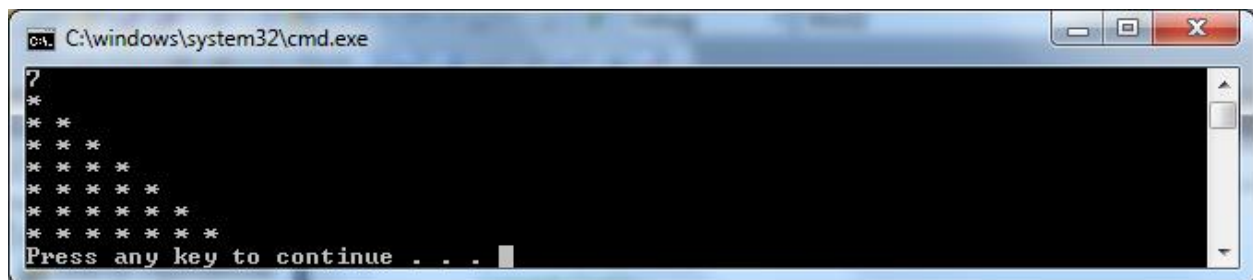
حال سعی کنید برنامه ی قسمت قبل را به گونه ای تغییر دهید که یک مثلث قائم الزاویه متساوی الساقین با طول ساق n رسم کند.

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: سعی کنید ابتدا الگوریتم خود را در قالب شبه کد (یا فلوچارت) ارائه دهید و سپس کد زدن را شروع کنید. (می توانید از شبه کد سوال قبل ایده بگیرید!)

3) برای مثال خروجی شما به ازای $n = 7$ به صورت زیر است:



```
C:\windows\system32\cmd.exe
?
*
* *
* * *
* * * *
* * * * *
* * * * *
* * * * *
* * * * *
Press any key to continue . . .
```

حال نتیجه را در کادر زیر بنویسید.

```
#include <stdio.h>
int main() {
    int num = 0;
    printf("Please Enter N Here: ");
    scanf("%d", &num);
    for (int i = 0; i < num; i++){
        for (int j = 0; j <= i; j++){
            printf("* ");
        }
        printf("\n");
    }
}
```

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.

برنامه ی شما باید کارهای زیر را انجام دهد:

- 1) ابتدا پیغام مناسبی به کاربر بدهید که از بین مربع و مثلث یکی را انتخاب کند و یا برنامه را خاتمه دهد. فرض کنید کاربر برای مربع عدد 1 و برای مثلث عدد 2 وارد می کند. در صورتی هم که بخواهد برنامه خاتمه یابد عدد 0 را وارد می کند. 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) سپس ضمن دادن پیغام مناسبی به کاربر شکلی را که انتخاب کرده اعلام کنید و از او در خواست کنید که عدد n را وارد نماید.
- 3) اگر مقدار n ورودی کوچک تر از 4 بود، ضمن دادن پیغام خطا به کاربر مجدداً به مرحله ی 1 برگردید.
- 4) با توجه به مقدار n شکل مورد نظر را چاپ کنید و به مرحله 1 برگردید.

راهنمایی 1: برای این که برنامه شما تا زمانی که کاربر 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 دستور **break** 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).

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، با استفاده از دستور **continue** مجدداً به ابتدای حلقه ی `while` باز می گردیم.

برای درک بهتر دو راهنمایی بالا قطعه کد زیر را نگاه کنید. ساختار کد زیر را تغییر ندهید. تنها کافی است که قسمت های کامنت را با کد گفته شده جایگزین کنید. کد زیر در فایل همراه دستور کار برای شما آماده شده است.

```

#include <stdio.h>
int main() {
    int command, n;
    while (true) {
        // پیغام مناسبی بدهید که کاربر با وارد کردن عدد 0 یا 1 یا 2 یکی از 3 امکان داده شده را انتخاب نماید
        // حال عدد command را از ورودی بگیرید
        if (command == 1)
            // پیغام مربع انتخاب شده را نمایش دهید
        else if (command == 2)
            // پیغام مثلث انتخاب شده را نمایش دهید
        else if (!command) {
            // پیغام پایان برنامه را نمایش دهید
            // یک دستور اضافه کنید تا از حلقه خارج شویم
        }
        else {
            // پیغام ورودی اشتباه را نمایش دهید و از کاربر درخواست کنید مجدداً تلاش کند
            // یک دستور اضافه کنید که به ابتدای حلقه بازگردیم
        }
        // ضمن دادن پیغامی از کاربر درخواست کنید که عدد n را وارد کند
        // عدد n را از کاربر بگیرید
        if (n < 4) {
            // پیغام ورودی اشتباه را نمایش دهید و از کاربر درخواست کنید مجدداً تلاش کند
            // یک دستور اضافه کنید که به ابتدای حلقه بازگردیم
        }
        if (command == 1) {
            // کد مربوط به رسم مربع
        }
        else if (command == 2) {
            // کد مربوط به رسم مثلث
        }
    }
    return 0;
}

