

## SEARCHING-LINEAR SEARCH

Searching in data structure refers to the process of finding a desired element in set of items. The desired element is called "target". The set of items to be searched in, can be any data-structure like – list, array, linked-list, tree or graph.

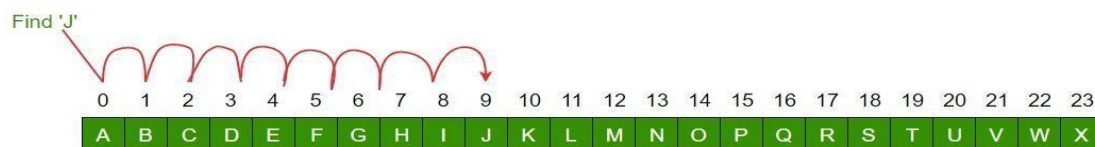
### LINEAR SEARCH

Linear search is made over all items one by one. Every item is checked and if a match is found then that particular item is returned, otherwise the search continues till the end of the data collection.

A simple approach is to do linear search, i.e

- Start from the leftmost element of arr[] and one by one compare x with each element of arr[] If x matches with an element, return the index.
- If x doesn't match with any of elements, return -1.

### Example:



### Algorithm

Linear Search ( Array A, Value x)

Step 1: Set i to 1

Step 2: if  $i > n$  then go to step 7

Step 3: if  $A[i] = x$  then go to step 6

Step 4: Set i to  $i + 1$

Step 5: Go to Step 2

Step 6: Print Element x Found at index i and go to step 8

Step 7: Print element not found

Step 8: Exit

**Program:**

```
#include <stdio.h>

int main()
{
int array[100], search, c, n;
printf("Enter the number of elements in array\n");
scanf("%d", &n);
printf("Enter %d integer(s)\n", n);
for (c = 0; c < n; c++)
scanf("%d", &array[c]);
printf("Enter a number to search\n");
for (c = 0; c < n; c++)
{
if (array[c] == search) /* If required element is found */
{
printf("%d is present at location %d.\n", search, c+1);
break;
}
}
if (c == n )
printf("%d isn't present in the array.\n", search);
return 0;
}
```

