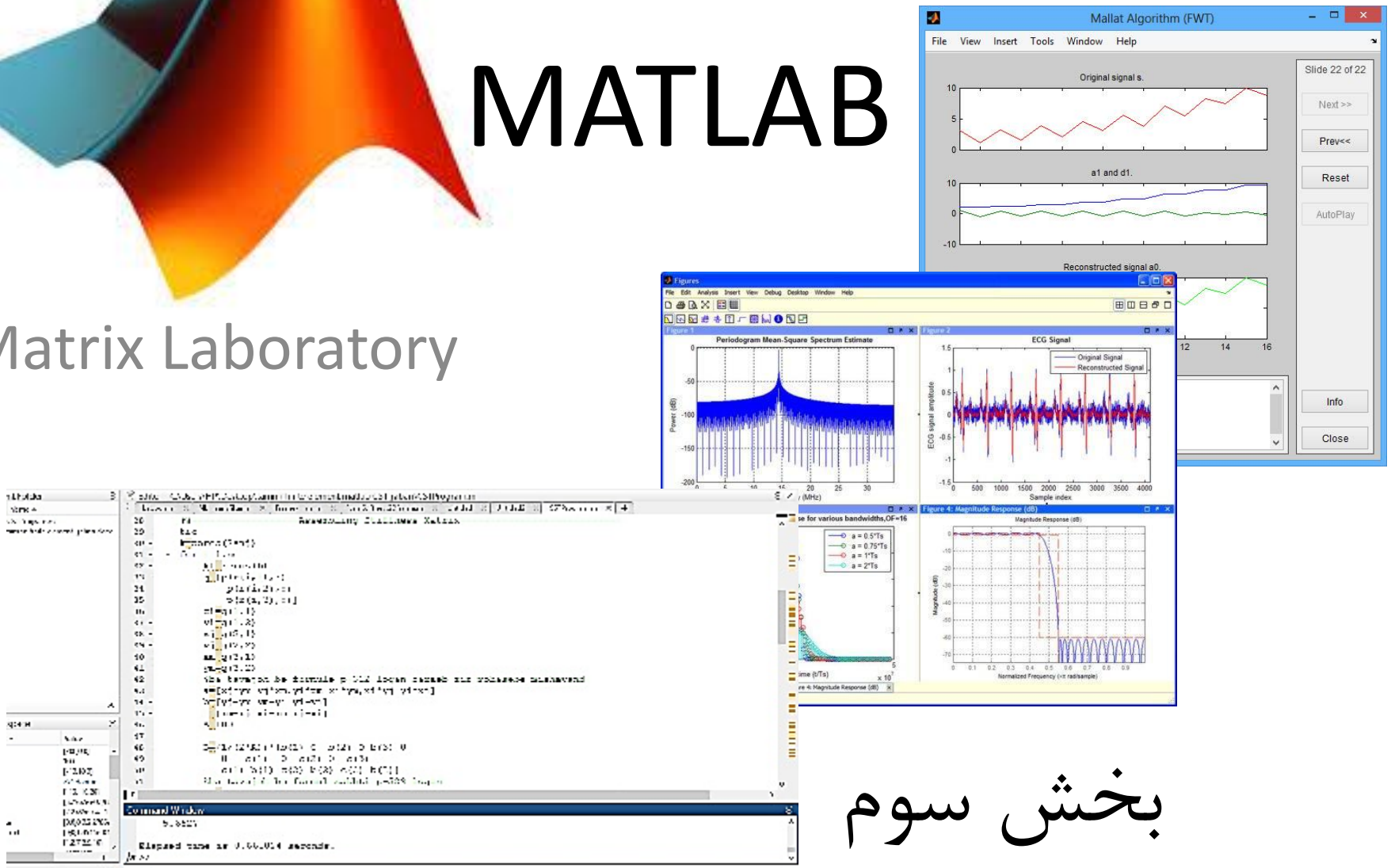
آموزش زبان برنامه نویسی متلب

MATLAB

Matrix Laboratory



بخش سوم

ساختارهای کنترل: چنانچه المانهای زیر در یک زبان برنامه نویسی وجود داشته باشند می توان هر نوع الگوریتمی را پیاده سازی کرد.

