

Lecture 1

Matrix manipulations

- Sub-matrices
- Reproduction
- Reshape
- Determinant, inversion

Flow control

- if, if else, find

Matrices

- $A = [1 \ 2 \ 3 \ 4; 5 \ 6 \ 7 \ 8]$
- $\text{reshape}(1:8, 4, 2)'$

```
MATLAB Mobile  
>> reshape(1:8, 4, 2)'  
  
ans =  
  
    1    2    3    4  
    5    6    7    8
```

```
MATLAB Mobile  
>> A = [1 2 3 4; 5 6 7 8]  
  
A =  
  
    1    2    3    4  
    5    6    7    8
```

Rows

- `>> A(1,:)`

`ans =`

1 2 3 4

- `>> A(2,:)`

`ans =`

5 6 7 8

Columns

- `>> A(:,1)`

`ans =`

1


5

- `>> A(:,[2 3])`

`ans =`

2 3

6 7



```
>> A(:, [2 3])

ans =

     2     3
     6     7
```

Sub-matrix

- `a=1:1:36`
- `A=reshape(a,4,9)`
- `% Extract rows 2:4 and columns 5:8 of A`
- `A(2:4,5:8)`

```
MATLAB Mobile  
--> A(2:4,5:8)  
  
ans =  
    18    22    26    30  
    19    23    27    31  
    20    24    28    32
```

```
MATLAB Mobile  
--> A  
  
A =  
     1     5     9    13    17    21    25    29    33  
     2     6    10    14    18    22    26    30    34  
     3     7    11    15    19    23    27    31    35  
     4     8    12    16    20    24    28    32    36
```

Reshape

```
v=1:12;  
reshape(v,3,4)
```

```
ans =
```

1	4	7	10
2	5	8	11
3	6	9	12

Reshape

- `a=1:1:36`
- `m=4;n=9`
- `A=reshape(a,4,9)`
- `% reshape vector a to matrix A`
- `% A is composed of m rows and n columns`

Gray image

cmw2.bmp

```
I=imread('cmw2.bmp');  
imshow(I)
```

```
I=imread('cmw2.bmp')  
;  
image(I)
```





```
>> dir
```

```
.          .session  Shared  
..         Published cmw2.bmp
```

```
>> I=imread('cmw2.bmp');
```

```
>> image(I)
```



Sub-matrices

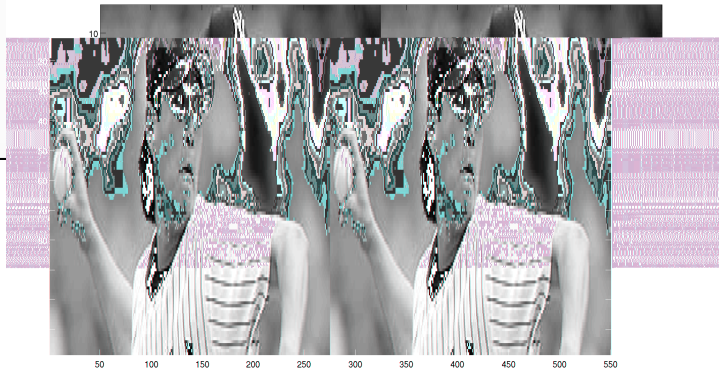
```
I=imread('cmw2.bmp');  
image(I);  
[m,n]=size(I);  
m2=ceil(m/2);  
n2=ceil(n/3*2);  
Is=I(1:m2,1:n2); figure  
imagesc(Is)  
colormap(gray)
```



```
...bmp');
```

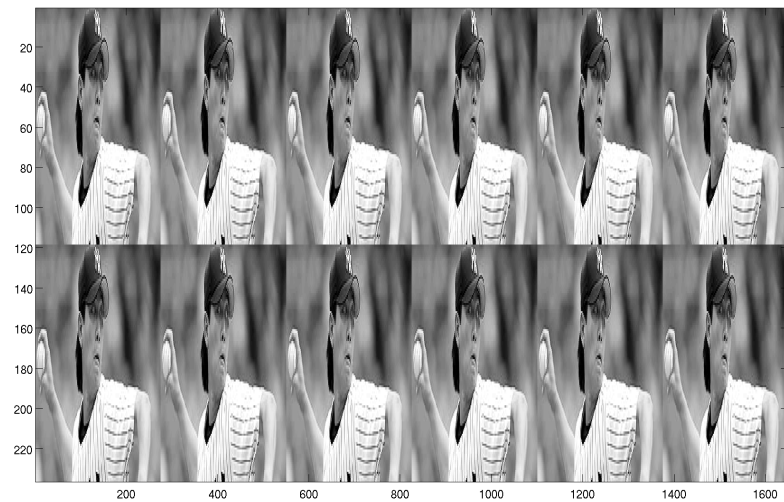
figure

```
>> I=imread  
image(I);  
[m,n]=size(  
m2=ceil(m/2  
n2=ceil(n/3  
Is=I(1:m2,1  
imagesc(Is)  
colormap(gr
```



Repeat

```
I2= repmat(I1,2,3);  
imagesc(I2);  
colormap(gray)
```



Repmat

- $A=[1\ 2;3\ 4]$
- $m=2;n=3;$
- `repmat(A,m,n)`
- % repeat A vertically m times
- % repeat the result horizontally n times



```
>> A=[1 2;3 4]
m=2;n=3;
repmat(A,m,n)
```

A =

```
1 2
3 4
```

ans =

```
1 2 1 2 1 2
3 4 3 4 3 4
1 2 1 2 1 2
3 4 3 4 3 4
```

Get Rows

- `A=reshape(1:16,4,4)`
- `a=3;b=1;c=2;`
- `A([a b c],:)`
- `% get rows specified by [a b c]`

```
>> A=reshape(1:16,4,4)
a=3;b=1;c=2;
A([a b c],: )
```

```
A =
```

```
    1     5     9    13
    2     6    10    14
    3     7    11    15
    4     8    12    16
```

```
ans =
```

```
    3     7    11    15
    1     5     9    13
    2     6    10    14
```


Get Columns

- `A=reshape(1:16,4,4)`
- `a=3;b=1;c=2;`
- `A(:,[a b c])`
- `% get columns specified by [a b c]`

```
>> A=reshape(1:16,4,4)
a=3;b=1;c=2;
A(:,[a b c])
```

```
A =
```

```
    1     5     9    13
    2     6    10    14
    3     7    11    15
    4     8    12    16
```

```
ans =
```

```
    9     1     5
        10     2     6
        11     3     7
        12     4     8
```

Row exchange

```
V=reshape(1:12,3,4)  
V([3 2],:)=V([2 3],:)
```

V =

1	4	7	10
3	6	9	12
2	5	8	11

Column exchange

```
V=reshape(1:12,3,4)
```

```
V(:,[2 3])=V(:,[3 2])
```

```
V =
```

