clc

a=[3 8 7 4 5 8]

a =

3 8 7 4 5 8

%mostrar solo los impares

a(mod(a,2)~=0)

ans =

3 7 5

%posiciones de los impares

clc

mod(a,2)~=0

ans =

1 0 1 0 1 0

i=1:6

i =

1 2 3 4 5 6

i.\*(mod(a,2)~=0)

ans =

1 0 3 0 5 0

ans(ans>0)

ans =

1 3 5

x=[ 2 0 3 4 0]

x =

2 0 3 4 0

find(x)

ans =

1 3 4

find(x>3)

ans =

4

i=find(mod(a,2)~=0)

i =

1 3 5

clc

a

a =

3 8 7 4 5 8

%sumar 1 a todos los numeros impares

i=find(mod(a,2)~=0);

a(i)=a(i)+1

a =

4 8 8 4 6 8

a=[3 8 7 4 5 8]

a =

3 8 7 4 5 8

a+(mod(a,2)~=0)

ans =

4 8 8 4 6 8

a=[3 8 7 4 5 8]

a =

3 8 7 4 5 8

min(a)

ans =

3

max(a)

ans =

8

a

a =

3 8 7 4 5 8

[m i]=max(a)

m =

8

i =

2

[m i]=min(a)

m =

3

i =

1

%sumar 1 al elemento maximo

i=find(a==max(a))

i =

2 6

a(i)=a(i)+1

a =

3 9 7 4 5 9

%se tienen los precios y los stocks de

%unos productos

p=[4 5 2 9 7];

s=[5 3 0 4 1];

%cual es el producto k, mas caro

[m k]=max(p)

m =

9

k =

4

k=find(p==max(p))

k =

4

%cual es sk stock de producto mas caro

k=find(p==max(p));

sk=s(k)

sk =

4

%capital en almacen

sum(p.\*s)

ans =

78

p\*s'

ans =

78

%cuantos productos cuestan mas de 3

sum(p>3)

ans =

4

%El stock total de los que cuestan >3

s.\*(p>3)

ans =

5 3 0 4 1

sum(s.\*(p>3))

ans =

13

sum(s(p>3))

ans =

13

i=find(p>3);

sum(s(i))

ans =

13

a

a =

3 9 7 4 5 9

fliplr(a)

ans =

9 5 4 7 9 3

x=1:0.1:3;

%crear un vector de 6 elementos de 2 a 4

i=(4-2)/5;

x=2:i:4

x =

Columns 1 through 6

2.0000 2.4000 2.8000 3.2000 3.6000 4.0000

x=linspace(2,4,6)

x =

Columns 1 through 6

2.0000 2.4000 2.8000 3.2000 3.6000 4.0000

x=linspace(2,4);

clc

ones(1,4)

ans =

1 1 1 1

zeros(1,4)

ans =

0 0 0 0

clc

a=1:10;

d=factorial(a);

n=a.^2;

e=2\*ones(1,10);

n=a.^e;

n(2:2:end)=n(2:2:end);

n(2:2:end)=-n(2:2:end);

S=sum(n./d)

S =

-2.7557e-006

clc

diary off