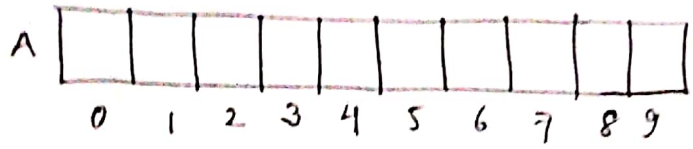
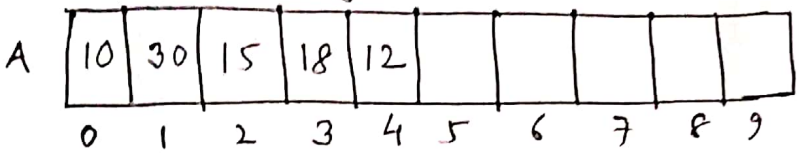
Recursive Linear Search

```
#include <stdio.h>
int Linear( int A[], int i, int n, int key);
int main(){
    int A[10], key, n; int flag;
    // read n from keyboard
    // read key from keyboard
    // read data A from keyboard
    flag = Linear(A, 0, n, key);
    if (flag == 0)
        printf("Not Found");
    else printf("Found at index %d", flag);
    return 0;
}
int Linear(int A[], int i, int n, int key){
        if (i >= n)
            return 0;
        if (A[i] == key)
            return i;
        else
            return Linear(A, i+1, n, key);
}
```



n = 5 (keyboard)

After reading data from keyboard



Key = 15

flag = Linear(A, 0, 5, 15) ⟹ main() ⇒ else

Monitor: Found at index 2

2

if (i >= n) → 0 5 False

if (A[0] == 15) → 10 False

return Linear(A, 1, 5, 15) → 2

if (i >= n) → 1 5 False

if (A[1] == 15) → 30 False

return Linear(A, 2, 5, 15) → 2

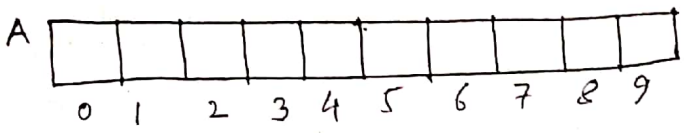
if (i >= n) → 2 5 False

if (A[2] == 15) → 15 True

return 2

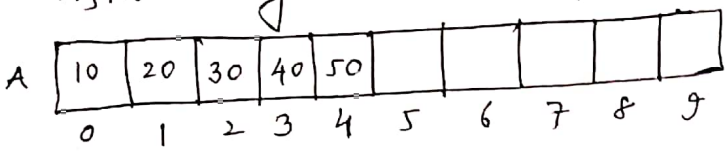
Recursive Binary Search:

```
#include<stdio.h>
int Binary(int A[], int low, int high, int key);
int main(){
    int A[10], key, n, flag;
   // read n from keyboard
   // read A data from keyboard
   // read key from keyboard
   flag= Binary(A, 0, n-1, key);
   if(flag==0)
       printf("Not Found");
   else printf("Found at index %d", flag);
   return 0;
}
int Binary(int A[], int low, int high, int key){
        if(low>high)
            return 0;
        mid = (low+high)/2;
        if(A[mid]==key)
            return mid;
        else if(A[mid]>key)
            return Binary(A, low, mid-1, key);
        else
            return Binary(A, mid+1, high, key);
    }
```



n = 5 (from keyboard)

After reading data from keyboard



key = 40

flag = Binary(A, 0, 4, 40) ⇒ main() ⇒ else ⇒

Monitor: Found at index 3

low high key

3

if (low > high) → 0 > 4 → False

mid = ⌊(0+4)/2⌋ = 2

if (A[2] == 40) → A[2] = 30 → False

else if (A[2] > 40) → 30 > 40 → False

else return Binary (1, 3, 4, 40) (low high key)

3

if (low > high) → 3 > 4 → False

mid = (3+4)/2 = 3

if (A[3] == 40) → A[3] = 40 → True

return 3