1	7	4	10
2	8	5	11
3	9	6	12

Column sum

```
V=reshape(1:12,3,4)
```

```
V =
```

```
 1  4  7 10
 2  5  8 11
 3  6  9 12
```

```
>> sum(V)
```

```
ans =
```

```
 6 15 24 33
```

Column sum

```
V=reshape(1:12,3,4)
```

```
V =
```

```
 1  4  7 10
 2  5  8 11
 3  6  9 12
```

```
>> sum(V,1)
```

```
ans =
```

```
 6 15 24 33
```

Row sum

```
V=reshape(1:12,3,4)
```

```
V =
```

```
    1    4    7   10  
    2    5    8   11  
    3    6    9   12
```

```
>> sum(V,2)
```

```
ans =
```

```
    22  
    26  
    30
```

Mean of columns

```
V=reshape(1:12,3,4)
```

```
V =
```

```
 1  4  7 10
 2  5  8 11
 3  6  9 12
```

```
>> mean(V)
```

```
ans =
```

```
 2  5  8 11
```


Mean of rows

```
V=reshape(1:12,3,4)
```

```
V =
```

```
 1  4  7 10  
 2  5  8 11  
 3  6  9 12
```

```
>> mean(V,2)
```

Inversion

```
A=reshape(1:4,2,2);  
B=inv(A)
```

B =

-2.0000	1.5000
1.0000	-0.5000

Linear system

$$2x + y - z = 1$$

$$-3x - 2y + 5z = 0$$

$$x + y + z = 5$$

inv

```
A=[2 1 -1;-3 -2 5;1 1 1]; inv(A)*b  
b=[1 0 5]'
```

```
ans =
```

```
-1.6000
```

```
5.4000
```

```
1.2000
```

Left division

$$A=[2 \ 1 \ -1; -3 \ -2 \ 5; 1 \ 1 \ 1];$$

$$b=[1 \ 0 \ 5]'$$

- $A \setminus b$
- % Left division
- % $Ax=b$ could be solved by left division

Determinant

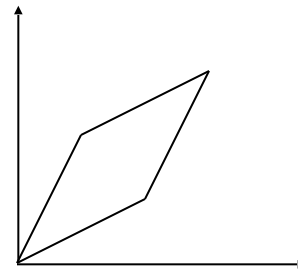
```
A=reshape(1:4,2,2);  
>> det(A)
```

```
ans =
```

```
-2
```

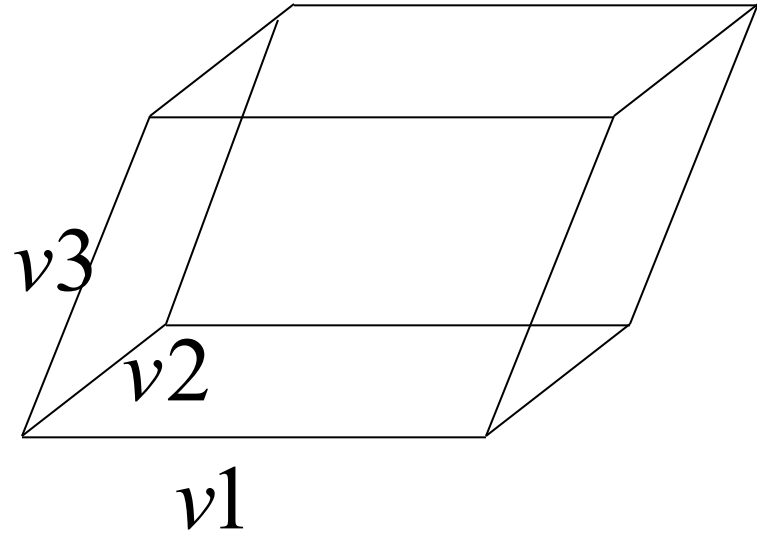
Area of a parallelogram

- v_1, v_2 denotes two vectors in a plane
- Calculate the area of the parallelogram determined by v_1 and v_2



Volume of a parallelepiped

$$A = [v_1; v_2; v_3]$$
$$\det(A)$$




```
reshape(randperm(9),3,3)
```

```
ans =
```

5	9	6
2	1	8
7	3	4

Append Horizontally

```
>> A=reshape(randperm(9),3,3);  
>> B=reshape(randperm(9),3,3);  
>> C=[A B]
```

