



مبانی کامپیوتر و برنامه سازی

فصل هفتم: تابع

انگیزه

- در بسیاری از مواقع انجام یک دنباله از دستورات، به کرات مورد نیاز است. برای نمونه، تا به حال به دفعات از دستور printf استفاده نموده‌ایم. یعنی، از دنباله دستوراتی که باعث نمایش می‌شوند استفاده نموده‌ایم. یا محاسبه قدر مطلق یک عدد در پر کاربرد است. بنابراین یک برنامه نوشته شده تعداد زیادی رونوشت از دستورات زیر دارد.

```
if (x<0)
    y=-x;
else
    y=x;
```

- برای پرهیز از چنین تکرارهای می‌توان به آنها نامی خاص داد و با فراخواندن نام، از آنها استفاده نمود. **به این قطعات برنامه که نام ویژه‌ای دارند تابع می‌گویند.**

- (تکرار قطعه کد فوق باعث ایجاد اشکال نمی‌شود، اما خوانایی و کارایی را پایین می‌آورد.)

```
for(i=2, R1=1; i<=X; i++)
    R1*=i;
```

فاکتوریل X را محاسبه کن

فلوچارت روبرو حاصل $X!/Y!$ را محاسبه می‌کند. برای محاسبه فاکتوریل X به یک حلقه تکرار نیاز است؛ برای محاسبه فاکتوریل Y همان حلقه را باید بازنویسی نمود. بنابراین چند خط دستور مشترک در این برنامه وجود دارد. با استفاده از توابع می‌توان از این تکرارها پرهیز نمود.

چندین خط برنامه

فاکتوریل Y را محاسبه کن

```
for(i=2, R2=1; i<=Y; i++)
    R2*=i;
```

حاصل فاکتوریل X تقسیم بر فاکتوریل Y را نمایش بده

تعریف تابع

- تابع، یک قطعه برنامه است که کار معینی را انجام می‌دهد و با استفاده از آن می‌توان از برنامه نویسی تکراری پیش‌گیری نمود.
- البته تعریف تابع در زبان C مشابه تعریف آن در ریاضیات است. یک تابع در ریاضیات با انجام محاسبات روی ورودی‌های خود یک خروجی تولید می‌کند. برای نمونه تابع $\sin(x)$ عدد x را به عنوان ورودی، گرفته محاسبه‌ای را انجام داده و یک عدد دیگر را باز می‌گرداند. با کمی دقت می‌توان دریافت که برای استفاده از توابع سه جزء ورودی، نام تابع، و خروجی آن باید مشخص شوند.

هر برنامه در زبان C، مجموعه‌ای از یک یا چندین تابع است.
تابع `main` نقطه شروع برنامه است.

نوشتن اولین تابع در زبان C

- برای استفاده از یک تابع در زبان C باید نوع ورودی‌ها، نوع خروجی، نام تابع، و برنامه آن معین شود. در اینجا تابع قدر مطلق را تعریف می‌کنیم:

```
int Abs(int x){  
    if (x<0)  
        return -x;  
    else  
        return x;  
}
```

شروع تابع

بدنه

پایان تابع

**تمام توابع باید قبل از تابع
main و بعد از #include ها
نوشته شوند.**

- در اینجا نام تابع Abs است. این تابع یک ورودی عددی از نوع صحیح با نام x دارد. خروجی این تابع نیز int است. برنامه مربوط به آن بسیار ساده بوده و با توجه به مقدار ورودی یک خروجی ارائه می‌کند.
- دستور return باعث قرار دادن مقدار داده شده به آن به عنوان خروجی و اتمام اجرای تابع می‌شود.

فراخوانی تابع در زبان C

• برای استفاده از توابع باید آنها را فراخواند. یعنی، باید مقادیر ورودی و متغییری که خروجی تابع در آن قرار گیرد را تعیین نمود. برای استفاده از تابع `Abs` می‌توان در تابع `main` یا هر تابع دیگری نوشت `i=Abs(-4)`؛ این دستور باعث اجرای چند خط برنامه تابع `Abs` می‌شود، در این اجرا ورودی تابع، یعنی `x` مقدار `-4` دارد و حاصل اجرای آن در `i` قرار می‌گیرد. دستور `return` باعث

1. قرار دادن مقدار داده شده به آن به عنوان خروجی و

2. اتمام اجرای تابع

می‌شود.