(الف) توالی (بتواند دستورها را پشت سر هم اجرا کند)

(ب) انشعاب (تصمیم گیری) **if , switch**

(ج) تکرار (حلقه های تکرار معین یا نامعین) **while , for**

---

if (شرط یا قیاس منطقی)

دستوراتی که در صورت صحیح بودن شرط اجرا خواهند شد

else

دستوراتی که در صورت صحیح نبودن شرط اجرا خواهند شد

اجباری نیست

end

$$f(x)=\begin{cases}-2 & x<-1\\ x & -1\le x\le 1\\ +2 & x>1\end{cases}$$

$$\begin{aligned}
&if \quad (x<-1)\\
&f(x)=-2\\
&elseif \quad (x>1)\\
&f(x)=2\\
&else\\
&f(x)=x\\
&end
\end{aligned}$$

ساختار کلی

$$\begin{aligned}
&if \quad (Logical.Condition1)\\
&--Calculation(1)--\\
&elseif \quad (Logical.Condition2)\\
&--Calculation(2)--\\
&elseif \quad (Logical.Condition3)\\
&--Calculation(3)--\\
&else\\
&--Calculation(4)--\\
&end
\end{aligned}$$

اجباری نیست

دستور switch حالت خیلی خلاصه شده از if است.

```
n ∈ {1, 2, 3, 4}

if (n==1)
    ____
elseif (n==2)
    ____
elseif (n==3)
    ____
;
```

```
Switch n
    Case 1
        ~~~
    Case 2
        ~~~
    Case 3
        ~~~
    Case 4
        ~~~
end
```

حالت کلی

```
switch switch_expression
    case case_expression
        statements
    case case_expression
        statements
    :
    otherwise
        statements
end
```

```
n ∈ {1, 2, 3, 4, 5, 6}

if (n==1)
    ____
elseif (n==2)
    ____
elseif (n==3)
    ____
;
```

```
Switch n
    Case {1,5}
        ~~~
    Case 2
        ~~~
    Case {3,6}
        ~~~
    Case 4
        ~~~
end
```

مثال

```
n = input('Enter a number: ');

switch n
    case -1
        disp('negative one')
    case 0
        disp('zero')
    case 1
        disp('positive one')
    otherwise
        disp('other value')
end
```

حلقه for زمانی کاربرد دارد که حداکثر تعداد تکرار ها **مشخص** باشد.

```
for (متغیر)=(دامنهٔ تغییرات)
    دستورات تکرار شونده
end
```

برای دامنهٔ تغییرات در حلقهٔ for می توان به کمک عملگر ":" یک دامنه ای از اعداد تعریف کرد یا حتی می توان دامنهٔ تغییرات را به صورت برداری از اعداد تعریف کرد.

i=[1 4 6 8 9]

با while می شود for را پیاده سازی کرد ولی برعکس نمی شود.

```
for i=1:100
j=i+1;
  disp (j)
end
```

```
for i=[1,4,6,8,9]
  disp (i^2)
end
```

چنانچه تعداد تکرارها مشخص نباشد می توان از حلقهٔ while استفاده کرد.

while (شرط ادامهٔ حلقه)

دستورات تکرار شونده

end

با while می شود for را پیاده سازی کرد ولی برعکس نمی شود.

```
i=0;
while true
i=i+1
end
```

این حلقه تا ابد تکرار می شود

تمرین: برنامه ای بنویسید که از بازهٔ اعداد ۱ الی ۱۰، اعداد فرد را نمایش دهد.

```
for i=1:10
if mod(i,2)==1
disp(i)
end
end
```

or

```
i=1;
while i<9
    i=i+2;
    disp(i)
end
```

تمرین: برنامه ای بنویسید که از بازهٔ اعداد ۱ الی ۱۰، اعداد ۵ و ۷ را حذف کرده و مابقی را نمایش دهد.

```
for i=1:10
 if (i~=5)&&(i~=7)
   disp(i)
 end
end
```

or

```
for i=1:10
 if (i==5)||(i==7)
  continue
 end
   disp(i)
end
```

تمرین: برنامه ای بنویسید که در بازهٔ اعداد ۱ الی ۱۰ شروع به شمارش کرده و به محض رسیدن به عدد ۷ شمارش را متوقف کند.

```
for i=1:10
 if (i==7)
   break
 end
disp(i)
end
```

or

```
i=1;
while (i~=10)
    i=i+1;
 if (i==7)
   break
 end
disp(i)
end
```

عملگرهای continue و break:

```
while (شرط ادامهٔ حلقه)

  if (قیاس منطقی)
      continue
  end

      دستورات تکرار شونده
end
```

```
while (شرط ادامهٔ حلقه)

  if (قیاس منطقی)
      break
  end

      دستورات تکرار شونده
end
```