C =

6	2	9	5	4	9
3	4	5	6	3	7
8	7	1	8	1	2

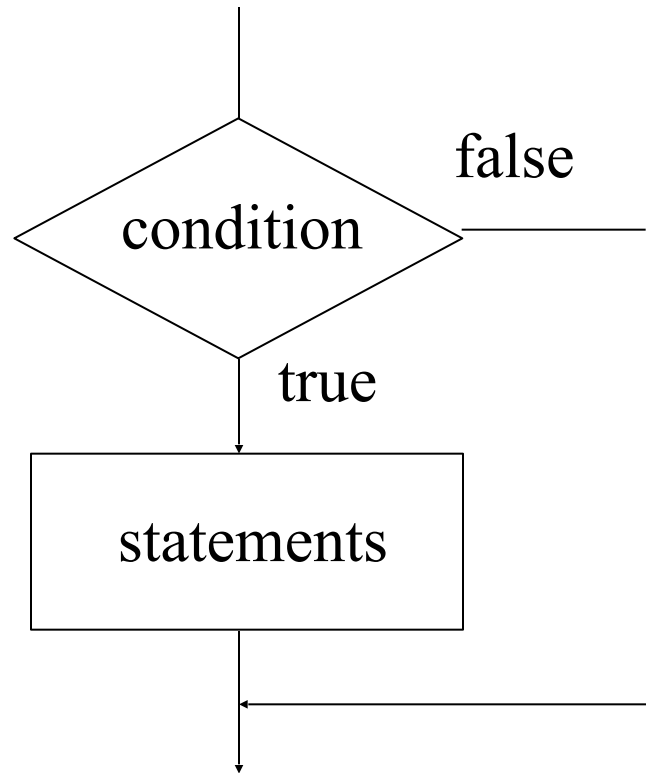
Append Vertically

```
>> C=[A;B]
```

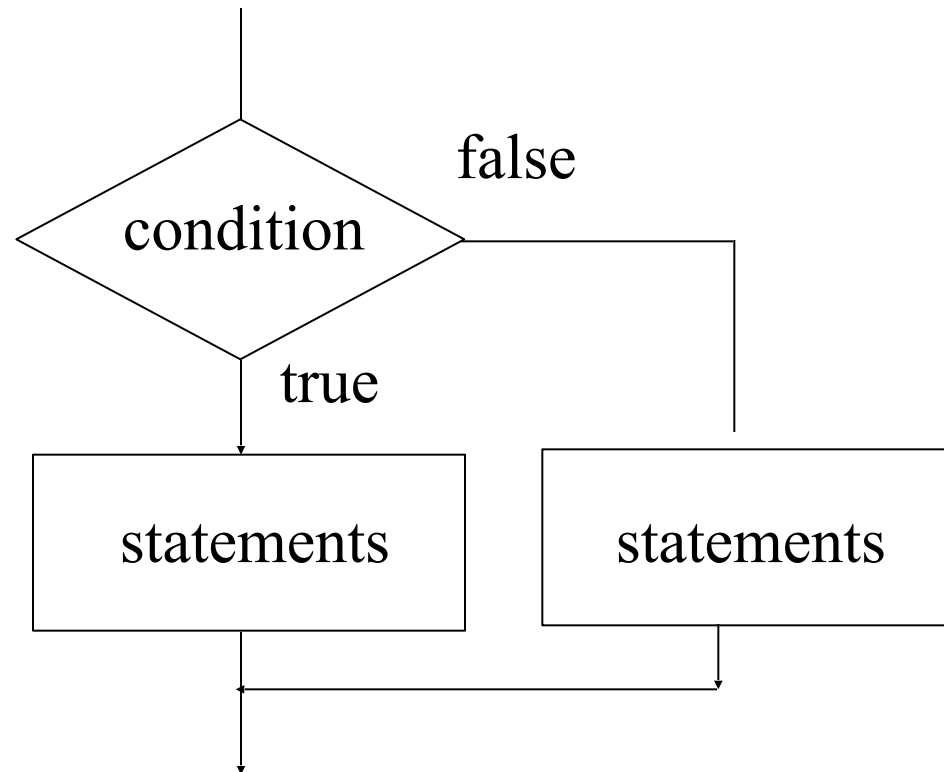
```
A =
```

6	2	9
3	4	5
8	7	1
5	4	9
6	3	7
8	1	2

if



if else



FOR Looping

- Series sum of ceils and floors
- Symbolic differentiation
- Counting ATCG
- Markov Chain

Summation

$$S(N) = \sum_{n=1}^N \left(\left\lfloor \frac{n^2}{5} \right\rfloor + \left\lceil \frac{2 * n}{3} \right\rceil \right)$$

$x = 1.5$



floor(x)



1

$x = 1.5$

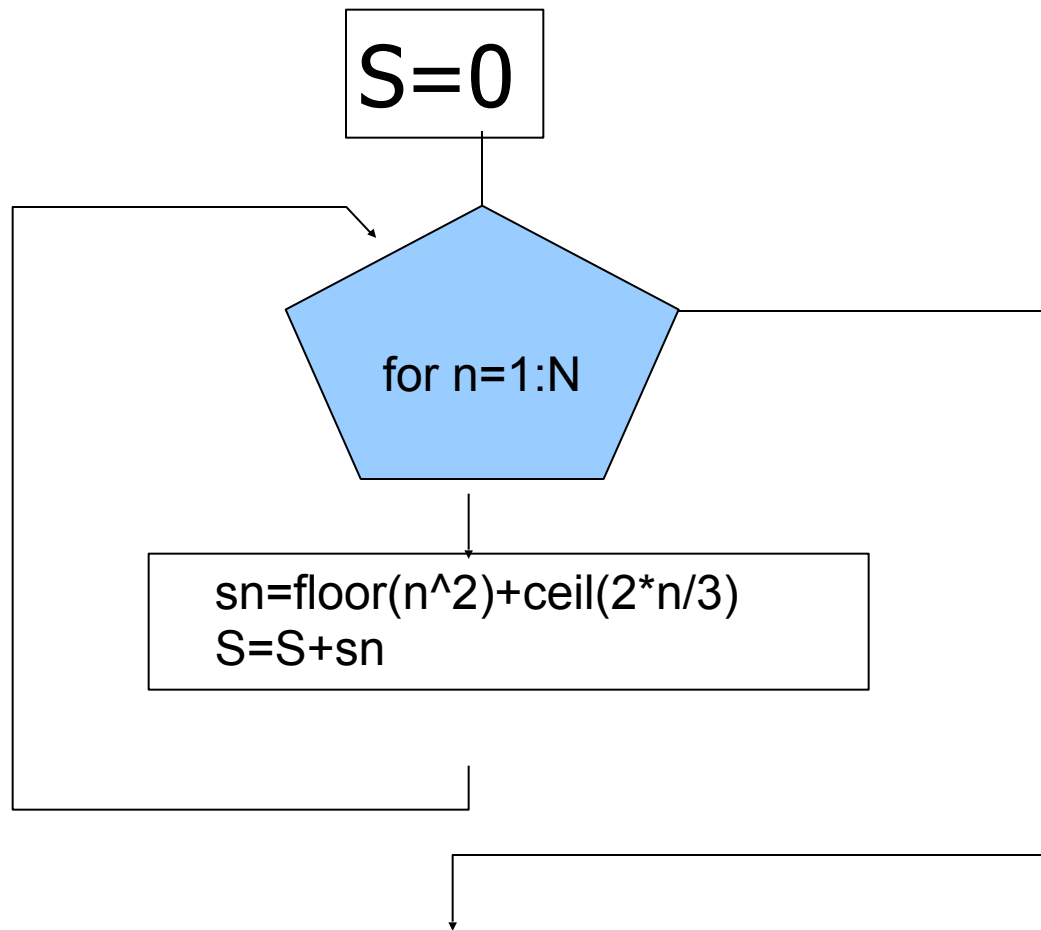


ceil(x)



