1.
```
# include <stdio.h>
# include <math.h>
  int main() {
        int t;
        scanf("%d",&t);
        while (t!=0){
            int n, Key ,status =0;
            scanf("%d",&n);
            int a[n];
            int i,count =0;
            for(i =0; i<n; i++){
                ~~a[i] =se~~
                scanf("%d",& a[i]);
           }
             scanf("%d",&Key);
             for(i=0; i<n; i++){
                count ++;
                if (Key == a[i]){
                     ~~pre~~status =1;
                         break;
                }
          }
          if (status ==1){
               printf(" Present "+count);
          }
          else{
              printf(" Not Present "+count);
          }
          t--
     }
      return 0;
  }
```

```
2.  #include<stdio.h>
    void binary (int n, int arr[], int key){
         int i, mid, low, high, count=0;
         low=0; high=n-1;
         for (i=0; i<n; i++){
              mid = (low +high)/2;
              if ((++count) && arr[mid]==key){
                   printf(" Present %d", count);
                   return;
              }
              else if ((++count) && arr[mid]>key){
                   high = mid-1;
              }
              else if ((++count) && arr[mid]>key){
                   low = mid + 1;
              }
         }
    }
    int main(){
           int t;
           scanf("%d", &t);
           while(t--){
               int n,i;
               scanf("%d", &n);
               int arr[n], key;
               for (i=0, i<n; i++){
                    scanf("%d", &arr[i]);
               }
               scanf("%d", &key);
               binary(n, arr, key);
    }
    [illegible]
```

```
}.      void linearSearch (int lp, int n, int arr[], int key, int jc);
}       int main (){
             int blockSize = 2, key;
             int status = 0;
             int n; int jc = 0;
             scanf("%d", &n);
             int arr[n];
             for (i=0; i<n ; i++){
                   scanf("%d", &arr[i]);
             }
             int lp = 0, hp = n-1;
             scanf("%d", & key);

             while (hp < n){
                  if (arr[hp] <= key){
                        if (arr[lp] == key){
                              status = 1;
                              printf("Present %d", jc);
                              return 0;
                        }
                        elseif (arr[hp] == key){
                              status = 1;
                              printf("Present %d", jc);
                              return 0;
                        }
```

```
            else {
                lp=np;
                np = np + blocksize;
                jc++;
            }
        }
        else {
                             n
            linearSearch(lp, ~~count~~, arr, key, jc, status);
            return 0;
        }
    }
                         n
    linear_Search(lp, ~~count~~, arr, key, jc, status);
}

                                    int n
void linearSearch(int lp, int arr[], int key, int jc, in
                                              status){
            int i;
            for(i=0;i<n; i++){
                ~~count~~ if(arr[i] == key){
                        jc++;
                        status = 1;
                        break;
                }
                jc++;
            }
            if(status == 1) {
                printf("Present %d", jc);
            }
            else {
                printf("Not present %d", jc);
            }
        }
```