

اصول کامپیوتر ۱

مبانی کامپیوتر و برنامه‌سازی

«جلسه‌ی سیزدهم»

دانشکده‌ی علوم ریاضی - دانشگاه شهید بهشتی

نیم‌سال اول ۹۰-۱۳۸۹

مدرس: سید علی کتان‌فروش

در جلسه‌ی قبل، برنامه‌ی زیر را برای تشخیص اول بودن عدد ارائه دادیم.

```
.....  
int n;  
cin >> n;  
int i=2;  
if ( n%2 != 0 ) {  
    for ( i=3; n%i!=0 && i*i<=n; i+=2 )  
        ;  
}  
  
if ( i*i <= n )  
    cout << n << " is composite." << endl;  
else  
    cout << n << " is prime." << endl;  
.....
```

مسئله. برنامه‌ای بنویسید که تمام اعداد
اول کوچکتر از ۱،۰۰۰،۰۰۰ را چاپ کند.

مسئله. برنامه‌ای بنویسید که تمام اعداد اول کوچکتر از ۱,۰۰۰,۰۰۰ را چاپ کند.

- ◆ برنامه‌ی «تشخیص اول بودن یک عدد» را مبنای قرار دهید.
- ◆ دستورات مربوط به دریافت n را حذف کنید.
- ◆ قطعه برنامه‌ی تشخیص اول بودن را در یک حلقه‌ی شمارنده قرار دهید.
- ◆ خروجی برنامه را به طور مناسب تغییر دهید.

برنامه‌ی اعداد اول کوچکتر از ۱,۰۰۰,۰۰۰

```
.....  
int n;  
cin >> n;  
int i=2;  
if ( n%2 != 0 ) {  
    for ( i=3; n%i!=0 && i*i<=n; i+=2 )  
        ;  
}  
  
if ( i*i <= n )  
    cout << n << " is composite." << endl;  
else  
    cout << n << " is prime." << endl;  
.....
```

برنامه‌ی اعداد اول کوچکتر از ۱,۰۰۰,۰۰۰

```
.....  
int n;  
for ( n = 1; n <= 1000000; n++ ) {  
    int i=2;  
    if ( n%2 != 0 ) {  
        for ( i=3; n%i!=0 && i*i<=n; i+=2 )  
            ;  
    }  
  
    if ( i*i <= n )  
        cout << n << " is composite." << endl;  
    else  
        cout << n << " is prime." << endl;  
}
```

برنامه‌ی اعداد اول کوچکتر از ۱,۰۰۰,۰۰۰

```
.....  
int n;  
for ( n = 1; n <= 1000000; n++ ) {  
    int i=2;  
    if ( n%2 != 0 ) {  
        for ( i=3; n%i!=0 && i*i<=n; i+=2 )  
            ;  
    }  
    if ( i*i > n )  
        cout << n << ' ' ;  
}  
cout << endl;  
.....
```

عدد ۲ را خارج از حلقه چاپ کنید و
حلقه را تنها بر روی اعداد فرد کوچکتر
از ۱,۰۰۰,۰۰۰ اجرا کنید.

برنامه‌ی اعداد اول کوچکتر از ۱,۰۰۰,۰۰۰

```
.....  
cout << 2 << ' ' ;  
int n;  
for ( n = 3; n <= 1000000; n+=2 ) {  
    int i,j;  
    j = sqrt(n);  
    for ( i=3; n%i!=0 && i<=j; i+=2 )  
        ;  
    if ( i > j )  
        cout << n << ' ' ;  
}  
cout << endl;  
.....
```


