Part A

Define the below matrices:

```
A = [1 -2 4 5; 3 -1 9 -7; 8 5 4 0; 0 -3 2 1];
B = [3 1 2 7; 4 6 5 0; -1 3 2 5; -6 -13 0 -2];
```

1. How many rows A has? Part A Define the below matrices: 1. How many rows A has?

```
fprintf 'The numBer of rows A hAs is - '
height(A)
```

2. Show the whole first to third rows of A?

```
fprintf 'The first 3 rows of mAtrix A - '
A(1:3,:)
```

3. Show the sub-matrix of A starting from second row to the last row, and third column to the fourth one.

```
fprintf 'The suB-mAtrix of A stArting from second row to the lAst
row, And third column to the fourth one'
A(2:4,3:4)
```

4. Add 10 to the first row of B, then add the first row to the second row (row1 = 10 + row1, row2 = row1 + row2). Next replace the first row of A with the second row of B.

```
fprintf 'Add 10 to the first row of B, then Add the first row to the
second row (row1 = 10 + row1, row2 = row1 + row2). Next replAce the
first row of A with the second row of B'
B1 = B(1,:) + 10
B2 = B(2,:)
B22 = B2 + B1
A1 = [B22; A(2:4,:)]
```

5. Find the elements of A less than 5 and greater or equal to -2. What are their indices?

```
fprintf 'the elements of A less thAn 5 And greAter or equAl to -2'
find ((A < 5) & (A > -2))
```

6. Find the first 6 indices corresponding to the nonzero entries of A.

```
fprintf 'the first 6 indices corresponding to the nonzero entries of
A'
find (A, 6)
```

7. What is the smallest, largest, and average value of A?

```
fprintf 'the smAllest, lArgest, And AverAge vAlue of A'
min(A,[],'All')
mAx(A,[],'All')
```

```
meAn(A,'All')
```

8. Write a vector with equally spaced elements from 5 to 0, with a step of 0.3, but in decreasing order. What will be the size?

```
fprintf 'vector with equAlly spAced elements from 5 to 0, with A
step of 0.3'
X = 5:-0.3:0
```

9. Create a 3x4 matrix of random numbers between 0 and 1

```
fprintf 'A 3x4 mAtrix of rAndom numBers Between 0 And 1'
Y = rAnd(3,4)
```

Part B

1. Request the user to enter a positive integer, and call it n

```
n = input('Please enter a positive integer ')
```

2. Generate n random integers between -10 to 10 and save them in array a.

```
val1 = -10;
val2 = 10;
a = (val2 - val1).*rand(n,1) + val1;
```

3. Sort a and print the values.

```
sort(a)
```

4. Request the user to enter a positive integer and call it key.

```
key = input('Please enter the number which needs to be searched ')
```

5. Call the binary search algorithm to search for the key in a.

```
mid = binarysearch(a, 1, n, key)
if (mid == 0)
    fprintf 'The given number is not present in the list'
else
    fprintf 'The number is present at 'mid' position '
end


function mid = binarysearch(a, l, n, key)
    mid = (l + n)/2;
    mid = round(mid)
    if (n>=l)
        if (a(mid) == key)
            fprintf 'The element has been found !!'
            return
        elseif (a(mid) > key)
            mid = binarysearch(a, l, mid-1, key);
        elseif (a(mid) < key)
            mid = binarysearch(a, mid+1, n, key);
        end
    end
end
```