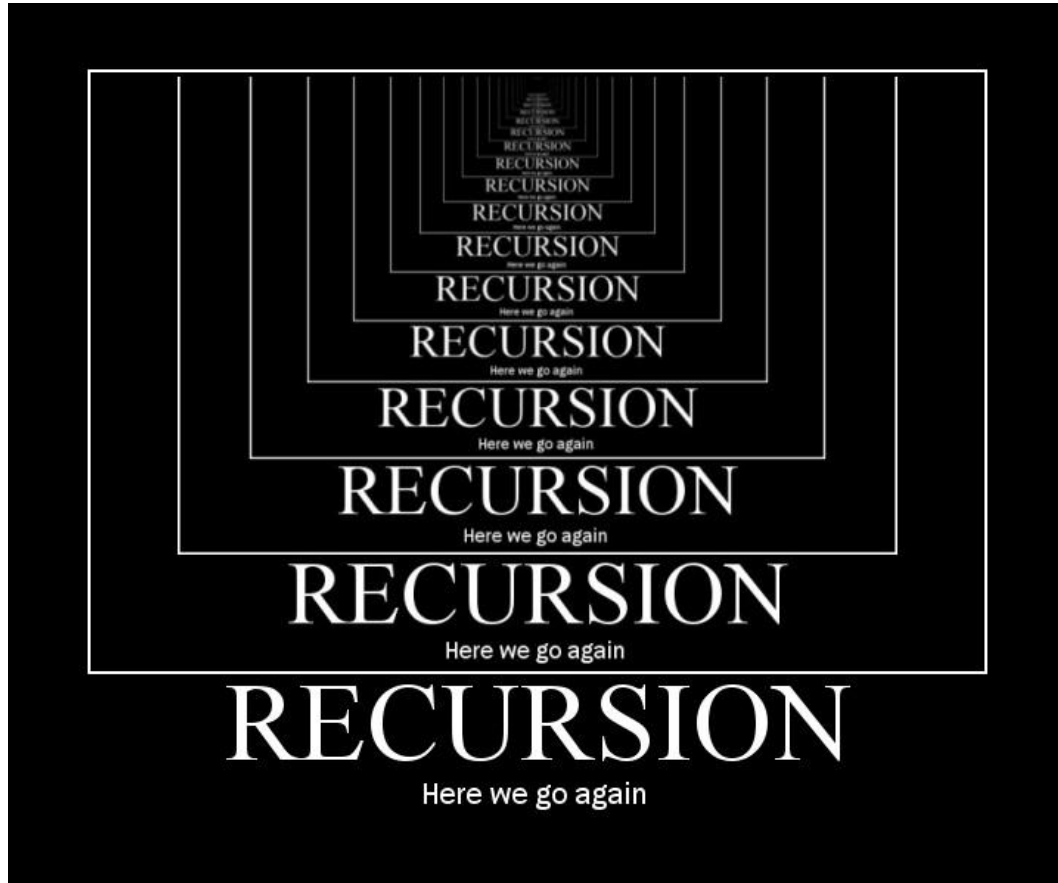


ساختمان‌های داده

Data Structures



توابع
بازگشتی

مقدمه

□ در حین برنامه نویسی زمانی که می خواهیم بخشی از کد که کار مشخصی انجام می دهد را بارها تکرار کنیم، معمولاً به حلقه های for و while فکر می کنیم.

□ این حلقه ها به ما کمک می کنند تا امکان تکرار کاری روی یک لیست یا مجموعه و یا تا رسیدن به شرایط خاصی را داشته باشیم.

□ با این حال، ساختار دیگری برای تکرار یک کار هم وجود دارد که اندکی متفاوت از حلقه ها است.

□ با صدا زدن یک تابع درون خودش، مثلاً برای حل بخش کوچک تری از همان مسئله، ما در واقع داریم عملی بازگشتی انجام می دهیم.

مقدمه

- ❑ توابع بازگشتی خود را آن قدر صدا می زنند تا مسئله حل شود و به نقطه پایانی برسد،
- ❑ به طور مثال مرتب کردن یک آرایه در الگوریتم مرتب سازی ادغامی، که در آن آرایه تا حد رسیدن به تنها یک عنصر به طور مداوم شکسته می شود.
- ❑ یک مثال ساده تر و انسانی تر، عملکرد ما حین خوردن یک پیتزا است.
- شما شروع به تقسیم پیتزا از وسط می کنید و هر دفعه آن را از وسط به دو نیمه تقسیم می کنید.
- در ضمن هر بار هم بررسی می کنید که آیا اندازه فعلی به کوچکی یک لقمه شده است یا نه.