Nested loops

حلقه‌های تو در تو

اعداد اول کوچکتر از ۱,۰۰۰,۰۰۰

```
workspace :bash
File Edit View Scrollback Bookmarks Settings Help
167 491171 491201 491213 491219 491251 491261 491273 491279 491297 491299 4
91327 491329 491333 491339 491341 491353 491357 491371 491377 491417 491423
 491429 491461 491483 491489 491497 491501 491503 491527 491531 491537 4915
39 491581 491591 491593 491611 491627 491633 491639 491651 491653 491669 49
1677 491707 491719 491731 491737 491747 491773 491783 491789 491797 491819
491833 491837 491851 491857 491867 491873 491899 491923 491951 491969 49197
7 491983 492007 492013 492017 492029 492047 492053 492059 492061 492067 492
077 492083 492101 492113 492227 492251 492253 492257 492281 492293 492299 4
92319 492377 492389 492397 492403 492409 492413 492421 492431 492463 492467
 492487 492491 492511 492523 492551 492563 492587 492601 492617 492619 4926
29 492631 492641 492647 492659 492671 492673 492707 492719 492721 492731 49
2757 492761 492763 492769 492781 492799 492839 492853 492871 492883 492893
492901 492911 492967 492979 493001 493013 493021 493027 493043 493049 49306
7 493093 493109 493111 493121 493123 493127 493133 493139 493147 493159 493
169 493177 493193 493201 493211 493217 493219 493231 493243 493249 493277 4
93279 493291 493301 493313 493333 493351 493369 493393 493397 493399 493403
 493433 493447 493457 493463 493481 493523 493531 493541 493567 493573 4935
79 493583 493607 493621 493627 493643 493657 493693 493709 493711 493721 49
3729 493733 493747 493777 493793 493807 493811 493813 493817 493853 493859
```

مثال. برنامه‌ای بنویسید که تمام اعداد
فیثاغورثی کوچکتر از ۱۰۰ را چاپ کند.

یعنی تمام سه‌تایی‌هایی از اعداد صحیح مثبت
مثل a, b, c به قسمی که $a < 100$ و

$$a^2 = b^2 + c^2, \quad a, b, c \in \mathbb{N}$$

قضیه فرما. به ازای هر عدد صحیح $n > 2$ ، معادله‌ی

$$x^n = y^n + z^n$$

هیچ جوابی در مجموعه‌ی اعداد صحیح ندارد.

برنامه‌ی اعداد فیثاغورثی

راه حل ۱

```
int main( ) {
    int a,b,c;
    for ( a=1; a<=100; a++) {
        for ( b=1; b<=100; b++) {
            for ( c=1; c<=100; c++ ) {
                if (a*a==b*b+c*c)
                    cout << a << ' ' << b
                        << ' ' << c << endl;
            }
        }
    }
    system("pause");
    return EXIT_SUCCESS;
}
```

Nested loops

حلقه‌های تو در تو

اعداد فیثاغورثی

```
workspace : bash
File Edit View Scrollback Bookmarks Settings Help
85 75 40
85 77 36
85 84 13
87 60 63
87 63 60
89 39 80
89 80 39
90 54 72
90 72 54
91 35 84
91 84 35
95 57 76
95 76 57
97 65 72
97 72 65
100 28 96
100 60 80
100 80 60
100 96 28
```

سه تایی‌های تکراری را چاپ نکنید.

Nested loops

برنامه‌ی اعداد فیثاغورثی

راه حل ۱

```
int main( ) {
    int a,b,c;
    for ( a=1; a<=100; a++) {
        for ( b=1; b<=100; b++) {
            for ( c=1; c<=100; c++ ) {
                if (a*a==b*b+c*c)
                    if ( b>c )
                        cout << a << ' ' << b
                            << ' ' << c << endl;
            }
        }
    }
    system("pause");
    return EXIT_SUCCESS;
}
```

سه تایی‌های تکراری را بررسی نکنید.

Nested loops

برنامه‌ی اعداد فیثاغورثی

راه حل ۲

```
int main( ) {
    int a,b,c;
    for ( a=1; a<=100; a++) {
        for ( b=1; b<=100; b++) {
            for ( c=1; c<b; c++ ) {
                if (a*a==b*b+c*c)
                    cout << a << ' ' << b
                        << ' ' << c << endl;
            }
        }
    }
    system("pause");
    return EXIT_SUCCESS;
}
```