استفادهٔ عملی از حلقه ها: (نوشتن سری ها)

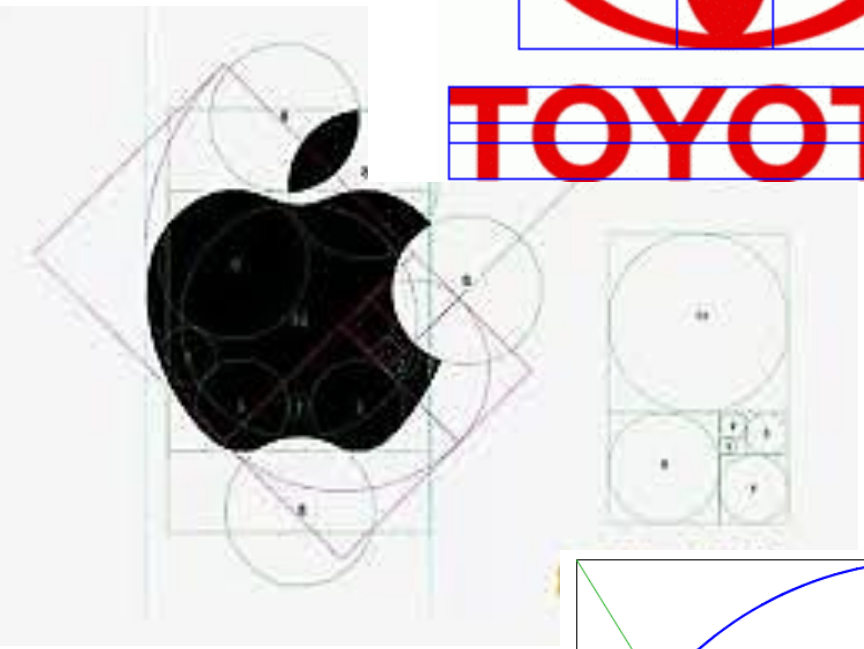
مثال: سری فیبوناچی (Fibonacci)

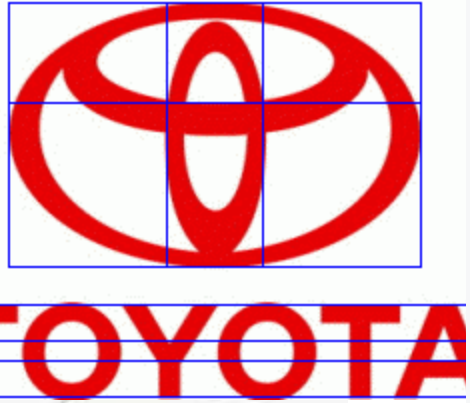
$$\begin{cases} F_k = 1, & k = 1 \\ F_k = 1, & k = 2 \\ F_k = F_{k-1} + F_{k-2} & k \geq 3 \end{cases} \rightarrow 1, 1, 2, 3, 5, 8, 13, 21, 34, \ldots$$

