برنامه نویسی با زبان C
تابع (بخش دوم)

مجتبی اعجمی

استادیار دانشگاه آزاد اسلامی واحد زنجان

مثال : برنامه ای بنویسید که اعداد تام(perfect) بین ۱ تا ۱۰۰۰۰ را چاپ کند

ویکی‌پدیا
دانشنامهٔ آزاد

عدد تام

عدد تام (به انگلیسی: Perfect Number)، عدد صحیح مثبتی است که برابر با مجموع مقسوم‌علیه‌های سرهٔ مثبت خود (همهٔ مقسوم‌علیه‌های مثبتش غیر از خود عدد) باشد. همچنین به‌طور هم ارز، عدد تام، عددی است که نصف مجموع همهٔ مقسوم‌علیه‌های مثبت خود باشد.[۱]

نمونه‌ها

نخستین عدد تام ۶ است؛ زیرا ۱+۲+۳=۶ یا به‌طور هم ارز، ۶=۲/(۱+۲+۳+۶). بعد از آن ۲۸ و بعد از آن به ترتیب ۴۹۶ و ۸۱۲۸ قرار دارند.

Perfect Number	Sum of its Divisors
6	1 + 2 + 3
28	1 + 2 + 4 + 7 + 14
496	1 + 2 + 4 + 8 + 16 + 31 + 62 + 124 + 248
8,128	1 + 2 + 4 + 8 + 16 + 32 + 64 + 127 + 254 + 508 + 1,016 + 2,032 + 4,064

مثال : برنامه ای بنویسید که اعداد تام(perfect) بین ۱ تا ۱۰۰۰۰ را چاپ کند

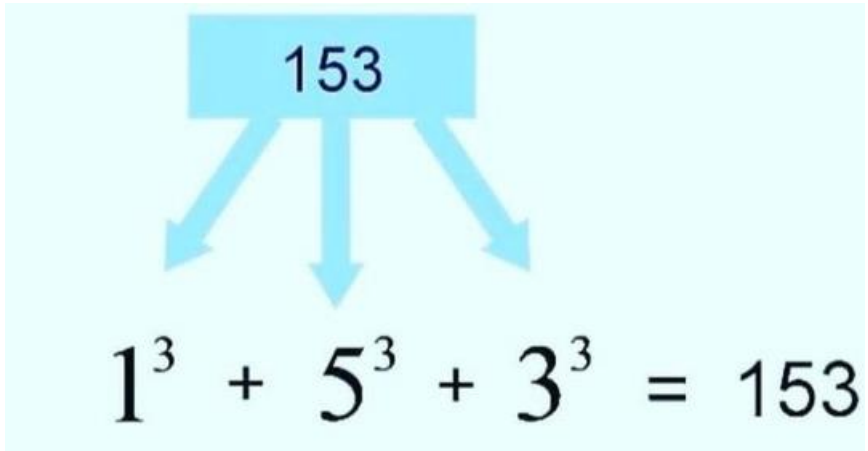
```
1  #include<iostream>
2  using namespace std;
3  int isPerfect( int n)
4  {
5      int sum = 1;
6      for ( int i=2; i<=n/2; i++){
7          if (n%i==0){
8              sum=sum+i;
9          }
10     }
11      if (sum == n )
12          return 1;
13      return 0;
14 }
```

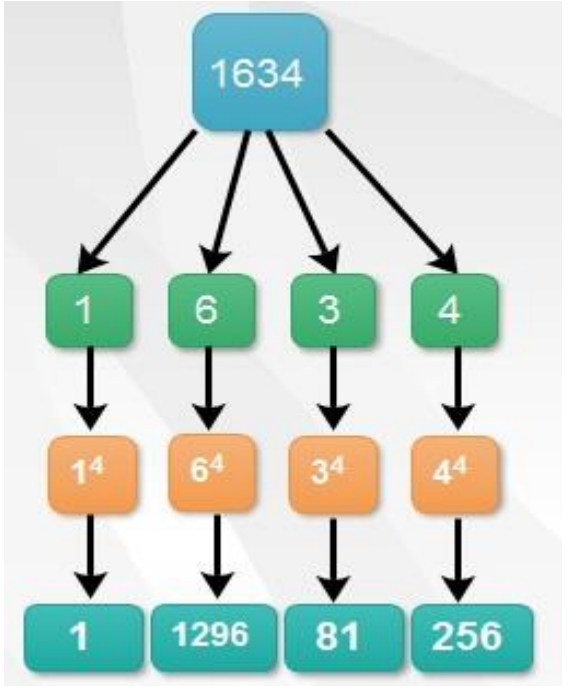
مثال : برنامه ای بنویسید که اعداد تام(perfect) بین ۱ تا ۱۰۰۰۰ را چاپ کند

```
15  int main()
16  {
17      cout << "Below are all perfect numbers till 10000\n";
18      for (int k =6; k<10000; k++)
19          if (isPerfect(k))
20              cout << k << " is a perfect number\n";
21
22      return 0;
23  }
```

اعداد خودشیفته(Narcissistic)