Nested loops

برنامه‌ی اعداد فیثاغورثی

راه حل ۳

```
int main( ) {  
    int a,b;  
    for ( a=1; a<=100; a++) {  
        for ( b=1; b<=100; b++) {  
            double c = sqrt(a*a-b*b);  
            if ( floor(c) == c && c < b)  
                cout << a << ' ' << b  
                    << ' ' << c << endl;  
        }  
    }  
    system("pause");  
    return EXIT_SUCCESS;  
}
```

تابع جزء صحیح `floor()`
کتابخانه‌ی `cmath`

Nested loops

برنامه‌ی اعداد فیثاغورثی

راه حل ۴

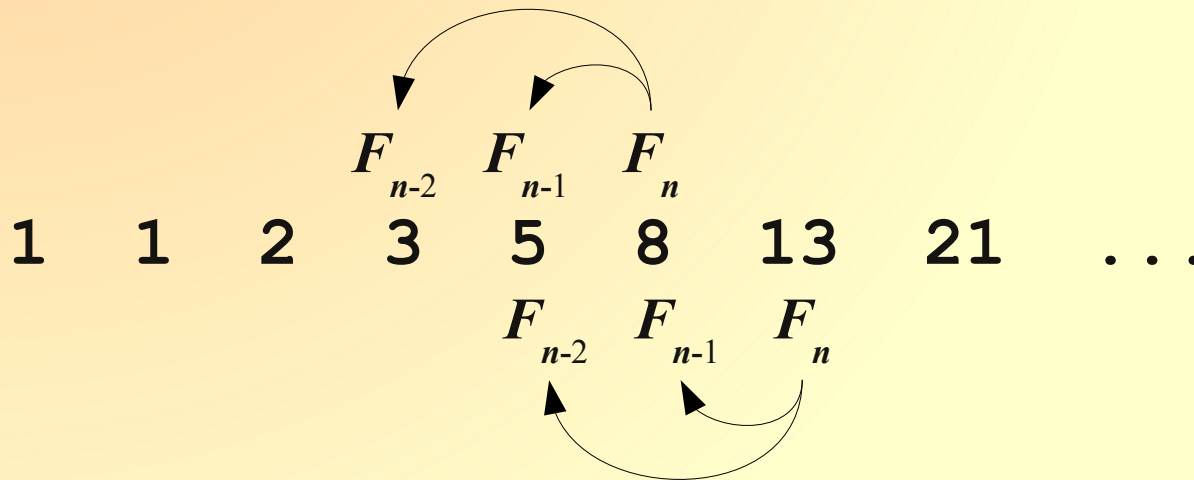
```
int main( ) {
    double sqrt2 = sqrt(2);
    int a,b;
    for ( a=1; a<=1000; a++) {
        for ( b=1; b<=a/sqrt2; b++) {
            double c=sqrt(a*a-b*b);
            if ( floor(c)==c )
                cout << a << ' ' << b
                    << ' ' << c << endl;
        }
    }
    system("pause");
    return EXIT_SUCCESS;
}
```


مسئله. برنامه‌ای بنویسید که جمله‌ی n -ام
دنباله اعداد فیبوناچی را چاپ کند.

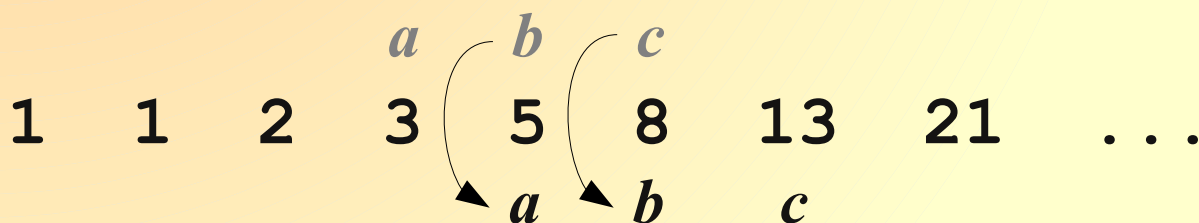
1,1,2,3,5,8,13,21,...

$$F_n = \begin{cases} 1 & \text{if } n=1,2 \\ F_{n-2} + F_{n-1} & \text{if } n > 2 \end{cases}$$

راهنمائی. تمام اعداد دنباله را از جمله‌ی اول تا جمله‌ی n -ام محاسبه کنید.



