6. Control Flow and if Statement

Control flow is extremely powerful; its lets past computations influence future operations. Matlab has several flow control constructs: `if , switch and case, for, while continue, break, try-catch, return.`

- **`if, else, and elseif`**

The `if` statement evaluates a logical expression and executes a group of statements when the expression is *true*. The optional `elseif` and `else` keywords provide for the execution of alternate groups of statements. An `end` keyword, which matches the `if`, terminates the last group of statements. The groups of statements are delineated by the four keywords—no braces or brackets are involved.

A simple example is to evaluate $f(x) = \begin{cases} x^2 & \text{if } x \le 2 \\ x^3 & \text{if } x > 2 \end{cases}$ at a given points:

```
if x<=2   f=x^2
else      f=x^3
end
```

Here is A second model using `elseif`, to determine if a certain amount is positive negative or zero.

```
x = input(' Type x = ');
if (x>0)      disp('x is positive')
elseif (x<0) disp('x is negative')
else     disp('x is zero')
end
```

Often we need some relational or logical operators to companion `if` statement. A relational operator compares two numbers by determining whether a comparison is *true* or *false*. Relational operators are shown in the following table:

Relational Operators		Logical Operators	
<	Less than	&&	Logical AND
<=	Less than or equal to	||	Logical OR
>	Greater than	&	Logical AND for arrays
>=	Greater than or equal to	|	Logical OR for arrays
==	Equal to	~	Logical NOT
~=	Not equal to		

- **for loop**

for loop allow a group of commands to be repeated a fixed, predetermined number of times. For example:

```
for i = 1:4
   x(i) = i^2
end

for k=2:5:20, y=k^3-7, end

for x=[2 0 3], y=x^3-5*x, end
```

hear an example for nested loops

```
clear
m=2; n=3;
for i = 1:m
   for j = 1:n
      H(i,j) = 1/(i+j);
   end
end
H
```

Here is another example showing if, else, and elseif and for

```
k=5;
for m = 1:k
    for n = 1:k
        if m == n
            a(m,n) = 2;
        elseif abs(m-n) == 2
            a(m,n) = 1;
        else
            a(m,n) = 0;
        end
    end
end
```

More exemples are :

a. Write a script that removes the numbers divisible by 4 from any array x. Assume the array x is given by $x = [-8,0,2,5,7,-20,4,6,9]$

```
x = [-8,0,2,5,7,-20,4,6,9];
y=[];
for n=1:length(x)
    if x(n)/4-fix(x(n)/4)~=0 y=[y,x(n)];
    end
end
y
```

Here fix(x(n)/4), equals the integer part of x(n)/4, so fix(x(n)/4)~=0 is satisfied if x(n) is divisible by 4. Equivalently, mod(a,b) may be used that equals zero if a is divisible b. See the following script.

```
clear
x=[-8,0,2,5,7,-20,4,6,9];
y=[];
for n=1:length(x)
    if  mod(x(n),4)~=0 y=[y,x(n)];
    end
end
y
```

The result is

```
y =
   2   5   7   6   9
```

b. Write a script to examine if a given number is a prime number or not. Use input command to read the integer,. Don't use factor or isprime commands.

```
i = input('Type an integer greater than 1: ');
a = 1;
for n = 2:i-1
    if i/n == fix(i/n)  a = 0;
        break
    end
end
if a == 0 fprintf('No %3.0f is not prime number.\n',i)
else      fprintf('Yes %3.0f is prime number.\n',i)
end
```

c. Develop an M-file that evaluate the following series:

$$f(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!}$$

The values of x and n are to be input. The series is Maclaurin expansion of e^x. Test your function for $x = 0.3, 0.5, 1$, and $n = 2, 5, 10$. Then compare with e^x.

```
clear
x = input('x = ');
n = input('n = ');
s = 1;  p = 1;
for k = 1:n
    p = p*x/k;
    s = s+p;
end
f = s
error = abs(exp(x)-f)
```

- ## while loops

A while loop evaluates a group of statements an indefinite number of times such as

```
c = 0; i=1;
while c==0
   i=i+2
   s=1/i
   if s<=0.1 c=1
   end
end
```

A second example is to find 10 partial sums and the sum, if it is convergent, of the series

$$\sum_{n=1}^{\infty} \frac{10}{3^n}$$

```
clear
a(1) = 10/3;  s(1) = a(1);
for k=2:10
    a(k)=10/3^k;
    s(k)=sum(a);
end
s
clear
k = 0;  a = 10/3;  s = a;
while a>eps & k<100
    s = s+a;
    k = k+1;
    a = 10/3^k;
end
s
```

- **`break`**

The `break` statement lets you exit early from a `for` loop or `while` loop. In nested loops, `break` exits from the innermost loop only.

Here is a complete program, illustrating `while`, `if`, `else`, and `end`, which uses interval bisection to find a zero of a polynomial:

```
a = 0; fa = -Inf;
b = 3; fb = Inf;
while b-a > eps*b
   x = (a+b)/2;
   fx = x^3-2*x-5;
   if fx == 0
      break
   elseif sign(fx) == sign(fa)
      a = x; fa = fx;
   else
      b = x; fb = fx;
   end
end
x
```

Exercise

1. Use a for loop to print the even numbers between 10 and 20.

2. Use a for loop to set up the vector with entries (4,5,6,7,8,9).

3. Write a script to print out the prime numbers less than 100 in increasing order. Print out the total number of the prime numbers found, a list of the prime numbers, and the sum of the prim numbers less than 100. (Don't use factor or isprime commands)

4. Find the sum of all primes less than 10. Use isprime command.

5. Enter the expressions $f = \frac{x^3}{x^2+\sin x}$, and $g = \frac{x-\sin x}{x^2}$. Use an if statement to determine whether $f(3) > g(3), f(3) = g(3)$, or $f(3) < g(3)$.

6. Develop M-file that evaluate the following series:

$$f(x) = x - \frac{x^2}{2} + \frac{x^3}{3} + \cdots + (-1)^{n+1}\frac{x^n}{n}$$

The values of x and n are to be input. The series is Maclaurin expansion of

$\log(1+x)$. Test your function for $x = 0.3, 0.5, 1,\ n = 2,5,10$ and compare with the exact value.

7. Use for loop to find the first 4 derivatives of $\log x$.

8. Use for loop to find the first 4 derivatives of $\frac{\sin x}{\cos x + \tan x}$ at $x = \pi/3$.

9. Use for loop to find the first derivative of $\log x$ at x=1,2,3,….10.

10. Use for loop to find the first 4 derivatives of $e^x \sin x$.

11. Using a simple while loop, write a script to sum the series 1 + 2 + 3 + ... such that the sum is as large as possible without exceeding 100. The program should display how many terms are used in the sum.

12. Compute 20! using two ways, one with a for loop and one without. Find out about the MATLAB built-in function factorial and use it to check your answers.

13. Write a script that takes as input an integer n and creates the $n \times n$ matrix A with (i,j)th component given by $A(i,j) = \sin(1/(i + j - 1))$.

14. Write a script that takes as input three numbers a, b and c and prints out either the solutions of the quadratic equation $ax^2 + bx + c = 0$, when these solutions are real, or a message indicating that the solutions are not real.

15. Using a simple while loop, write a script to sum the series 1 + 3 + 5 ... such that the sum is as large as possible without exceeding 50. The program should display how many terms are used in the sum.

16. Using a simple while loop, write a script to sum the series 2 + 4 + 6 ... such that the sum is as large as possible without exceeding 80. The program should display how many terms are used in the sum.