2

Flow chart



Series

$$a_n = a_{n-1} + 2 * a_{n-2}$$

$$n \in \mathbb{N}$$

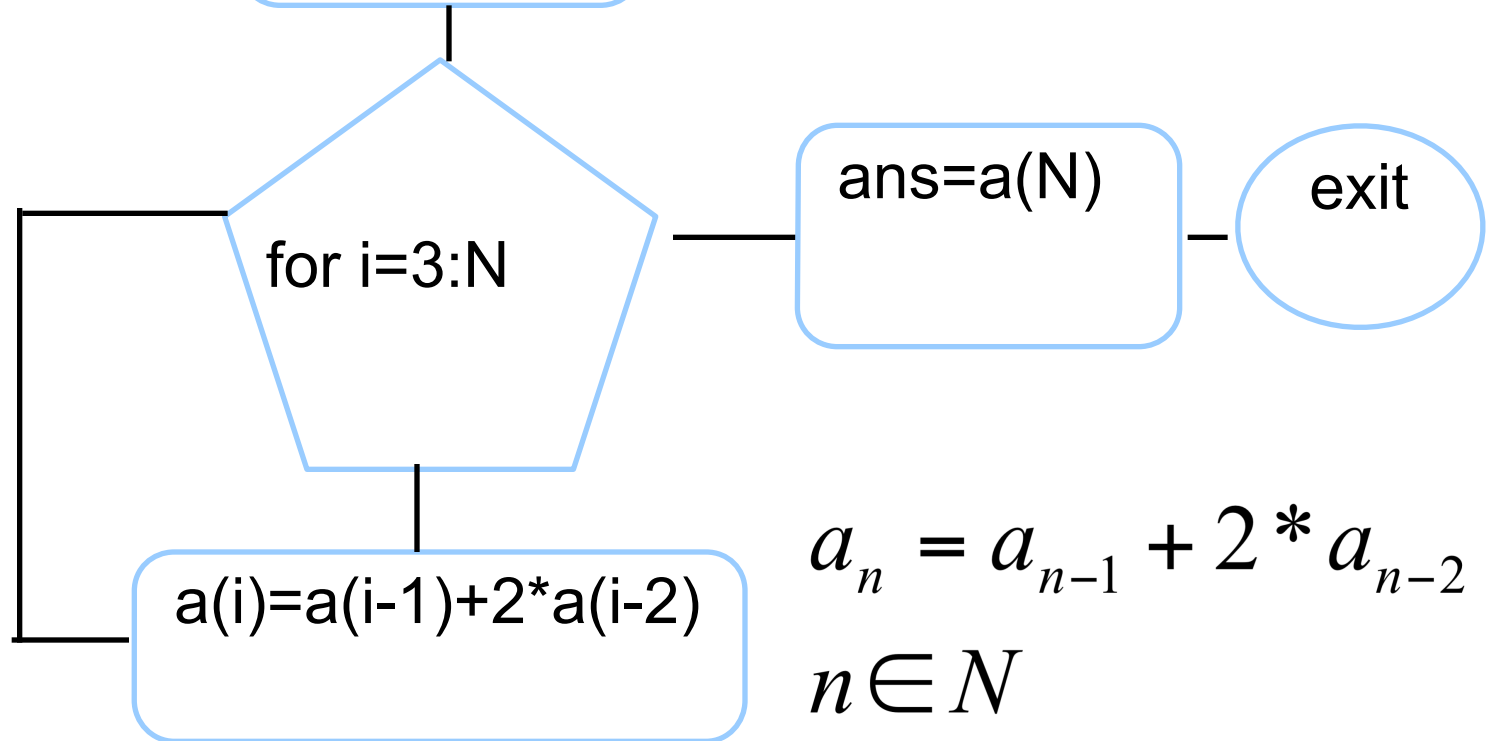
Problem

- INPUT: a_1, a_2 and n
- OUTPUT: a_n

- Draw a flow chart to determine a_n by a for-loop
- Write a matlab function to calculate a_n for given a_1 , a_2 and n

```
a(1)= a1;  
a(2)= a2;
```

```
function ans=fs(N,a1,a2)
```



$$a_n = a_{n-1} + 2 * a_{n-2}$$
$$n \in N$$

Symbolic Differentiation

```
>> x=sym('x');  
>> diff(tanh(x))
```

```
ans =
```

```
1-tanh(x)^2
```

Symbolic differentiation

```
>> x=sym('x');  
>> diff(x.^3+2*x)
```

```
ans =
```

```
3*x^2+2
```

Symbolic Differentiation

demo_diff.m

- Input a string, fstr
- Plot the function specified by fstr
- Plot the derivative of the given function

- MATLAB Web Server Demos

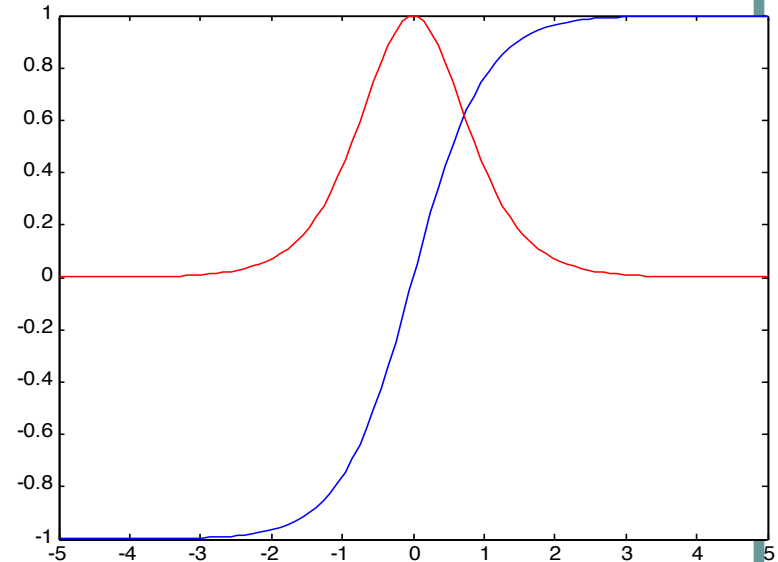
Example

function of x : $\tanh(x)$

$fx1 =$

Inline function:

$$fx1(x) = 1 - \tanh(x).^2$$



Histogram

- Sample space, $\{1,2,3,4,5,6\}$
- Count occurrences of possible outcomes in a series