لازم نیست تمام اعداد دنباله را نگهداریم.
 برای محاسبه، تنها سه جمله‌ی متوالی دنباله
 کافی است.



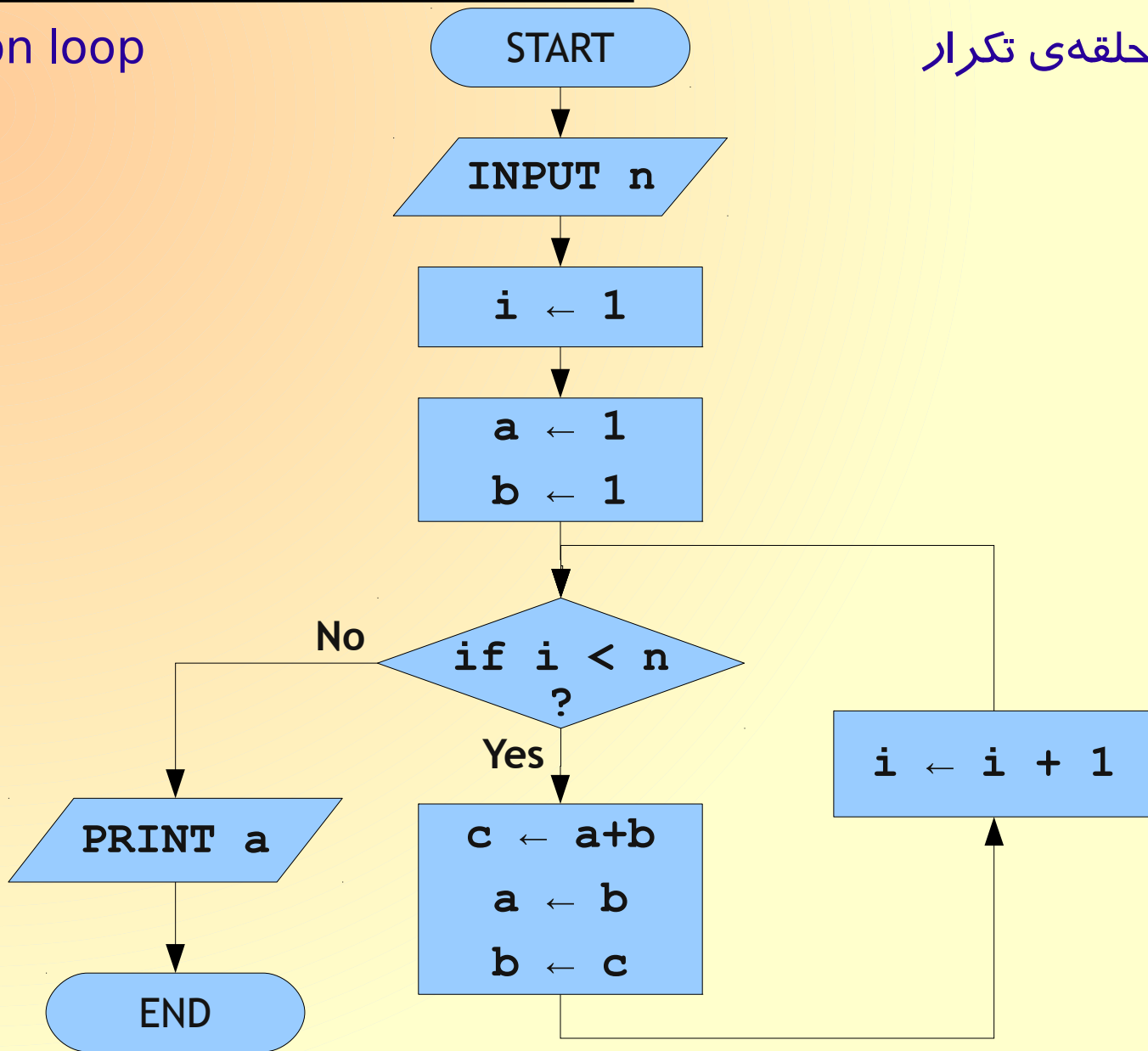
```
c ← a+b
```

```
a ← b
```

```
b ← c
```