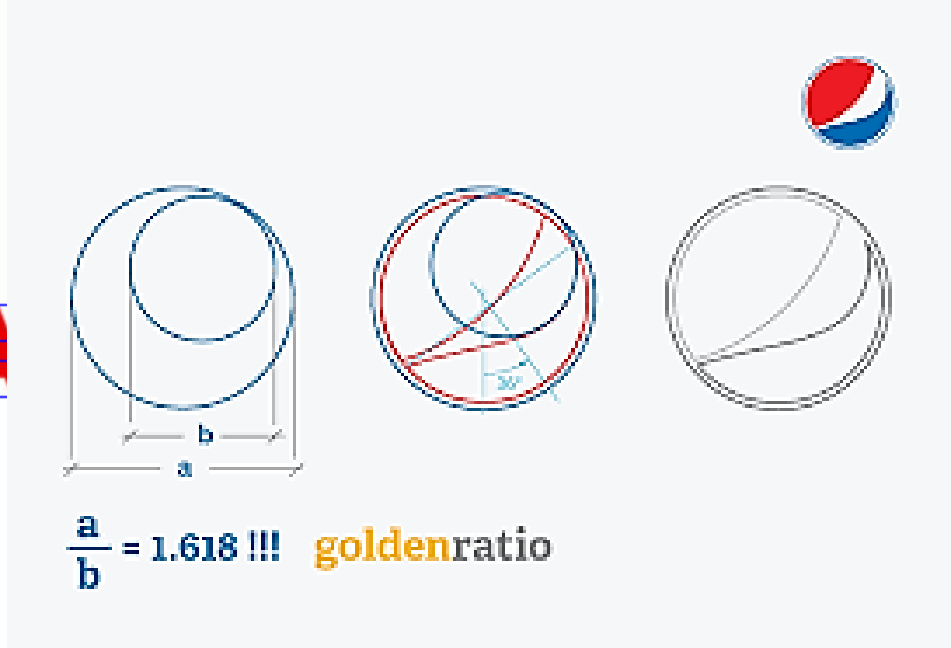
یکی از خاصیت های سری فیبو ناچی تولید عدد طلایی است یعنی در بینهایت، حد نسبت هرجمله به جملع قبل آن به عدد طلایی فی (Φ) میل می کند. فیبوناچی در آثار هنری خود این نسبت را رعایت نموده است.

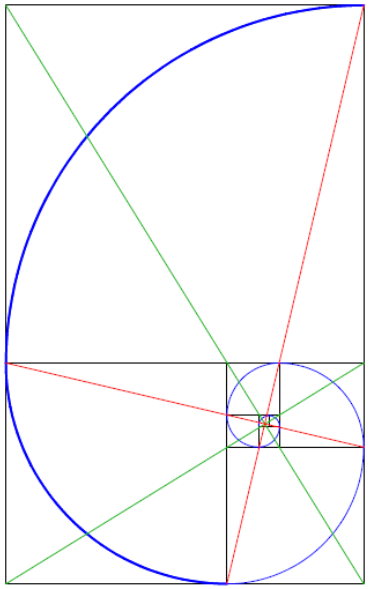
$$\lim_{x \to \infty} \frac{F_k}{F_{k-1}} = \phi = \frac{1+\sqrt{5}}{2}$$

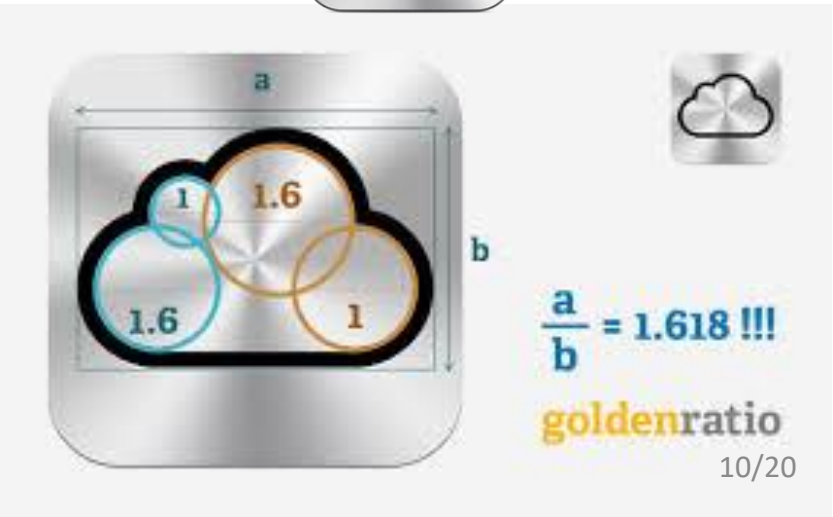
این نسبت در طبیعت به وفور مشاهده می شود. به عنوان مثال نسبت یک بند انگشت به طول انگشت. نسبت طول انگشت به طول کف دست. نسبت طول کف دست به طول آرنج تا نوک انگشت، لاله گوش و ...