تابع بازگشتی چیست؟

□ تابع بازگشتی: تابعی است که در درون آن، خودش را می صدا میزند و باعث تکرار اجرای کدهای خود شود.

□ به طور کلی می توان گفت که تابع بازگشتی دو بخش اصلی دارد:

- **حالت پایه (پایانی):** در واقع شرطی است که مشخص می کند فراخوانی های تکراری بازگشتی کجا باید خاتمه پیدا کنند.
- **بخش (بازگشتی):** فراخوانی خود.

مزایا و معایب تابع بازگشتی

□ مزایا

- توابع بازگشتی موجب می شوند که کد تمیز و منظم باشد.
- یک وظیفه پیچیده را می توان با استفاده از توابع بازگشتی به زیرمساله های کوچک تر شکست.
- تولید توالی با استفاده از توابع بازگشتی، نسبت به انجام این کار با استفاده از تکرارهای تو در تو (حلقه ها)، آسان تر است.

□ معایب

- گاهی، دنبال کردن منطق نهفته در پشت توابع بازگشتی دشوار است.
- فراخوانی بازگشتی پرهزینه (ناکارآمد) است، زیرا حافظه و زمان زیادی مصرف می کند.
- عیب یابی (دیباگ کردن) توابع بازگشتی دشوار است.

تابع بازگشتی فاکتوریل

$$n! = 1 * 2 * 3 * 4 * \dots * n$$

□ مثال: تابع فاکتوریل

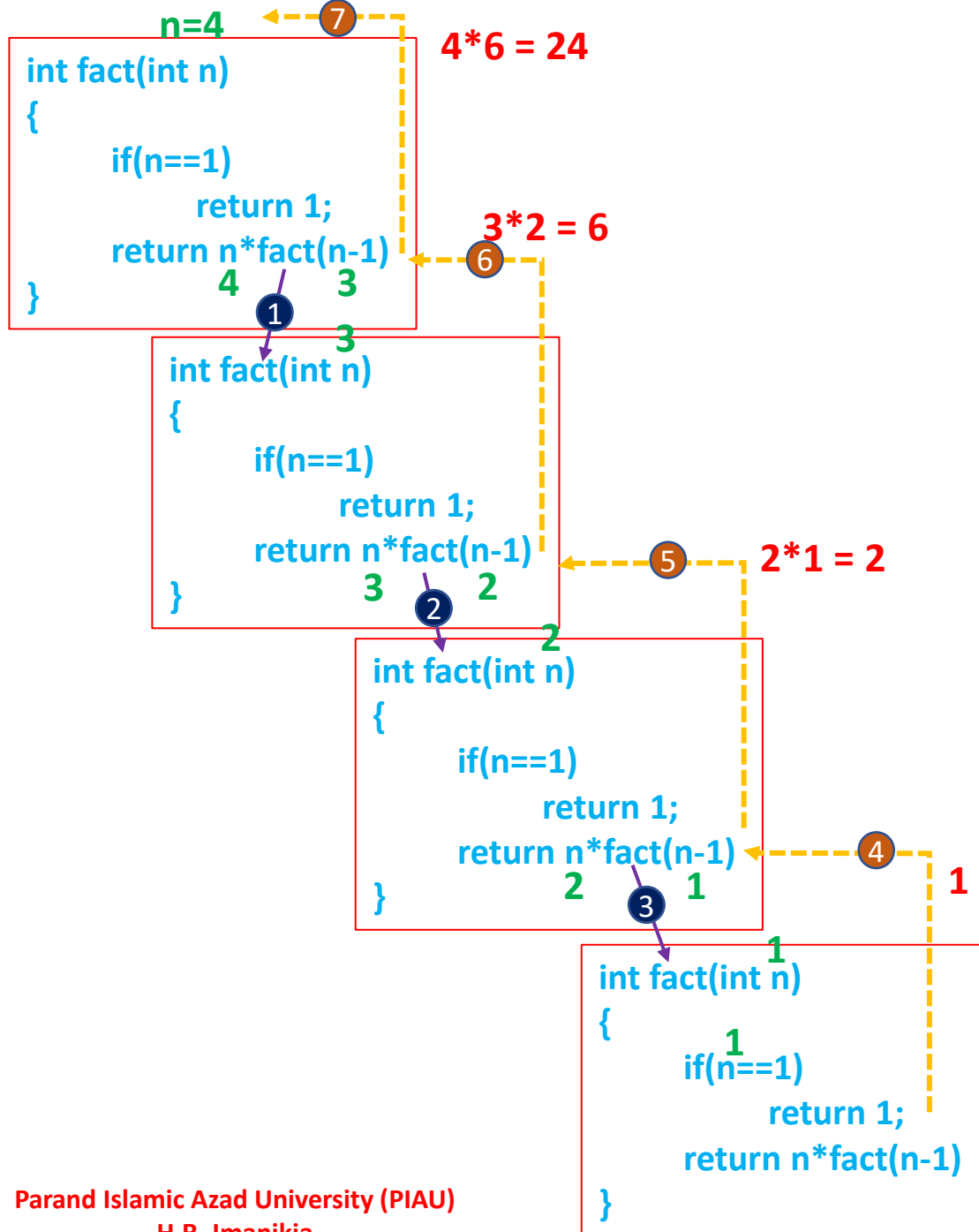
```
int fact(int n)
{
    int p=1;
    for(int i=1; i<=n; i++)
        p = p* i;
    return p;
}
```

