

```
%{
problema 1_modalitatea 1:
cream separat o functie de tip M cu numele functia4.m
o apelam scriind:
%}
y1 = functia4(1)
y2 = functia4(-2)
y3 = functia4([2 3])
y4 = functia4([1 2 ; 3 4])

% problema 1_modalitatea 2: cream o functie de tip inline
functia4b = inline('x.^2+exp(x)')
z1 = functia4b(1)
z2 = functia4b(-2)
z3 = functia4b([2 3])
z4 = functia4b([1 2 ; 3 4])

% problema 1_modalitatea 3: cream o functie de tip anonymous (folosim un function handle)
functia4c = @(x) x.^2+exp(x)
w1 = functia4c(1)
w2 = functia4c(-2)
w3 = functia4c([2 3])
w4 = functia4c([1 2 ; 3 4])

%{
problema 2: cream separat o functie de tip M cu numele functia5.m
o apelam scriind, de exemplu:
%}
H5 = functia5(5)
H7 = functia5(7)

%{
problema 3: cream separat o functie de tip M cu numele functia6.m
o apelam scriind, de exemplu:
%}
M5 = functia6(5)
M7 = functia6(7)
```

```
%{  
problema 4: trebuie scrisa pentru n=10, n=50, n=100 etc.  
pentru a vedea cum arata eroarea  
%}  
k = 1:10; S_n = sum(1./k.^2)  
suma = pi^2/6  
eroarea = abs(S_n - suma)
```

```
%{  
problema 5: trebuie scrisa pentru n=4, n=10, n=100, n=1000 etc.  
pentru a vedea cum arata eroarea  
%}  
x1 = -.5; val1 = functia7(x1,4)  
suma1 = log(x1+1)  
eroarea1 = abs(val1 - suma1)
```

```
x2 = .5; val2 = functia7(x2,4)  
suma2 = log(x2+1)  
eroarea2 = abs(val2 - suma2)
```

```
x3 = 1; val3 = functia7(x3,4)  
suma3 = log(x3+1)  
eroarea3 = abs(val3 - suma3)
```

```
x4 = -1; val4 = functia7(x4,1000)  
suma4 = log(x4+1)  
eroarea4 = abs(val4 - suma4)
```

```
% problema 6_modalitatea 1:  
%{  
definim, mai intai, functia8, intr-un fisier de tip M, care defineste algoritmul de calcul  
(metoda de aproximare a integralei (alta decat cea folosita de MATLAB))  
careia vrem sa ii aproximam integrala  
%}
```

```
% apoi introducem efectiv functia pe care vrem sa o integram;  
% alegem functia  $\exp(-x^2/2)$ 
```

```

fct_test = inline('exp(-x.^2/2)')
a = input('Introduceti limita a de la integrala definita = ')
b = input('Introduceti limita b de la integrala definita = ')
n = input('Introduceti numarul de pasi in aproximarea cu metoda trapezelor = ')
I_aprox = functia8(a,b,n,fct_test)
I = quad('exp(-x.^2/2)',a,b)
%{
calculez si integrala cu metoda MATLAB, adica, de exemplu, folosind comanda quad
(vine de la quadrature: adica metoda de a determina o aproximare numerica a
integralei definite prin metoda cuadraturilor (i.e. calcul de arii cunoscute))
%}
fprintf('Am aproximat integrala definita, dintr-o functie data,\nde la a = %.2f la b = %.2f cu un numar de n
= %.0f de pasi\n',a,b,n)
fprintf('\n')
fprintf('Integrala calculata de noi prin metoda trapezelor are valoarea %.8f \niar valoarea ei calculata de
MATLAB prin comanda quad este %.8f\n',I_aprox,I)
fprintf('\n')
fprintf('Diferenta dintre cele doua valori/metode de calcul este de %.12f\n',abs(I-I_aprox))
fprintf('\n')

```

```

% problema 6_modalitatea 2:
%{
definim, mai intai, functia9, intr-un fisier de tip M, care defineste algoritmul de calcul
(metoda de aproximare a integralei (alta decat cea folosita de MATLAB))
careia vrem sa ii aproximam integrala
(functia9 este functia8 scrisa in alta modalitate)
%}

```

```

% apoi introducem efectiv functia pe care vrem sa o integram;
% alegem functia exp(-x.^2/2)

```

```

fct_test2 = inline('exp(-x.^2/2)')
a = input('Introduceti limita a de la integrala definita = ')
b = input('Introduceti limita b de la integrala definita = ')
n = input('Introduceti numarul de pasi in aproximarea cu metoda trapezelor = ')
I_aprox2 = functia9(a,b,n,fct_test2)
I = quad('exp(-x.^2/2)',a,b)
%{
calculez si integrala cu metoda MATLAB, adica, de exemplu, folosind comanda quad
%}
fprintf('Am aproximat integrala definita, dintr-o functie data,\nde la a = %.2f la b = %.2f cu un numar de n
= %.0f de pasi\n',a,b,n)
fprintf('\n')
fprintf('Integrala calculata de noi prin metoda trapezelor are valoarea %.8f \niar valoarea ei calculata de
MATLAB prin comanda quad este %.8f\n',I_aprox2,I)
fprintf('\n')

```

```
fprintf('Diferenta dintre cele doua valori/metode de calcul este de %.12f\n',abs(I-I_aprox2))  
fprintf('\n')
```

```
% problema 7_modalitatea 1:
```

```
L = input('Introduceti pragul L = ');
```

```
s = 0; k = 0;
```

```
while s < L - (k+1)^2
```

```
    k = k+1;
```

```
    s = s + k^2;
```

```
end
```

```
fprintf('Suma de patrute de valoare maxima care este mai mica decat L=%.2f este %.2f\n',L,s)
```

```
fprintf('\n')
```

```
fprintf('Ea este obtinuta pentru adunarea primilor %.0f termeni\n',k)
```

```
fprintf('\n')
```

```
% problema 7_modalitatea 2:
```

```
L = input('Introduceti pragul L = ');
```

```
n = input('Introduceti numarul de termeni n = ');
```

```
k = 1:n;
```

```
s2 = cumsum(k.^2)
```

```
% gasim termenii care verifica: suma patratelor este < L dat de noi
```

```
% si luam maximul acestora
```

```
rang_max = max(find(s2<L))
```

```
fprintf('Suma de patrute de valoare maxima \n care este mai mica decat L = %.2f\n este obtinuta pentru \n  
rangul %.0f\n',L,rang_max)
```