TOYOTA


b
a
a/b = 1.618 !!! goldenratio




iCloud
a
b
1
1.6
1.6
1
a/b = 1.618 !!!
goldenratio

با دستور edit میتوان وارد صفحهٔ ساخت m.Files شد:

```
clc;
clear;

F(1)=1;
F(2)=1;

F(3)=F(2)+F(1);
F(4)=F(3)+F(2);
F(5)=F(4)+F(3);
F(6)=F(5)+F(4);
F(7)=F(6)+F(5);
F(8)=F(7)+F(6);
```

```
clc;
clear;

F(1)=1;
F(2)=1;

for k=3:100
    F(k)=F(k-1)+F(k-2);
end
```

```
for j=1:99
    r(j)=F(j+1)/F(j);
end
```

plot (r);

معادل هم هستند

q = F(2:end) ./ F(1:end-1);

عملگر های نقطه به نقطه: *. یا /. ?

آیا می توان دو ماتریس را باهم مقایسه کرد تا مشاهده شود آیا درایهٔ متناظری دارند یا نه؟

دو عملگر برای بررسی های کلی

بررسی انطباق یا عدم انطباق جامع مقادیر با دستور all اجرا می شود که ساختار کلی آن به صورت زیر است. این دستور همانند and عمل کرده و قیاس منطقی بیان شده در داخل پرانتز باید برای تمامی مقادیر صدق کند تا خروجی all برابر با ۱ (یعنی true یا صحیح) شود.

all(عبارت منطقی یا مقایسه ای)=?

آیا همه مقادیر مورد نظر شرط مورد نظر را برآورد می کنند؟

بررسی انطباق یا عدم انطباق حداقل یکی از مقادیر با دستور any اجرا می شود که ساختار کلی آن به صورت زیر است. این دستور همانند or عمل کرده و قیاس منطقی بیان شده در داخل پرانتز کافی است تا برای یکی از مقادیر صدق کند تا خروجی any برابر با ۱ (یعنی true یا صحیح) شود.

any(عبارت منطقی یا مقایسه ای)=?

آیا مقداری از مقادیر مورد نظر وجود دارد که شرط مورد نظر را برآورد کند؟

مثال:

X=[1 4 5 7]

any (X==3) ➔ 0

any (X>3) ➔ 1 all (X<8) ➔ 1

ثبت مدت زمانی که طول می کشد یک قسمت مشخصی از برنامه بین دو عبارت tic و toc اجرا شود.

با تغییر مقدار N و بیشتر شدن آن مدت زمان اجرای کد هم بیشتر می شود.

با توجه به قدرت کامپیوتر مدت زمان های مختلفی بدست می آید.

```
clc;
clear;

N=100;

tic;

F(1)=1;
F(2)=1;

for k=3:N
    F(k)=F(k-1)+F(k-2);
end

toc;

r=F(2:end)./F(1:end-1);
```

به دلیل اشکالاتی که در نوع کد نوشته شده وجود دارد زمان زیادی صرف اجرا می شود.
به همین خاطر نوار عمودی سمت راست m فایل نارنجی رنگ شده است.

اگر کد نوشته شده اصولی باشد نوار عمودی سمت راست m فایل سبز رنگ می شود.

علت بروز آن مشکل این است که از روش pre-allocation استفاده نمی شود.