A series

```
n=10;  
s=ceil(rand(1,10)*6);
```

Histogram

s =

6 2 4 3 6 5 3 1 5 3



```
count=hist(s,[1 2 3 4 5 6])
```



count =

1 1 3 1 2 2

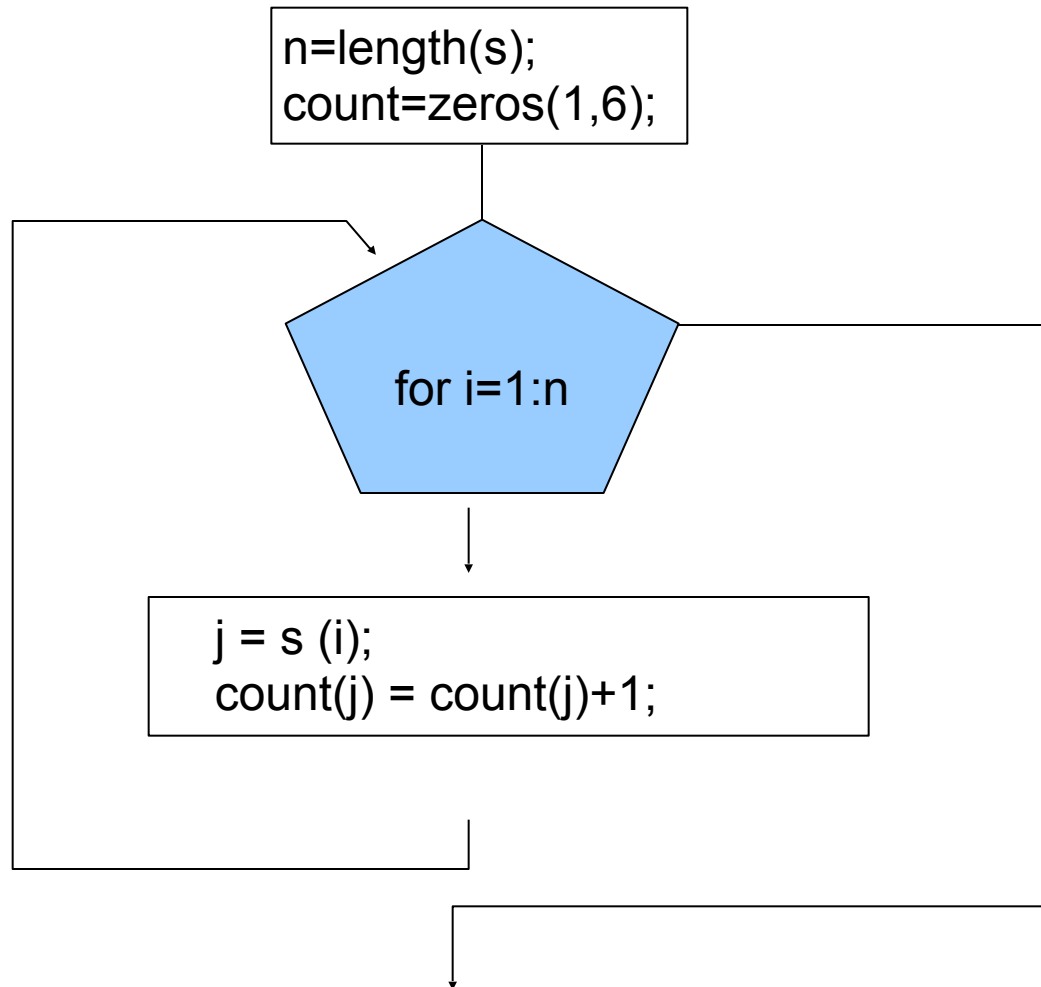
```
hist(s,[1 2 3 4 5 6])
```

Histogram

- INPUT: a series
- INPUT: [1 2 3 4 5 6]
- OUTPUT: occurrences of possible outcomes

Flow chart I :

function count=my_hist(s)



```
j=s(i);
```

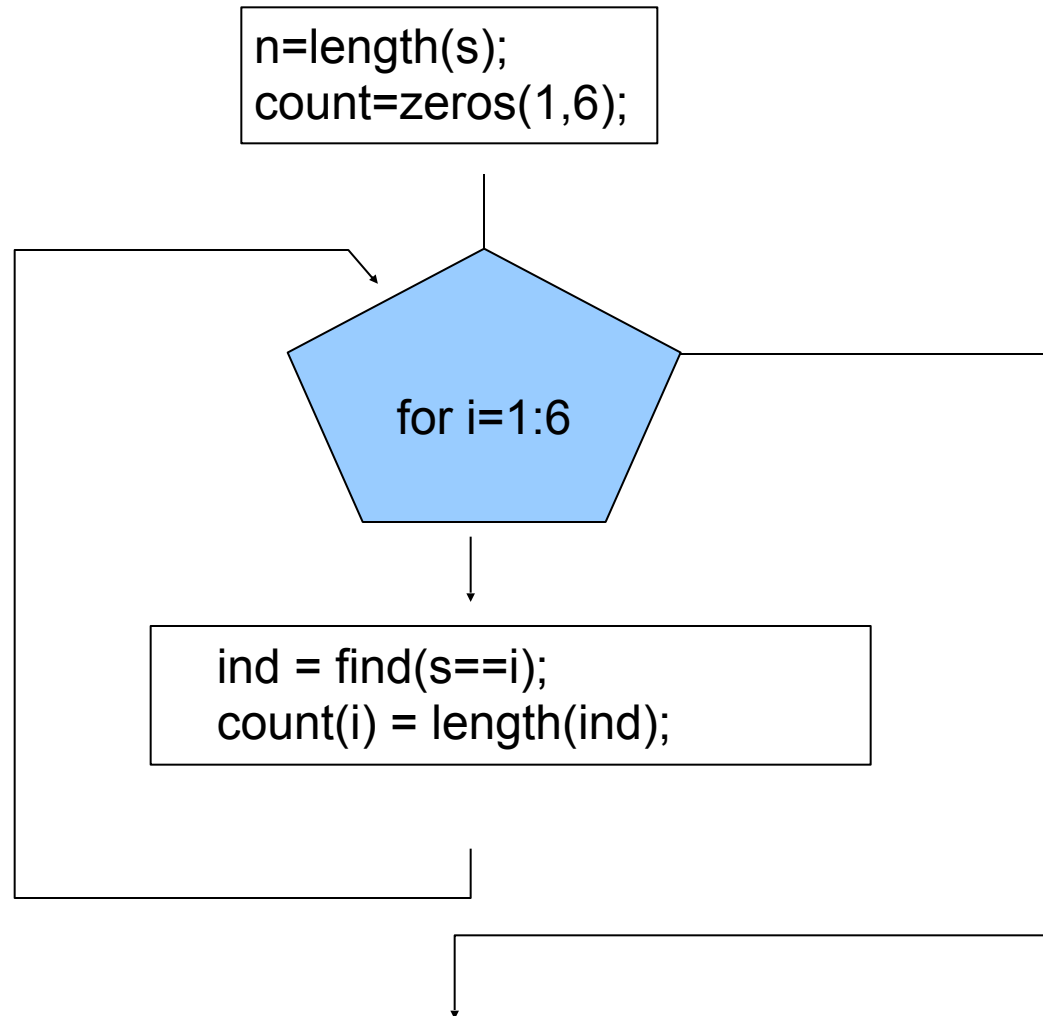
```
count(j) = count(j) + 1;
```

- j stores the ith element

- increase one for counting j

Flow chart II

function count=my_hist2(s)



```
function count=my_hist2(s)
```

```
n=length(s);  
count=zeros(1,6);  
for i=1:6  
    ind = find(s==i);  
    count(i) = length(ind);  
end
```

DNA sequence

ATCG sequence

load mitochondria
mitochondria(1:200)

mitochondria(1:200)

gatcacaggtctatcacctattaaccactcacggga
gctctccatgcatttggtatcttcgtctgggggggtgtgca
cgcgatagcattgcgagacgctggagccggagcac
cctatgtcgcagtatctgtctttgattcctgcctcattctatt
attatcgcacctacgttcaatattacaggcgaacata
cctacta

ss='gatcacaggtctatcacccattaaccact
cacgggagctctccatgcatttggtatttcgt
ctgggggggtgtgcacgcgatagcattgcga
gacgctggagccggagcacccctatgtcgc
agtatctgtctttgattcctgcctcattctatttt
atcgcacctacgttcaatattacaggcgaac
atacctacta'

ATCG

- Calculate probabilities of a,t,c,g in a given ATCG sequence.