(1) بدون تعریف بازگشتی (با حلقه)

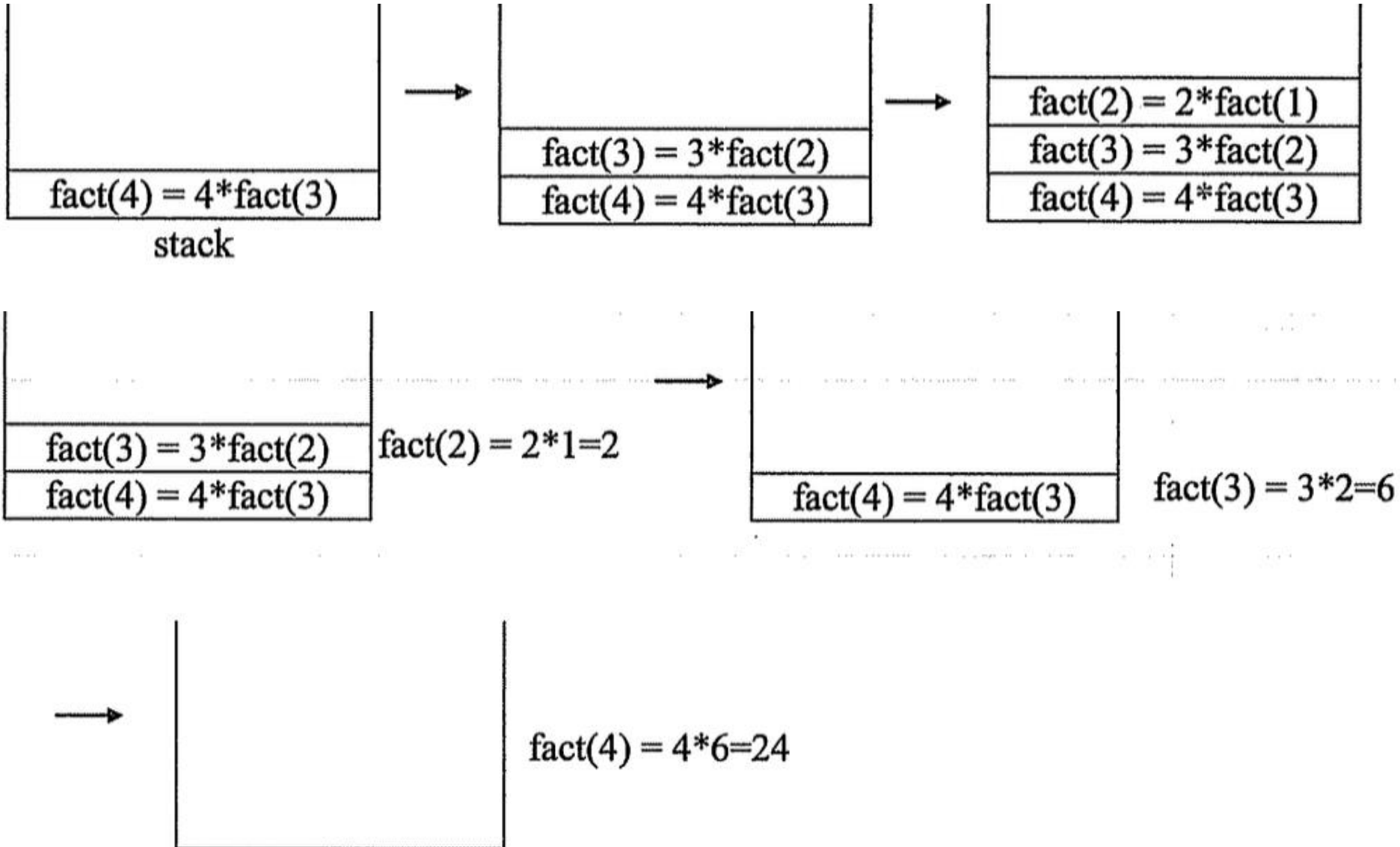
{ $n! = n * (n - 1)!$
 $1! = 1$ بخش بازگشتی
 بخش پایه (پایان)

