1. Fent us SUM

1+2+...+n Vn

```
n = 10
```

```
n =
    10
```

```
%n = input('entra valor n = ')
sum(1:n)
```

```
ans =
    55
```

1+2^p+....+n^p   Vn, Vp

```
n = 2
```

```
n =
     2
```

```
p = 10
```

```
p =
    10
```

```
%n = input('entra valor n = ')
%p = input('entra valor p = ')
sum((1:n).^p)
```

```
ans =
        1025
```

5 Escriviu una **funció** d'argument m que calculi

$$\sqrt{12}\sum_{n=0}^{m}\frac{(-1/3)^n}{2n+1}$$

Calculeu el valor per $m = 5$, $m = 10$, i $m = 20$ i compareu el resultat amb π.

```
resultat = ejercicio5(5)
```

```
resultat =
          2.41699868665013
```

6 Avalueu les funcions

$$f(x) = \sqrt{x^2+1} - 1\,, \quad g(x) = x^2/\sqrt{x^2+1} + 1$$

per a la successió de valors de $x_n = 8^{-n}$, $n \geq 1$. Encara que $f(x) = g(x)$, l'ordinador dóna resultats diferents. Quins resultats són de fiar i quins no? Per què? Justifiqueu la vostra resposta.

```
resultat = ejercicio6(20)
```

```
g = function_handle with value:
    @(x)x.^2./(sqrt(x.^2+1)+1)
resultat = 21×4
                    0                    1    0.414213562373095 ···
                    1                0.125  0.00778221853731864
                    2             0.015625 0.000122062862828676
                    3          0.001953125   1.9073468138231e-06
                    4       0.000244140625  2.98023219436061e-08
                    5      3.0517578125e-05  4.65661287307739e-10
                    6    3.814697265625e-06  7.27595761418343e-12
                    7  4.76837158203125e-07  1.13686837721616e-13
                    8  5.96046447753906e-08  1.77635683940025e-15
                    9  7.45058059692383e-09                     0
    ⋮
```

```
resultat = array2table(resultat, ...,
    'VariableNames',{'n','xn','f','g'})
```

resultat = 21×4 table

	n	xn	f	g
1	0	1	0.414213562...	0.414213562...
2	1	0.125	0.007782218...	0.007782218...
3	2	0.015625	0.000122062...	0.000122062...
4	3	0.001953125	1.907346813...	1.907346813...
5	4	0.000244140625	2.980232194...	2.980232194...
6	5	3.051757812...	4.656612873...	4.656612871...
7	6	3.814697265...	7.275957614...	7.275957614...
8	7	4.768371582...	1.136868377...	1.136868377...
9	8	5.960464477...	1.776356839...	1.776356839...
10	9	7.450580596...	0	2.775557561...
11	10	9.313225746...	0	4.336808689...
12	11	1.164153218...	0	6.776263578...
13	12	1.455191522...	0	1.058791184...
14	13	1.818989403...	0	1.654361225...
15	14	2.273736754...	0	2.584939414...

	n	xn	f	g
16	15	2.842170943...	0	4.038967834...
17	16	3.552713678...	0	6.310887241...
18	17	4.440892098...	0	9.860761315...
19	18	5.551115123...	0	1.540743955...
20	19	6.938893903...	0	2.407412430...
21	20	8.673617379...	0	3.761581922...

Calcular el valor x_{10} del mètode iteratiu següent:

$$x_k = \frac{1}{2}\left(x_{k-1} + \frac{2}{x_{k-1}}\right) \qquad k \geq 1 \ i \ x_0 = 2\,.$$