برنامه‌ای برای محاسبه $10! / (6! * 4!)$

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

```
long Fact(int n){  
    int i, R=1;  
    for(i=2; i<=n; i++)  
        R*=i;  
    return R;  
}
```

```
void main(){  
    long m;  
    m=Fact(10)/(Fact(6)*Fact(4));  
    printf(“%d”, m);  
}
```

در اینجا تابع Fact سه بار فراخوانده شده است، یک بار با $n=10$ بار دیگر با $n=6$ و بار آخر با $n=4$. اگر از تابع Fact استفاده نمی‌شد مجبور بودیم چند خط برنامه آن را سه بار تکرار کنیم (-;

ساختار یک برنامه در زبان C

- `#include <stdio.h>`
- (نام و نوع ورودی) نام تابع نوع برگشتی
{
 - بدنه تابع}
- (نام و نوع ورودی) نام تابع نوع برگشتی
{
 - بدنه تابع}
- `main(){`
 - بدنه تابع}

تمام توابع قبل از `main` تعریف می‌شوند و هر تعداد تابعی می‌توان نوشت.

در بدنه یک تابع می‌توان هر تابعی را فراخواند. برای نمونه می‌توان در تابع `Fact` از تابع `Abs` استفاده نمود.

تابعی برای تشخیص اول بودن اعداد

```
#include <stdio.h>

long IsPrime(int n){
    int i;
    for(i=2; i<=n/2; i++)
        if(n%i==0)
            return 0;
    return 1;
}

void main(){
    printf("%d", IsPrime(10));
    printf("%d", IsPrime(21));
    printf("%d", IsPrime(103));
}
```

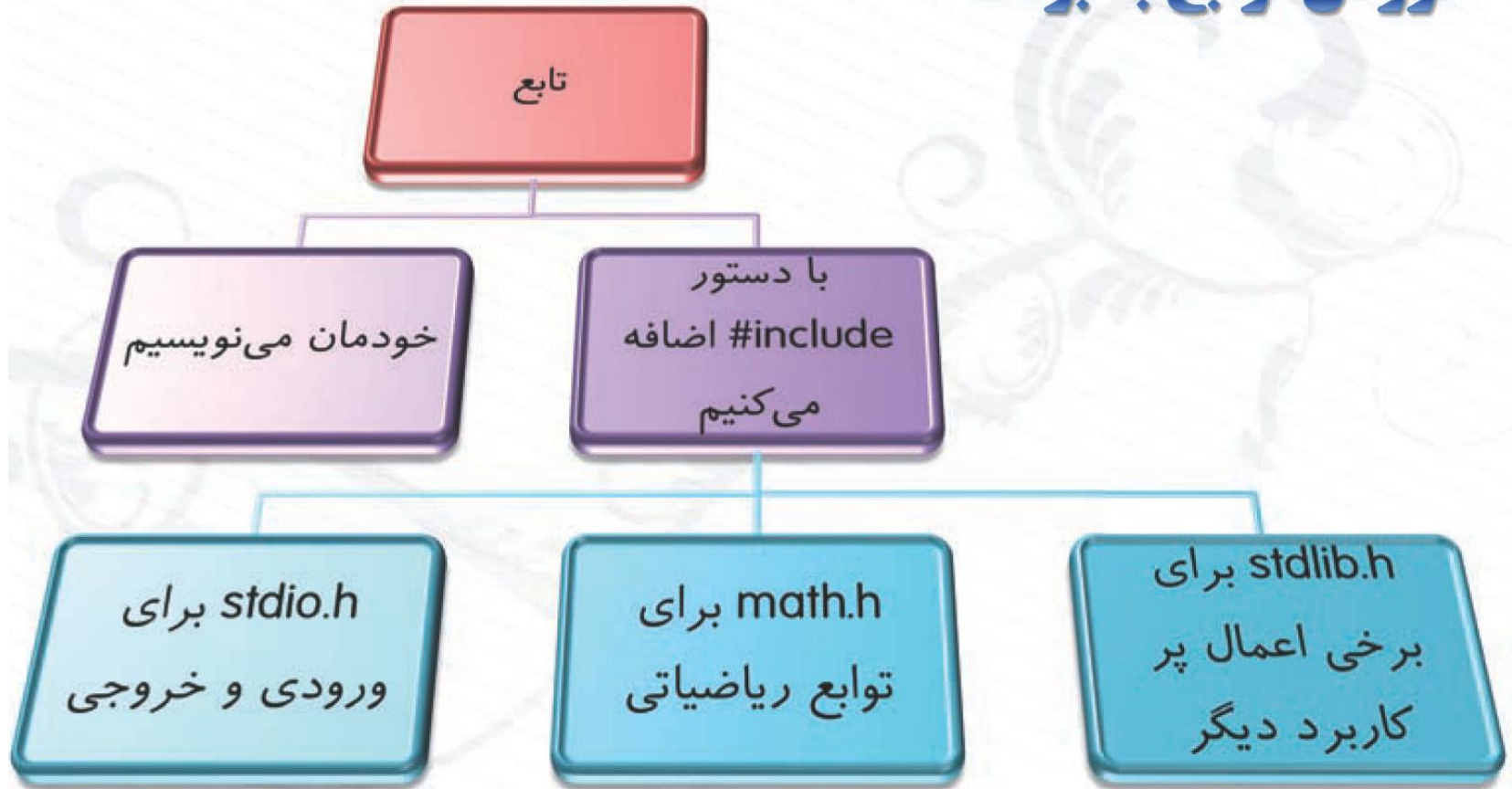
تابع با استفاده از یک حلقه for بخش پذیری تمام اعداد بین 2 تا $n/2$ بر n را بررسی می‌کند.