(2) با تعریف بازگشتی

```
int fact(int n)
{
    if(n==1)
        return 1;
    else
        return n*fact(n-1)
}
```

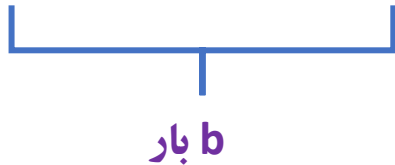


تحليل رفتار تابع بازگشتی با کمک پشته

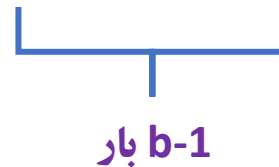


تابع بازگشتی برای محاسبه ضرب

$$a * b = a + a + \dots + a$$



$$a * b = a + a + \dots + a$$



$$a * b = \begin{cases} a & \text{if } b = 1 \\ a + a * (b - 1) & \text{if } b > 1 \end{cases}$$

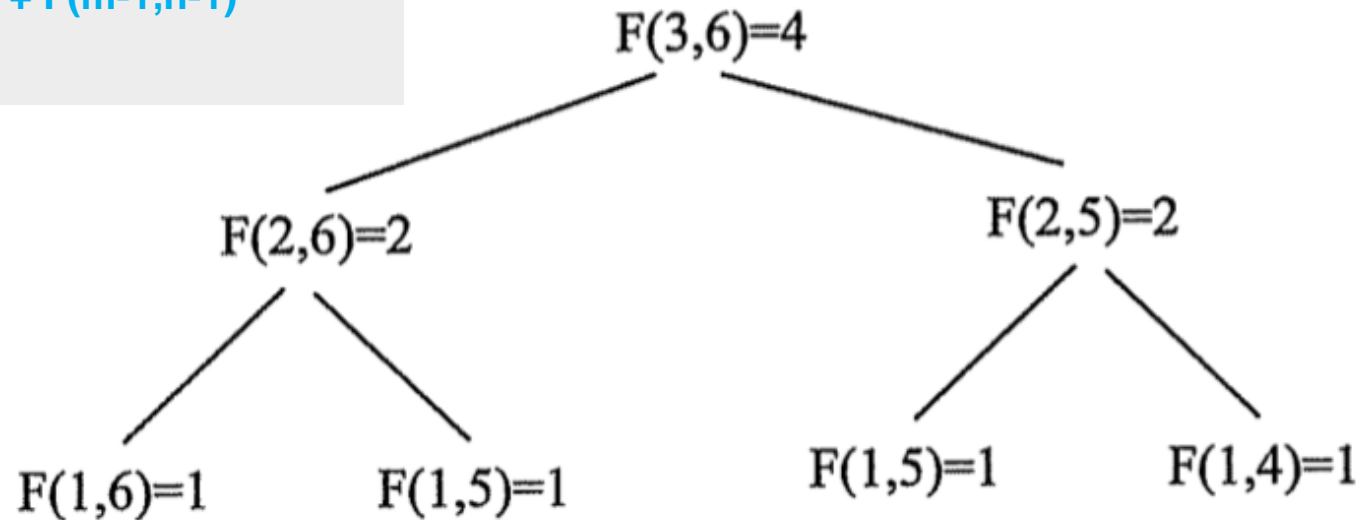
```
int multiply (int a, int b)
{
    if (b == 1) return a;
    return (a + multiply (a, b-1));
}
```

بررسی خروجی یک تابع بازگشتی

```
int F(int m, int n)
{
    if(m==1 || n==0 || m==n)
        return 1;
    else
        return F(m-1,n) + F(m-1,n-1)
}
```

□ در برنامه زیر، خروجی $F(3,6)$ چه می شود؟

• برای اینکار از درخت بازگشت استفاده می نمایم.



تابع بازگشتی سری فیبوناچی

1, 1, 2, 3, 5, 8, 13, 21, ...

□ سری زیر به سری فیبوناچی معروف است.