A narcissistic number is a number that is the sum of its own digits each raised to the power of the number of digits.





```cpp
#include<iostream>
using namespace std;
int power(int x,  int y) {
    int p=1;
    for(int i=1;i<=y;i++){
        p = p*x;
    }
    return p;
}
int order(int x) {
    int c = 0;
    while (x) {
        c++;
        x = x / 10;
    }
    return c;
}
```

مثال : برنامه ای بنویسید که اعداد خود شیفته بین ۱ تا ۱۰۰۰ را چاپ کند

```
20  int isArmstrong(int x){
21
22      int n = order(x);
23      int temp = x, sum = 0;
24      while (temp) {
25          int r = temp % 10;
26          sum = sum + power(r, n);
27          temp = temp / 10;
28      }
29
30      if(sum == x){
31          return 1;
32      }
33      else {
34          return 0;
35      }
36  }
```

مثال : برنامه ای بنویسید که اعداد خود شیفته بین ۱ تا ۱۰۰۰ را چاپ کند

مثال : برنامه ای بنویسید که اعداد خود شیفته بین ۱ تا ۱۰۰۰ را چاپ کند

```
39  int main()
40  {
41      cout << "Below are all narcissistic numbers till 1000\n";
42      for (int k =2; k<1000; k++)
43          if (isArmstrong(k))
44              cout << k << " is a narcissistic number\n";
45
46      return 0;
47  }
```

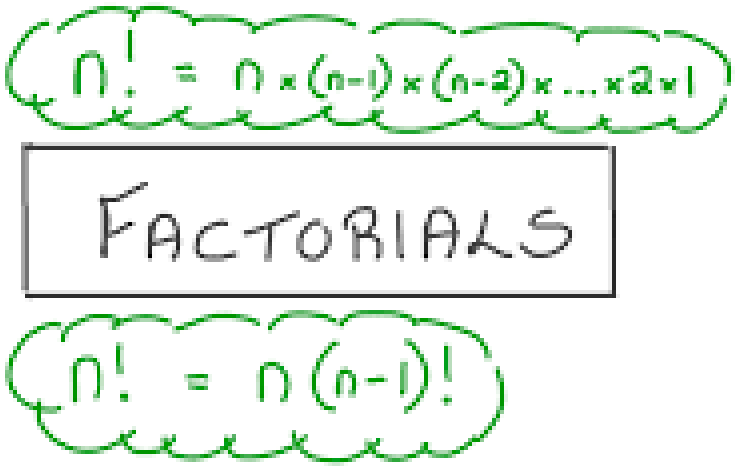
تابع با خروجی void

```
1  #include <iostream>
2  using namespace std;
3
4  void fun()
5  {
6      cout << "Hello";
7
8      return;
9  }
10
11 int main()
12 {
13     fun();
14     return 0;
15 }
```

تابع بازگشتی (recursive function)

- به تابعی بازگشتی می گوییم که در روند اجرا خودش را فراخوانی کند.
- برای نوشتن تابع بازگشتی برای یک مساله باید موارد زیر در نظر گرفته شود:
  - باید برای مسئله حداقل یک *حالت پایه* تعریف کرده و مقدار تابع را در آن حالت بدانیم.
  - روند فراخوانی ها باید به حالت(های) پایه ختم شود و به دور منجر نشود.

مثال تابع بازگشتی: تابع فاکتوریل



$$\text{fact}(n) = \begin{cases} 1 & \text{if } n = 0 \\ n \cdot \text{fact}(n-1) & \text{if } n > 0 \end{cases}$$

مثال تابع بازگشتی: تابع فاکتوریل

```
 4  int fact(int n)
 5  {
 6      if(n==0){
 7          return 1;
 8      }
 9      else {
10          return n*fact(n-1);
11      }
12  }
13
14  int main()
15  {
16      int k;
17      cin>> k;
18      cout<< fact(k);
19  }
```

مثال تابع بازگشتی: تابع جمله nم سری فیبوناتچی

```cpp
#include <iostream>
using namespace std;

int fib(int n)
{
    if(n==1){
        return 1;
    }
    else if(n==2) {
        return 1;
    }
    else if (n>2){
        return fib(n-1)+fib(n-2);
    }
}
```

The Fibonacci Sequence

1,1,2,3,5,8,13,21,34,55,89,144,233,377...

1+1=2	13+21=34
1+2=3	21+34=55
2+3=5	34+55=89
3+5=8	55+89=144
5+8=13	89+144=233
8+13=21	144+233=377

مثال تابع بازگشتی: تابع جمله nم سری فیبوناتچی

```
17  int main()
18  {
19      int k;
20      cin>> k;
21      cout<< fib(k);
22  }
```

تمرین ۱

- برنامه ای بنویسید که x را از کاربر گرفته و تانژانت x را محاسبه و چاپ کند.

$$\tan x = \frac{\sin x}{\cos x}$$

بسط تیلور sin x

$$\sin x \sim x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!}$$

بسط تیلور cos x

$$\cos x \sim 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!}$$

تمرین ۲

- تابعی بازگشتی بنویسید که دو عدد ورودی x و y را گرفته و به صورت بازگشتی x به توان y محاسبه کند.

تمرین ۳

- تابعی بازگشتی بنویسید که دو عدد ورودی x و y را گرفته و به صورت بازگشتی بزرگترین مقسوم علیه بین xو y محاسبه کند.