```

به یک نمونه از اجرای برنامه توجه کنید:

```
C:\windows\system32\cmd.exe

Which shape do you want? Enter one of the following numbers:
0 for terminating the program
1 for drawing a square
2 for drawing a triangle
1
You want to draw a square!
give me n please:
4
* * * *
* * * *
* * * *
* * * *
Which shape do you want? Enter one of the following numbers:
0 for terminating the program
1 for drawing a square
2 for drawing a triangle
2
You want to draw a triangle!
give me n please:
5
*
* *
* * *
* * * *
* * * * *
Which shape do you want? Enter one of the following numbers:
0 for terminating the program
1 for drawing a square
2 for drawing a triangle
1
You want to draw a square!
give me n please:
3
The n input was lower than 4 (not in correct range). Please try again.
Which shape do you want? Enter one of the following numbers:
0 for terminating the program
1 for drawing a square
2 for drawing a triangle
4
You have entered a wrong number. Please try again.
Which shape do you want? Enter one of the following numbers:
0 for terminating the program
1 for drawing a square
2 for drawing a triangle
0
Hope to see you again.
Press any key to continue . . .
```

حال نتیجه را در کادر زیر بنویسید.


```

#include <stdio.h>
int main() {
    int menu_command = -1, size_command = -1;
    while (1){
        printf("Which Shape Do You Want? Enter One of The Following Numbers: \n");
        printf("0 for terminating the program\n");
        printf("1 for drawing a square\n");
        printf("2 for drawing a triangle\n");
        scanf("%d", &menu_command);
        if (menu_command != 0 && menu_command != 1 && menu_command != 2){
            printf("You have entered a wrong number. Please try again.\n");
        }
        else{
            if (menu_command == 0){
                printf("Hope to see you again.\n");
                break;
            }
            else if (menu_command == 1){
                printf("You want to draw a square!\n");
                printf("Give me n please: \n");
                scanf("%d", &size_command);
                if (size_command < 4){
                    printf("The n input was lower than 4 <not in the correct range>. Please try again.\n");
                }
                else{
                    for (int i = 0; i < size_command; i++){
                        for (int j = 0; j < size_command; j++){
                            printf("* ");
                        }
                        printf("\n");
                    }
                }
            }
            else if (menu_command == 2){
                printf("You want to draw a triangle!\n");
                printf("Give me n please: \n");
                scanf("%d", &size_command);
                if (size_command < 4){
                    printf("The n input was lower than 4 <not in the correct range>. Please try again.\n");
                }
                else{
                    for (int i = 0; i < size_command; i++){
                        for (int j = 0; j <= i; j++){
                            printf("* ");
                        }
                        printf("\n");
                    }
                }
            }
        }
    }
}

```

5. انجام دهید! (بازی حدس عدد) ←

با قرار دادن کدی که در قسمت اول آزمایش تصحیح نمودید در یک حلقه `while(true)` برنامه ای بنویسید که با تعیین یک عدد تصادفی بین 0 تا 100 از کاربر بخواهد تا آن عدد را پیدا کند. در صورتی که کاربر پس از چند تلاش توانست عدد مورد نظر را پیدا کند برنامه با چاپ پیام مناسب خاتمه پیدا کند. برای تولید عدد تصادفی از کد زیر استفاده کنید.

```
#include <stdlib.h>
```

```
.  
.
```

```
srand(your_student_number);
```

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

سپس به جای عدد 5 در کد اولیه متغیر `random_number` را قرار دهید.

نتیجه را در کادر زیر بنویسید.

```
#include <stdio.h>
#include <stdlib.h>

int main(){
    srand(810196688);
    int random_number = rand() % 100, guess = -1;

    printf("Welcome to the Game!\n");
    while (1){
        printf("Enter Your Guess:\n");
        scanf("%d", &guess);

        if (guess == random_number){
            printf("Congratulations! You have Gussed The Correct Number.\n");
            break;
        }
        else if (guess > random_number) {
            printf("Your guess was greater than the number in my mind!\n");
        }
        else {
            printf("Your guess was less than the number in my mind!\n");
        }
    }
    return 0;
}
```

موفق باشید