- هر جمله از جمع دو جمله قبلی خود بدست می آید. (تعریف بازگشتی بر حسب جملات قبلی)
- دو جمله اول ۱ هستند. (پایه)

$$\text{Fib}(n) = \begin{cases} 1 & \text{اگر } n=1 \text{ یا } n=2 \\ \text{Fib}(n-1) + \text{Fib}(n-2) & \text{اگر } n > 2 \end{cases}$$

```
int Fib(int m, int n)
{
    if(n<=2)
        return 1;
    else
        return Fib(n-1) + Fib(n-2)
}
```

تابع بازگشتی سری فیبوناچی

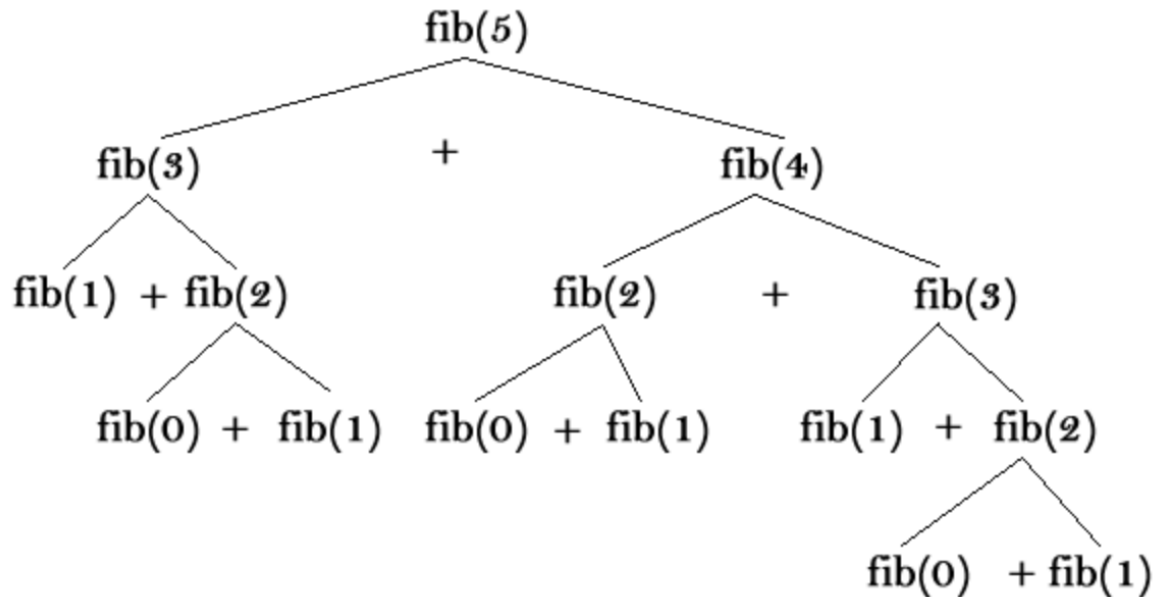
1, 1, 2, 3, 5, 8, 13, 21, ...

$$\text{Fib}(n) = \begin{cases} 1 & \text{if } n=1 \text{ یا } n=2 \\ \text{Fib}(n-1) + \text{Fib}(n-2) & \text{if } n > 2 \end{cases}$$

```
int Fib(int m, int n)
{
    if(n<=2)
        return 1;
    else
        return Fib(n-1) + Fib(n-2)
}
```

□ سری زیر به سری فیبوناچی معروف است.

- هر جمله از جمع دو جمله قبلی خود بدست می آید. (تعریف بازگشتی بر حسب جملات قبلی)
- دو جمله اول ۱ هستند. (پایه)



تمرینات سری دوم

(۱) برنامه و تابعی بنویسید که یک آرایه n عنصری را گرفته و مجموع عناصر آن را چاپ نماید.

- یکبار آن را بصورت غیر بازگشتی

- و بار دیگر بصورت بازگشتی بنویسید.

(۲) برنامه ای بنویسید عدد صحیح مثبتی را گرفته و توسط تابعی هر کدام از ارقام آن را از انتها به ابتدا چاپ نماید.

5836 → 6385

تمرینات سری دوم

۳) برنامه ای بنویسید که به کمک تابعی بازگشتی مقدار $\binom{n}{m}$ را به ازای مقادیر m و n محاسبه کند.

فرمول بازگشتی رابطه ترکیب بصورت زیر است:

$$\binom{n}{m} = \begin{cases} 1 \\ \binom{n-1}{m} + \binom{n-1}{m-1} \end{cases}$$

اگر $m = 0$ یا $m = n$ آنگاه

در غیر اینصورت آنگاه