$$F_1 = 1$$

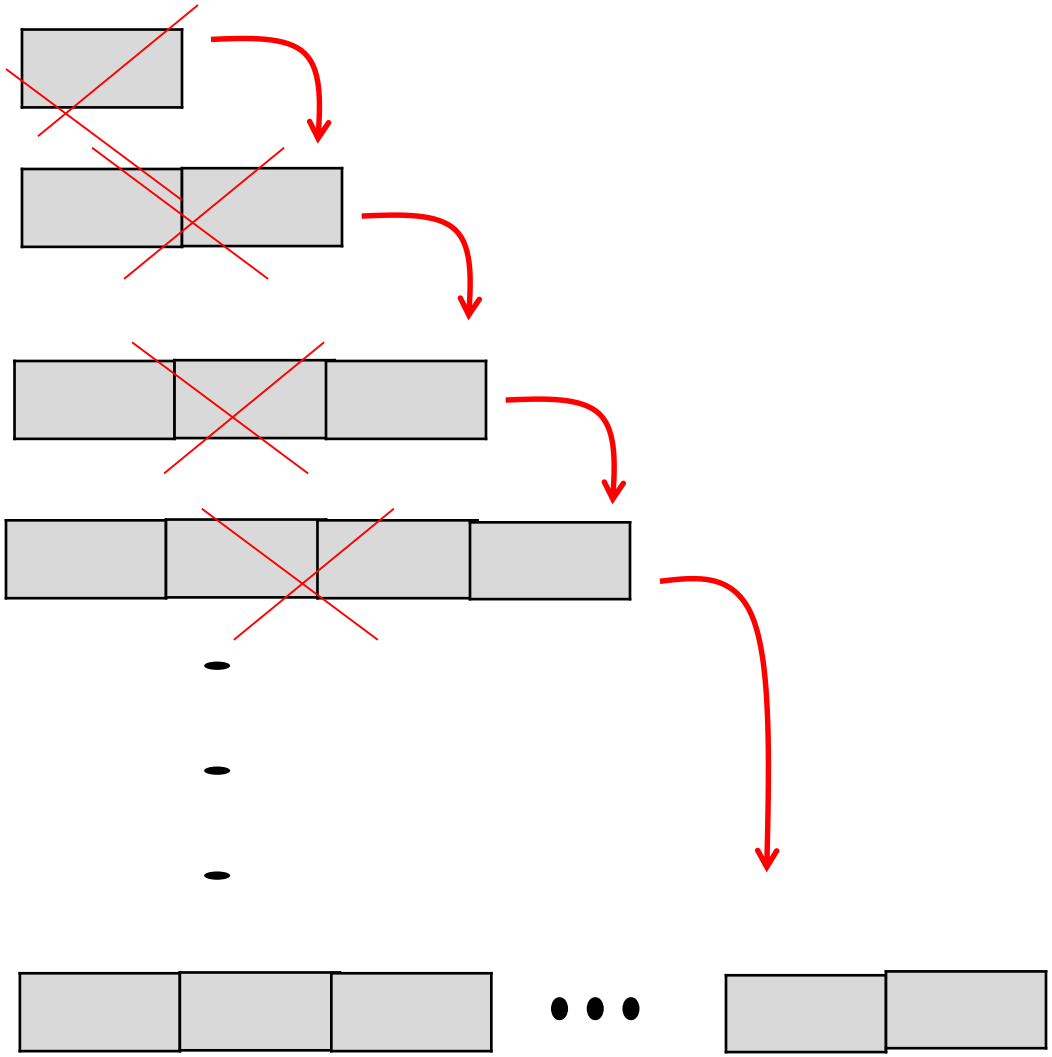
$$F_2 = 1$$

$$F_3 = F_2 + F_1$$

$$F_4 = F_3 + F_2$$

$$\vdots$$

$$F_{1000} = F_{999} + F_{998}$$



برای رفع این مشکل بهتر است در همان ابتدا فضایی به اندازه N در حافظه تعریف کرده و به این متغیر اختصاص می دهیم.

```
F=zeros(1, N);
```

تمرین:

برنامه ای بنویسید که در آن دنبالهٔ اعداد بسط فیبوناچی را تا عدد کمتر از ۵۰۰۰ چاپ کند.

برای این منظور عبارت زیر را در کد نوشته شده پیشین بگنجانید. توجه شود که خانه های اضافه ای که در حافظ ایجاد کردیم می بایست در انتهای اجرای کد حذف شوند.