Iteration loop

حلقه‌ی تکرار



برنامه‌ی اعداد فیبوناچی

```
int main( ) {  
    int n;  
    cout << " n ? " ;  
    cin >> n;  
  
    int a,b,c;  
    a=1; b=1;  
    int i;  
    for ( i=1; i<n; i++ ) {  
        c=a+b;  
        a=b;  
        b=c;  
    }  
    cout << a << endl;  
    . . . .
```

Iteration loop

حلقه‌ی تکرار

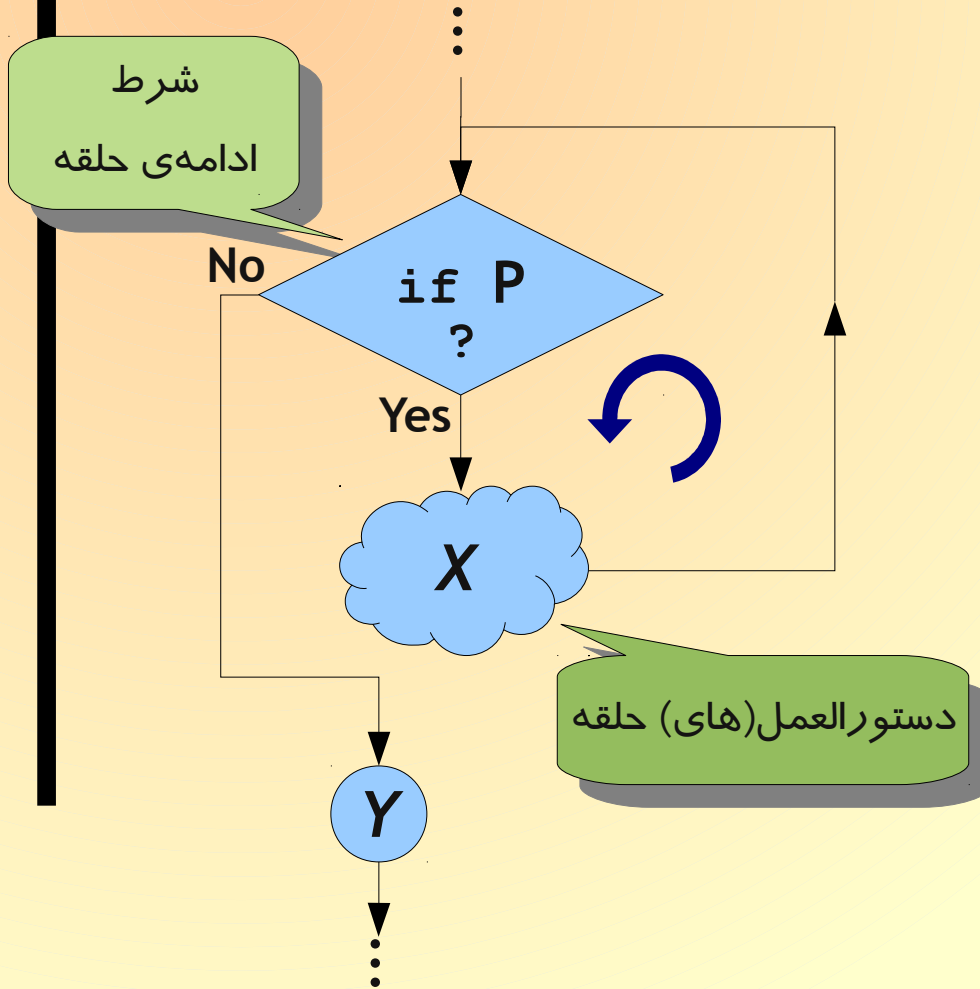
مسئله.

کوچکترین عدد فیبوناچی بزرگ‌تر از ۱۰۰۰ چیست؟

Iteration loop

حلقه‌ی تکرار

حلقه‌های شرط.