$$\Pr(x = 'a') = ?$$

$$\Pr(x = 't') = ?$$

$$\Pr(x = 'c') = ?$$

$$\Pr(x = 'g') = ?$$

ATCG

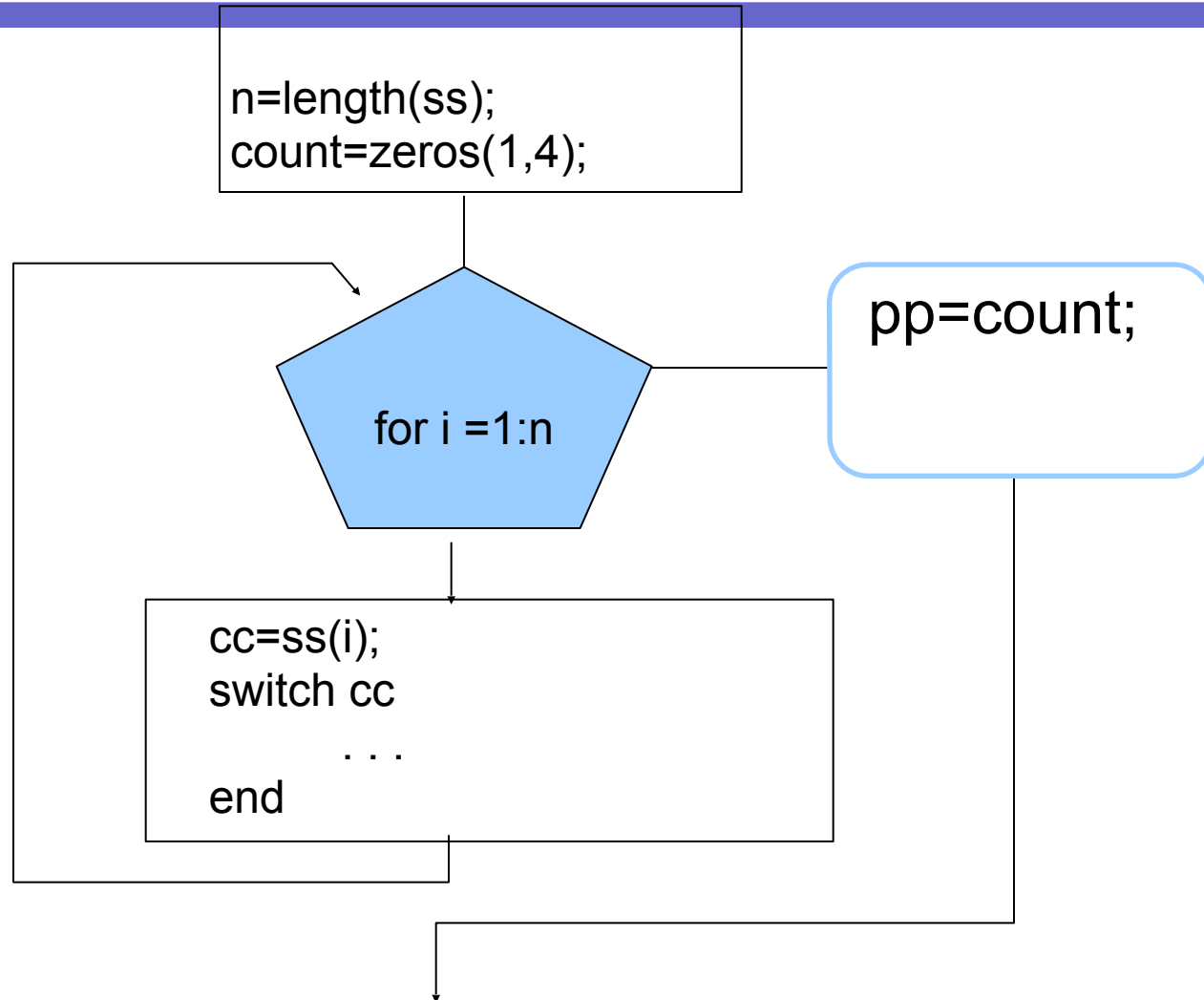
- Count numbers of a,t,c,g in a given ATCG sequence.

Input : ss

Output : numbers of 'a', 't', 'c' and 'g'

Flow chart

function pp=p_atcg(ss)



Switch

```
cc=ss(i);  
  switch cc  
    case 'a'  
      count(1)=count(1)+1;  
    case 't'  
      count(2)=count(2)+1;  
    case 'c'  
      count(3)=count(3)+1;  
    case 'g'  
      count(4)=count(4)+1;  
  end
```

function pp=p_atcg(ss)

```
n=length(ss);
count=zeros(1,4);
for i =1:n
    cc=ss(i);
    switch cc
        case 'a'
            count(1)=count(1)+1;
        case 't'
            count(2)=count(2)+1;
        case 'c'
            count(3)=count(3)+1;
        case 'g'
            count(4)=count(4)+1;
    end
end
```

- INPUT: ss and s
- Count occurrences of s within ss

ss =

s = 'g'