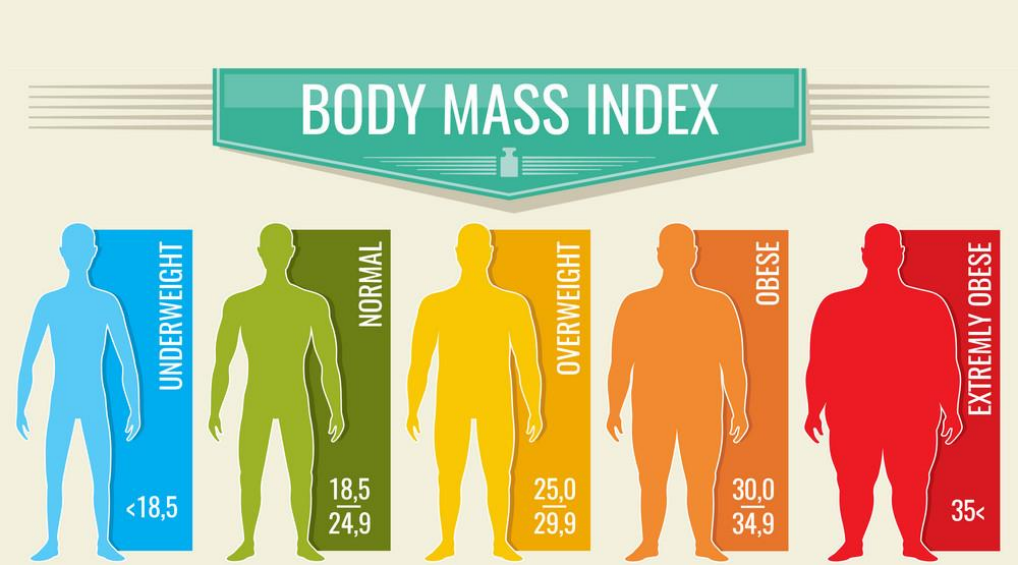
```
F(k:end)=[];     % Deletion
```

```
for k=3:N
    F(k)=F(k-1)+F(k-2);

    if F(k)>5000
        break;
    end
end
```

شاخص سلامتی (Heath Index)

$$BMI = \frac{W}{h^2} = \frac{weight\,(kg)}{(height\,(m))^2}$$



شاخص سلامتی (Heath Index)

```
clc;
clear;
close all;  %بستن همه نمودارهای رسم شده

disp ('this program will calculate your health index')

W=input('enter your weight (kg) : ');
h=input('enter your height (m) : ');
a=W/(h^2);
    disp (' ')  %ایجاد خط خالی
    disp 'your health index ='; disp(a)
if a>25
    disp('you are over-weighted')
elseif a<20
    disp('you are under-weighted')
else
    disp('your weight is normal')
end
```

تمرین: همین برنامه را طوری کامل کنید که اگر شخصی قد خود را به اشتباه بر حسب سانتی متر وارد نمود مقدار وارد شده را خود برنامه تصحیح کند.

```
clc;
clear;
close all;
while true
disp ('this program will calculate your health index')
W=input('enter your weight (kg) : ');
h=input('enter your height (m) : ');
a=W/(h^2);
    disp (' ')
    disp (['your health index =' num2str(a)]);
if a>25
    disp('you are over-weighted')
elseif a<20
    disp('you are under-weighted')
else
    disp('your weight is normal')
end
end
```

Ctrl + C ➔ break

```
clc;
clear;
close all;
while true
disp ('this program will calculate your health index')
W=input('enter your weight (kg) : ');
h=input('enter your height (m) : ');
a=W/(h^2);
    disp (' ')
 disp (['your health index =' num2str(a)]);
if a>25
    disp('you are over-weighted')
elseif a<20
    disp('you are under-weighted')
else
    disp('your weight is normal')
end
Answer=input('do you want to continue  [yes/No] ? ', 's');
if strcmpi (Answer, 'No') || strcmpi (Answer, 'N')
break;
end
end
```

در این برنامه مقدار bmi و وضعیت افراد در متغیر List که از نوع آرایه سلولی است ثبت می شود.

```
clc;
clear;
close all;
List={};
i=1;
```

```
while true
    clc
W=input('enter your weight (kg) : ');
h=input('enter your height (m) : ');
bmi=W/(h^2);
List{1,i}=bmi;
if bmi>25
    List{2,i}='Over-weighted';
elseif bmi<20
    List{2,i}='Under-weighted';
else
    List{2,i}='Normal';
end
Answer=input('do you want to continue
[yes/No] ? ', 's');
if strcmpi (Answer, 'No') || strcmpi (Answer,
'N')
break;
end
i=i+1;
end
disp(List);
```

تمرین:

برنامه شاخص سلامت را طوری بنویسید که ویژگی های زیر را داشته باشد:

(الف) برای هر تعداد نفر قابل اجرا بوده و نیاز به اجرای مجدد نداشته باشد.

(ب) با بررسی مقدار شاخص سلامت اعلام کند که شخص اضافه وزن دارد یا کمبود وزن دارد.

(ج) با توجه به قد شخص محدودة حداقل و حداکثر وزن مجاز وی را نشان دهد.

(د) اعلام کند که شخص مورد نظر چند کیلو گرم اضافه وزن یا کمبود وزن دارد.