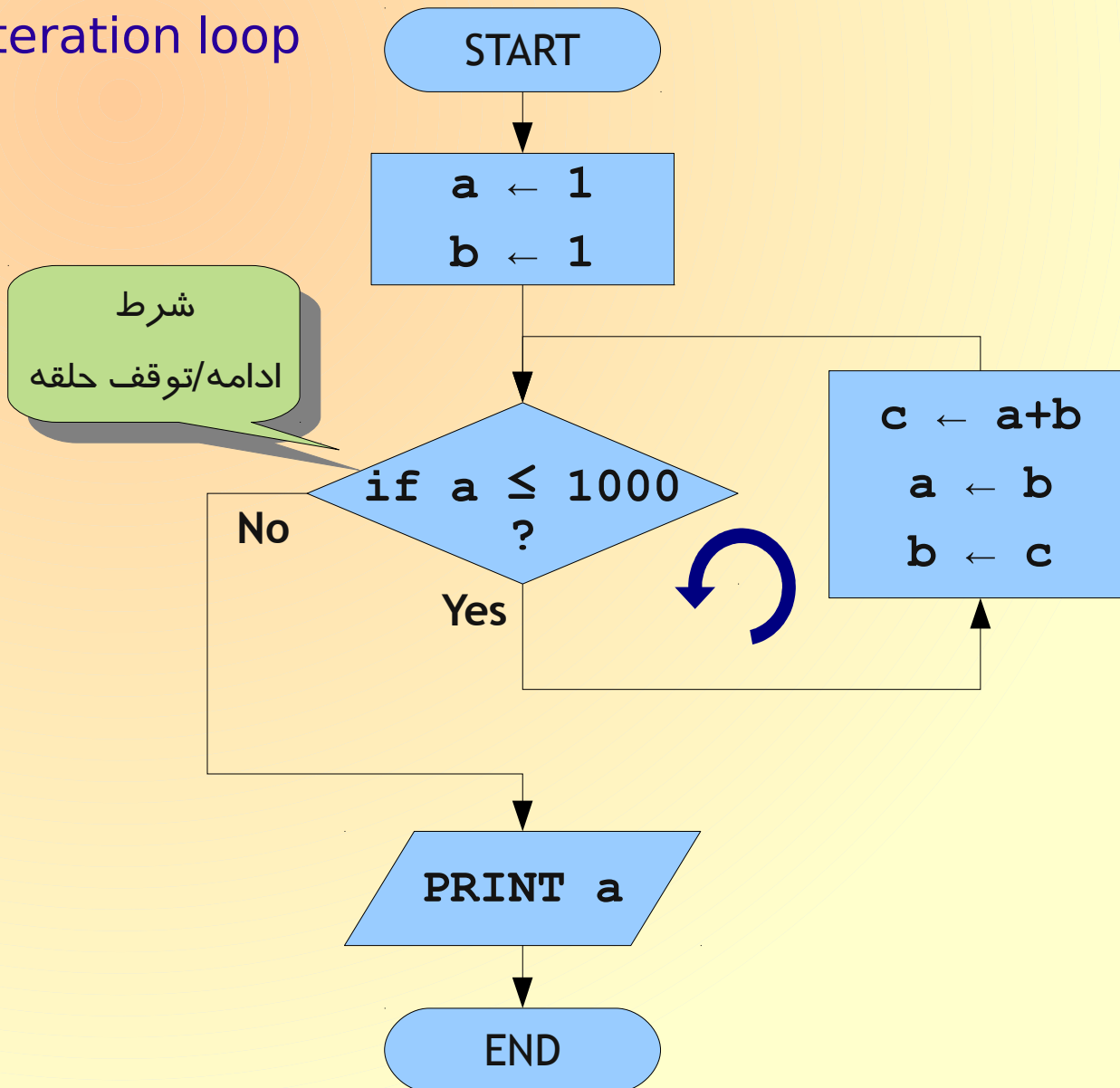
```
while ( P )  
    X;  
Y
```

الگوی C/C++

Iteration loop

حلقه‌ی تکرار

حلقه‌های شرط



اولین عدد فیبوناچی بزرگ‌تر از ۱۰۰۰

```
int main( ) {  
    int a,b,c;  
    a=1; b=1;  
    while ( a<=1000 ) {  
        c=a+b;  
        a=b;  
        b=c;  
    }  
    cout << a << endl;  
    system("pause");  
    return EXIT_SUCCESS;  
    ....  
}
```