gtttcctact



```
ind=find(ss==s);  
length(ind)
```

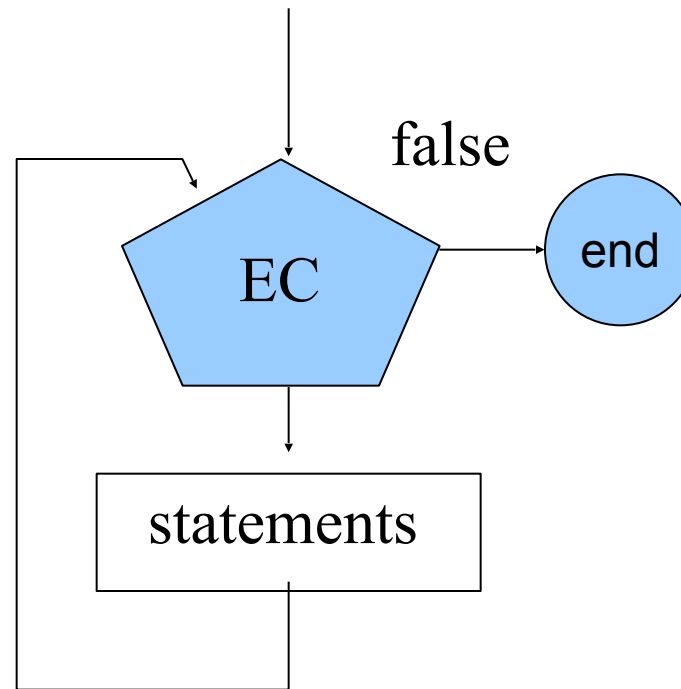
↓ ans =

While-Loop programming

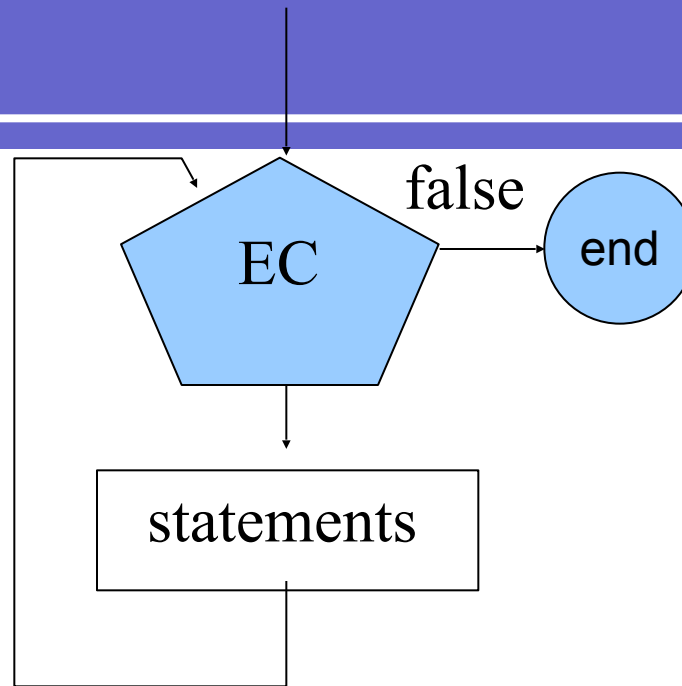
- While-loop flow chart
- Decimal to binary representations

While loop

```
while entry_condition  
    statements;  
end
```



Flow chart



- Execute body statements repeatedly until the entry condition fails
- The entry condition should eventually become false by iteratively executing body statements

Decimal to binary representations

- Input : positive decimal numbers
- Output : binary representations of the input

Mod

```
>> r=mod(10,3)
```

```
ans =
```

```
1
```

```
>> q=(10-r)/3
```

```
ans =
```

```
3
```


Quotient

```
>> r=mod(100,7)
```

```
ans =
```

```
2
```

```
>> q=(100-r)/7
```

```
ans =
```

```
14
```

Decimal to binary representation

- Let M be a positive decimal number
- Let b be an array of length n

$$b = [b_n, \dots, b_2, b_1]$$

where $b_n > 0$

- b denotes the binary representation of M if

$$M = b_n 2^{n-1} + b_{n-1} 2^{n-2} + \dots + b_1 2^0$$

$$C = 'b_n b_{n-1} \dots b_1'$$

Example

- $M > 0$
- $M=1, b_1=1$
- $M=5, C='101'$ $b = [b_3, b_2, b_1]$

$$M = b_3 2^2 + b_2 2^1 + b_1 2^0$$

$$\begin{array}{ccc} | & | & | \\ 1 & 0 & 1 \end{array}$$

Example

- $M=25$, $C: '11001'$ $b = [b_5, b_4, b_3, b_2, b_1]$

$$M = b_5 2^4 + b_4 2^3 + b_3 2^2 + b_2 2^1 + b_1 2^0$$

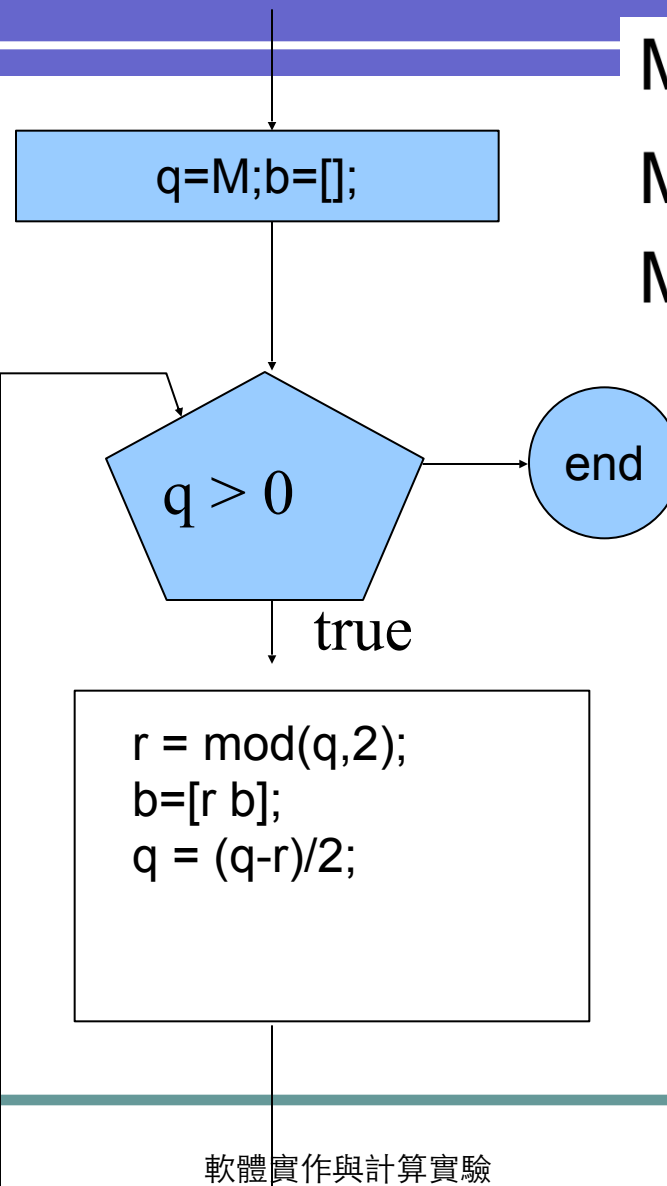
$$\begin{array}{cccccc} | & | & | & | & | \\ 1 & 1 & 0 & 0 & 1 \end{array}$$

- The problem is to find b for given M

Example

- $M > 0$ $M = b_n 2^{n-1} + b_{n-1} 2^{n-2} + \dots + b_1 2^0$
- $M=1, b_1=1$
- $M=5, b=[1 \ 0 \ 1]$
- $M=25, b=[1 \ 1 \ 0 \ 0 \ 1]$
- The problem is to find b for given M