Fent ús de les instruccions

Bucle `for`

Bucle `while`

En tots els casos comparar el resultat obtingut amb el valor $\sqrt{2}$

```
n = 10;
k = 2;
x(1) = 2;
while (k <=n && x(k-1) > 5e-14)
    x(k) = 0.5*(x(k-1)+(2/x(k-1)));
    k = k+1;
end
R = [1:length(x); x;abs(x-sqrt(2));abs(x-sqrt(2))/abs(sqrt(2))]'
```

```
R = 18×4
         1                    2     0.585786437626905 ···
         2                  1.5    0.0857864376269049
         3     1.41666666666667   0.00245310429357137
         4     1.41421568627451   2.12390141451912e-06
         5     1.41421356237469   1.59472435257157e-12
         6     1.41421356237309   2.22044604925031e-16
         7     1.41421356237309   2.22044604925031e-16
         8     1.41421356237309   2.22044604925031e-16
         9     1.41421356237309   2.22044604925031e-16
        10     1.41421356237309   2.22044604925031e-16
    ⋮
```

```
n = 15;
x2(1) = 2;
for k = 2:1:n
    x2(k) = 0.5*(x2(k-1)+(2/x2(k-1)));
    if x2(k) < 5e-14
       break
    end
end

R2 = [1:length(x2); x2;abs(x2-sqrt(2));abs(x2-sqrt(2))/abs(sqrt(2))]'
```

R2 = 15×4

1	2	0.585786437626905	⋯
2	1.5	0.0857864376269049	
3	1.41666666666667	0.00245310429357137	
4	1.41421568627451	2.12390141451912e-06	
5	1.41421356237469	1.59472435257157e-12	
6	1.41421356237309	2.22044604925031e-16	
7	1.41421356237309	2.22044604925031e-16	
8	1.41421356237309	2.22044604925031e-16	
9	1.41421356237309	2.22044604925031e-16	
10	1.41421356237309	2.22044604925031e-16	

⋮

9 Definim el nombre e com $e = \sum_{k=0}^{\infty} \frac{1}{k!}$. Per calcular-ne una aproximació considerem el mètode iteratiu definit per

$$x_k = x_{k-1} + \frac{1}{k!}, \quad k \geq 1, \quad x_0 = 1$$

Calculeu els 20 primers termes de la recurrència, compareu els vostres resultats amb el valor $exp(1)$ retornat per Matlab.

```
x(1) = 1;
p = 1;
k = 1;
nmax = 20;
tol(k) = abs(x(k)-exp(1))
```

tol = 1×18

1.71828182845905	0.718281828459046	0.218281828459046	⋯

```
while(tol(k) > eps) && (k<nmax)
    p = p*k;
    k = k+1;
    x(k) = x(k-1) +1/p;
    tol(k) = abs(x(k)-exp(1));
end
R = [1:length(x); x; tol]'
```

```
R = 18×3
     1                  1     1.71828182845905
     2                  2     0.718281828459046
     3                2.5     0.218281828459046
     4   2.66666666666667     0.051615161792379
     5   2.70833333333333     0.0099484951257125
     6   2.71666666666667     0.00161516179237919
     7   2.71805555555556     0.000226272903490088
     8   2.71825396825397     2.78602050771681e-05
     9   2.71827876984127     3.05861777549765e-06
    10   2.71828152557319     3.02885853287194e-07
  ⋮
```