مثال. برنامه‌ای بنویسید که لیستی از اعداد مثبت را از کاربر دریافت کند و میانگین آن‌ها را محاسبه و چاپ کند.

فرض کنید کاربر با وارد کردن عدد صفر، به پایان رسیدن لیست را به برنامه اعلام می‌کند.

خواندن لیستی از اعداد که با عدد صفر پایان می‌یابد و محاسبه‌ی میانگین

```
int n;  
double x,S;  
  
n = 0;  
S = 0;  
cin >> x;  
while ( x != 0 ) {  
    S = S + x;  
    n = n + 1;  
    cin >> x;  
}  
cout << "Mean = " << S/n << endl;  
....
```

آیا برنامه‌ی زیر درست است؟

```
int n;  
double x, S;  
  
n = 0;  
S = 0;  
  
while ( x != 0 ) {  
    cin >> x;  
    S = S + x;  
    n = n + 1;  
}  
cout << "Mean = " << S/n << endl;  
....
```

آیا برنامه‌ی زیر درست است؟

```
int n;  
double x,S;  
  
S = 0;  
cin >> x;  
for ( n=0; x!=0; n++ ) {  
    S = S + x;  
    cin >> x;  
}  
cout << "Mean = " << S/n << endl;  
.....
```

مسئله. الگوریتم تقسیمات متوالی برای نمایش اعداد در مبنای ۲ را پیاده‌سازی کنید.

مثال. برای $n=82$ داریم:

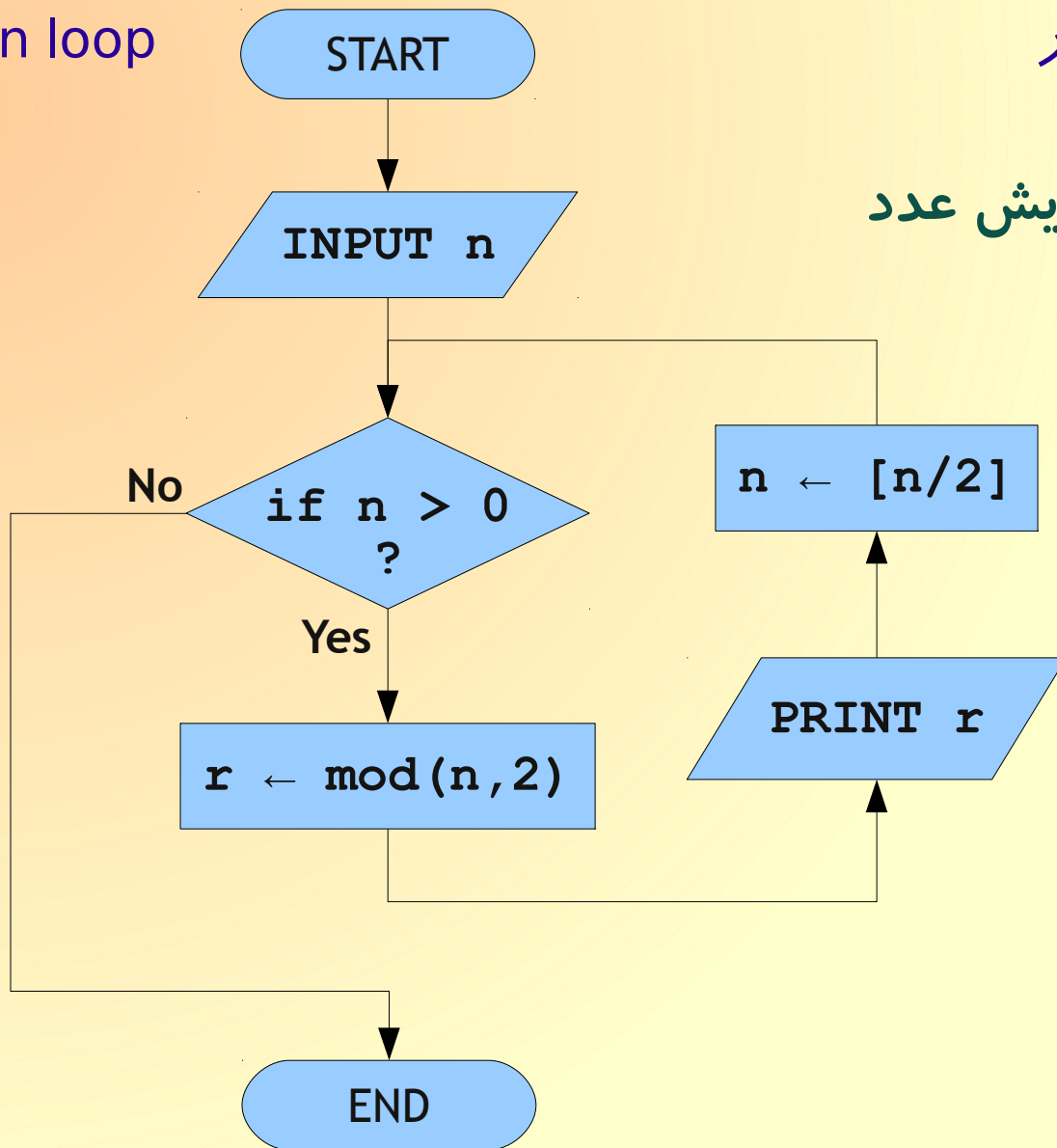
$$82 = (1010010)_2$$

Quotient	Remainder
$82 \div 2 = 41$	0
$41 \div 2 = 20$	1
$20 \div 2 = 10$	0
$10 \div 2 = 5$	0
$5 \div 2 = 2$	1
$2 \div 2 = 1$	0

Iteration loop

حلقه‌ی تکرار

الگوریتم نمایش عدد
در مبنای ۲



نمایش عدد در مبنای ۲

راه حل ۱

```
int main() {  
    int n;  
    cout << "n ? ";  
    cin >> n;  
  
    while ( n > 0 ) {  
        int r = n % 2;  
        cout << r;  
        n = n / 2;  
    }  
    cout << endl;  
    ....  
}
```


Iteration loop

حلقه‌ی تکرار

نمایش عدد در مبنای ۲

```
workspace : sh
File Edit View Scrollback Bookmarks Settings Help
agui:~/workspace> ./binary
n ? 17
10001

agui:~/workspace> ./binary
n ? 31
11111

agui:~/workspace> ./binary
n ? 82
0100101

agui:~/workspace> ./binary
n ? 128
00000001

agui:~/workspace> ./binary
n ? 1536
00000000011
```

ارقام باید از ته به سر چاپ شوند.

نمایش عدد در مبنای ۲

```
int n;  
cin >> n;  
  
string R = "";  
while ( n > 0 ) {  
    if ( n%2 == 0 )  
        R = "0" + R;  
    else  
        R = "1" + R;  
    n = n / 2;  
}  
cout << R << endl;  
.....
```

راه حل ۲

عملگر + بین رشته‌های حرفی،
رشته‌ی جدیدی تولید می‌کند
که از کنار هم قرار دادن دو
رشته بدست می‌آید.

مسئله. تنها با استفاده از متغیرهای ساده‌ی عددی، برنامه‌ای بنویسید که عدد صحیح و مثبت n را دریافت کند و آنرا در مبنای دو نمایش دهد.

راهنمائی. بجای استفاده از روش تقسیمات متوالی و معکوس کردن خروجی، روشی بدست آورید که مستقیماً ارقام را از پرارزشترین رقم (اولین رقم سمت چپ) به کم‌ارزشترین (اولین رقم سمت راست) محاسبه کند.