می‌خواهیم تابع IsPrime به ازای اعداد اول 1 بازگرداند. همین که n بر عددی تقسیم شود دستور return باعث بازگرداندن 0 شده و تابع پایان می‌پذیرد.

آیا باید دوباره چرخ را اختراع نماییم؟

- آیا باید هر تابعی که نیاز شد را شخصاً بنویسیم؟ آیا باید خودمان تابعی برای به توان رساندن اعداد بنویسیم؟
- پاسخ پرسش‌های بالا خیر است. ما می‌توانیم در یک برنامه به زبان C از توابعی که دیگران نوشته‌اند استفاده کنیم. تعداد توابعی که شرکت‌های برنامه نویسی تا کنون به زبان C افزوده‌اند چندین هزار است! تا همین لحظه ما از `printf` و `scanf` استفاده کرده‌ایم که توسط شرکت سازنده برای ما ایجاد شده‌اند.
- برای استفاده از توابعی که دیگران نوشته‌اند باید با استفاده از دستور `#include` آن توابع را به برنامه خود اضافه کنیم. دو تابع `printf` و `scanf` در پرونده سرآمد (header file) با نام `stdio.h` نوشته شده‌اند.

افزودن توابع به برنامه



تعداد ورودی و خروجی توابع

- یک تابع می‌تواند هر تعداد ورودی داشته باشد. حتی یک تابع می‌تواند ورودی نداشته باشد! نمونه:

```
#include <stdio.h>
float Max(float x, float y){
    if(x>y)
        return x;
    else
        return y;
}
void main(){
    printf(“%f”, Max(2, 4));
    return;
}
```

تابع Max دو ورودی و خروجی اعشاری دارد.

تابع main خروجی ندارد. بنابراین نوشتن `i=main()` نامعتبر است.

- یک تابع می‌تواند خروجی نداشته باشد! اگر تابعی خروجی نداشته باشد آنگاه دستور **return** داخل آن عددی ندارد. برای معرفی یک تابع بدون خروجی نوع آن را **void** معرفی می‌کنیم.

تابعی برای محاسبه وتر مثلث قائم الزاویه

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

double D(double x, double y){
    return sqrt(pow(x,2)+pow(y,2));
}

void main(){
    printf(“%f”,D(3,4));
}
```

تابع pow ورودی اول را به توان ورودی دوم می‌رساند. $\text{pow}(y, 2)$ مقدار y^2 را باز می‌گرداند.

تابع sqrt جذر ورودی را می‌دهد.

تمام توابع ریاضی در سر آمد math.h تعریف شده‌اند. بنابراین برای استفاده از آنها باید `#include <math.h>` را در ابتدای برنامه آورد.