Determine b

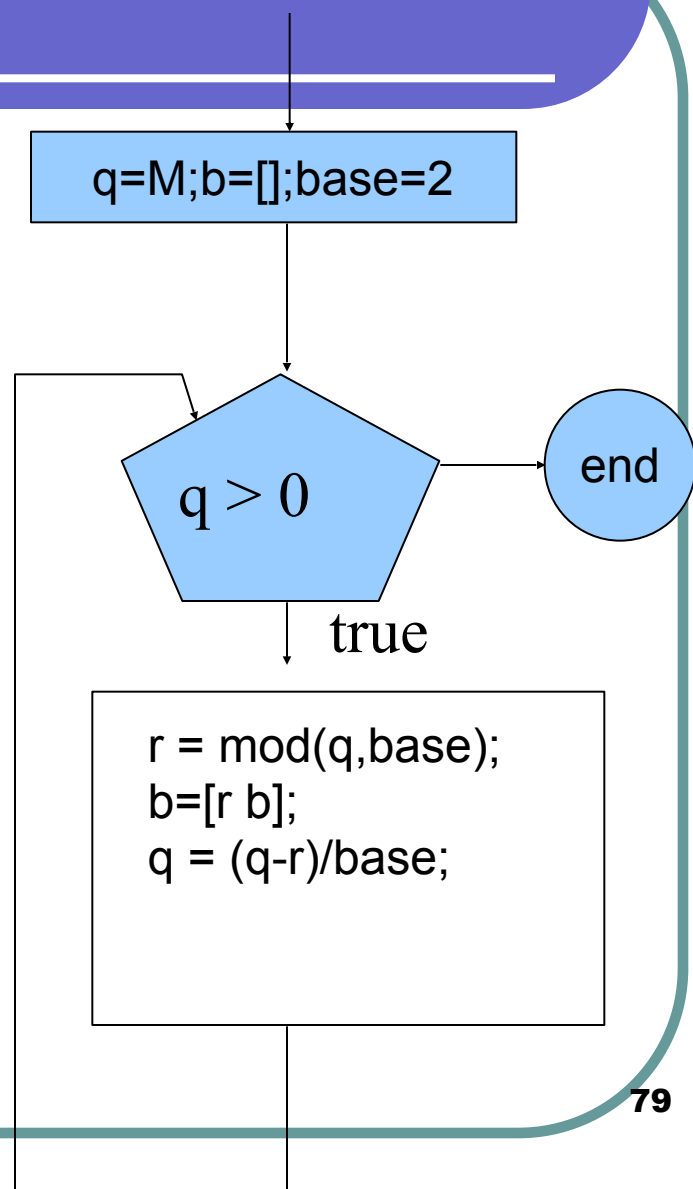
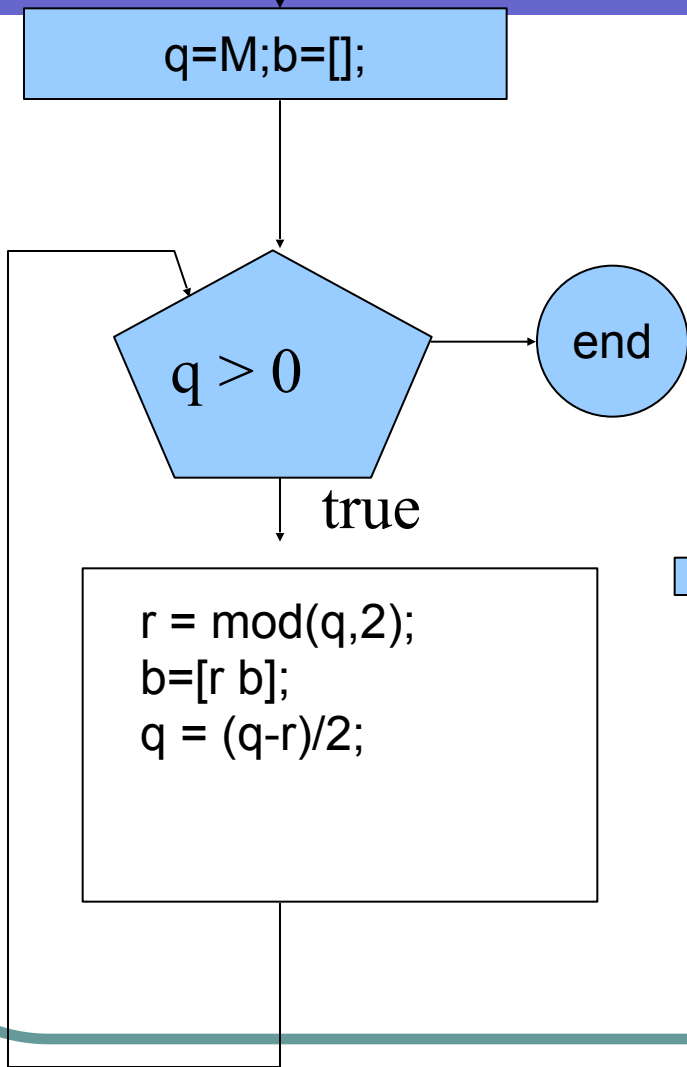


$M=1, b_1=1$

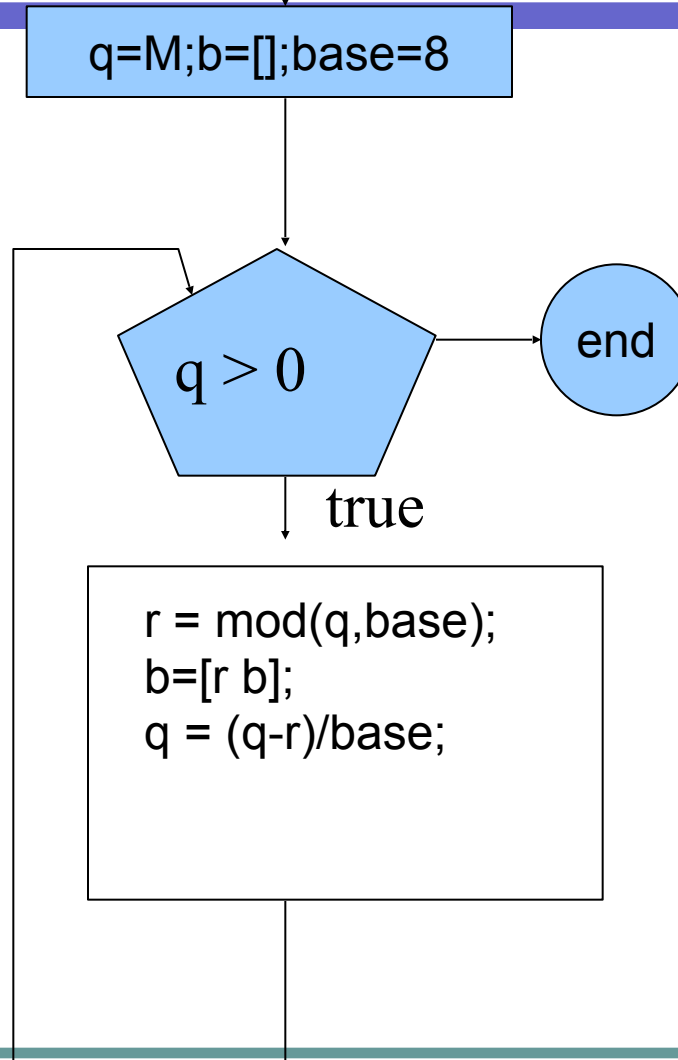
$M=5, b=[1\ 0\ 1]$

$M=25, b=[1\ 1\ 0\ 0\ 1]$

Change base



Decimal to Octal representation




```
>> dec2oct(7)
```

```
ans =
```

```
7
```

```
>> dec2oct(8)
```

```
ans =
```

```
1 0
```

```
>> dec2oct(18)
```

```
ans =
```

```
2 2
```

```
>> dec2oct(64)
```

```
ans =
```

```
1 0 0
```