

Αναζήτηση – Ταξινόμηση

Σειριακή Αναζήτηση

Πίνακας: *int A[10]*

A. Εύρεση του πρώτου στοιχείου.	B. Εύρεση όλων των στοιχείων.
<pre>printf("Give key:"); scanf("%i", &key); p = -1; i = 0; while ((p == -1) && (i < 10)) { if (A[i] == key) { p = i; } i++; } if (p == -1) { printf("Not found\n"); } else { printf("Found in pos:%d\n", p); }</pre>	<pre>printf("Give key:"); scanf("%i", &key); p = -1; i = 0; for (i=0; i<10; i++) { if (A[i] == key) { p = i; printf("Found in pos:%d\n", p); } } if (p == -1) { printf("Not found\n"); }</pre>

Δυναμική Αναζήτηση

Ταξινομημένος Πίνακας: *int A[10]*

```
printf("Give key:");  
scanf("%i", &key);  
lh = 0;  
rh = 9;  
p = -1;  
while ((lh <= rh) && (p == -1))  
{  
    mid = (lh + rh) / 2;  
    if (A[mid] == key) { p = mid; }  
    else if (A[mid] > key) { rh = mid - 1; }  
    else { lh = mid + 1; }  
}  
  
if (p == -1)  
{  
    printf("Not found\n");  
}  
else  
{  
    printf("Found in pos:%d\n", p);  
}
```

Ταξινόμηση BubbleSort

Πίνακας: *int A[10]*

A. Αύξουσα	B. Φθίνουσα
<pre>for (i=1; i<10; i++) { for (j=9; j>=i; j--) { if (A[j-1] > A[j]) { temp = A[j-1]; A[j-1] = A[j]; A[j] = temp; } } }</pre>	<pre>for (i=1; i<10; i++) { for (j=9; j>=i; j--) { if (A[j-1] < A[j]) { temp = A[j-1]; A[j-1] = A[j]; A[j] = temp; } } }</pre>

Δυναμική Αναζήτηση (αναδρομή)

```
int binSearch( int A[], int key, int lb, int rb)
{
    int mid;

    if (lb > rb)
        return -1;

    mid = (lb + rb) / 2;

    if (key == A[mid]) { return (mid); }
    else if (key < A[mid]) { binSearch(A, key, lb, mid - 1); }
    else { binSearch(A, key, mid + 1, rb); }
}
```

Ταξινόμηση QuickSort

```
void quickSort( int A[], int lb, int rb )
{
    int pivot, i, j, t;

    if( lb < rb )
    {
        pivot = A[lb];
        i = lb;
        j = rb+1;
        while ( i <= j )
        {
            do i++; while( A[i] <= pivot && i <= j );
            do j--; while( A[j] > pivot && i <= j );
            if( i < j ) { t = A[i]; A[i] = A[j]; A[j] = t; }
        }
        if( lb < j ) { t = A[lb]; A[lb] = A[j]; A[j] = t; }
        quickSort( A, lb, j-1 );
        quickSort( A, j+1, rb );
    }
}
```