Polinomis

x^3-3*x^2+3*x+3

= ((x-3)*x+3)x+3 -> Regla de Horner

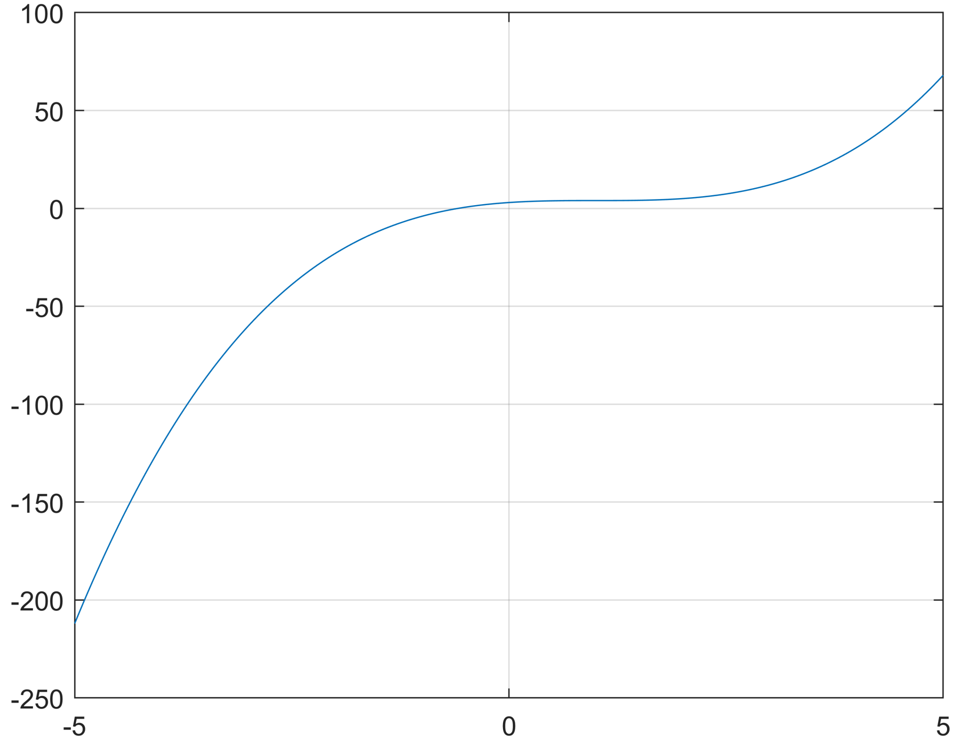
```
p = [1 -3 3 3]
```

```
p = 1×4
     1    -3     3     3
```

```
x = linspace(-5,5,200)
```

```
x = 1×200
    -5   -4.94974874371859   -4.89949748743719 ···
```

```
%%polyval utilitza horner
y = polyval(p,x);
plot(x,y)
grid
```




```
roots(p)
```

```
ans = 3×1 complex
     1.7937005259841 +      1.3747296369986i
     1.7937005259841 -      1.3747296369986i
    -0.587401051968199 +                  0i
```

```

```

Polinomi de Wilkinson

```
roots(1:10)
```

```
ans = 9×1 complex
    -1.33859118589308 +                  0i
    -1.09704408762781 +    0.756988433625517i
    -1.09704408762781 -    0.756988433625517i
   -0.465689376213834 +     1.22950234299199i
   -0.465689376213834 -     1.22950234299199i
    0.310290909092414 +     1.24228190323516i
    0.310290909092414 -     1.24228190323516i
    0.921738147695767 +    0.796363824394796i
    0.921738147695767 -    0.796363824394796i
```

```
p = poly(1:10)'
```

```
p = 11×1
           1
```

```
-55
1320
-18150
157773
-902055
3416930
-8409500
12753576
-10628640
⋮
```

```
q = p
```

```
q = 11×1
         1
       -55
      1320
    -18150
    157773
   -902055
   3416930
  -8409500
  12753576
 -10628640
     ⋮
```

```
format long g
q(2)=q(2)+1/2^(13)
```

```
q = 11×1
                        1
        -54.9998779296875
                     1320
                   -18150
                   157773
                  -902055
                  3416930
                 -8409500
                 12753576
                -10628640
     ⋮
```

```
roots(q)
```

```
ans = 10×1 complex
    9.71680694440454 +     0.423492822039221i
    9.71680694440454 -     0.423492822039221i
    7.39701333900533 +     0.749263119716399i
    7.39701333900533 -     0.749263119716399i
    5.67612069756221 +                     0i
    5.10313498919566 +                     0i
    3.99274462886153 +                     0i
    3.00023859699748 +                     0i
    1.99999844991441 +                     0i
    1.00000000033645 +                     0i
```

Equaciones lineales

$$a)\begin{cases} x + 2y = 3 \\ 0.499x + 1.001y = 1.5 \end{cases}$$

```
format short g
A = [1 2; 0.499  1.001]
```

```
A = 2×2
           1           2
       0.499       1.001
```

```
B = [3;1.5]
```

```
B = 2×1
           3
         1.5
```

```
SOL = A\B
```

```
SOL = 2×1
           1
           1
```

```
det(A)
```

```
ans =
       0.003
Error using det
Matrix must be square.
```

```
cond(A)
```

```
function [resultat] = ejercicio5(m)
resultat = 1;
p = 1;
    for n = 0:1:m
         p = p * (-1/3);
         aux = p/(2*n+1);
         resultat = resultat + aux;
    end
resultat = resultat * sqrt(12);
```

```
end
```

```
function [resultat] = ejercicio6(n)
k = 0:n;
x = 8.^(-k);
f = @(x)sqrt(x.^2+1)-1;
g = @(x)x.^2./(sqrt(x.^2+1)+1)
resultat = [k;x;f(x);g(x)]';
end
```