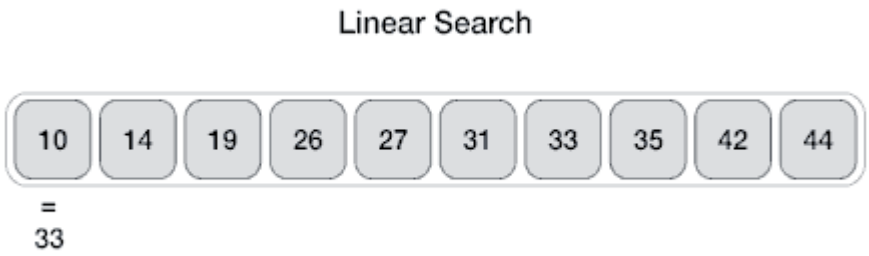
Data Structure and Algorithms Linear Search

Linear search is a very simple search algorithm. In this type of search, a sequential 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.



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
```

Pseudocode

```
procedure linear_search (list, value)

   for each item in the list

      if match item == value
```

```
         return the item's location

      end if

   end for

end procedure
```

Linear Search Program in C

Here we present the implementation of linear search in C programming language. The output of the program is given after the code.

Linear Search Program

```
#include <stdio.h>

#define MAX 20

// array of items on which linear search will be conducted.
int intArray[MAX] =
{1,2,3,4,6,7,9,11,12,14,15,16,17,19,33,34,43,45,55,66};

void printline(int count) {
   int i;

   for(i = 0;i <count-1;i++) {
      printf("=");
   }

   printf("=\n");
}

// this method makes a linear search.
int find(int data) {

   int comparisons = 0;
   int index = -1;
   int i;

   // navigate through all items
   for(i = 0;i<MAX;i++) {
```

```
      // count the comparisons made
      comparisons++;

      // if data found, break the loop
      if(data == intArray[i]) {
         index = i;
         break;
      }
   }

   printf("Total comparisons made: %d", comparisons);
   return index;
}

void display() {
   int i;
   printf("[");

   // navigate through all items
   for(i = 0;i<MAX;i++) {
      printf("%d ",intArray[i]);
   }

   printf("]\n");
}

main() {
   printf("Input Array: ");
   display();
   printline(50);

   //find location of 1
   int location = find(55);

   // if element was found
   if(location != -1)
      printf("\nElement found at location: %d" ,(location+1));
   else
      printf("Element not found.");
}
```

If we compile and run the above program, it will produce the following result −

Output

```
Input Array: [1 2 3 4 6 7 9 11 12 14 15 16 17 19 33 34
43 45 55 66 ]
==================================================
Total comparisons made: 19
Element found at location: 19
```

```
#include <stdio.h>
#define MAX 20
// array of items on which linear search will be conducted.
int intArray[MAX] = {1,2,3,4,6,7,9,11,12,14,15,16,17,19,33,34,43,45,55,66};
void printline(int count) {
  int i;
  for(i = 0;i <count-1;i++) {
    printf("=");
  }
  printf("=\n");
}
// this method makes a linear search.
int find(int data) {
  int comparisons = 0;
  int index = -1;
  int i;
  // navigate through all items
  for(i = 0;i<MAX;i++) {
    // count the comparisons made
    comparisons++;
    // if data found, break the loop
    if(data == intArray[i]) {
      index = i;
      break;
    }
  }
  printf("Total comparisons made: %d", comparisons);
  return index;
}
```

```
void display() {
   int i;
   printf("[");
  // navigate through all items
   for(i = 0;i<MAX;i++) {
      printf("%d ",intArray[i]);
   }
   printf("]\n");
}
main() {
   printf("Input Array: ");
   display();
   printline(50);
   //find location of 1
   int location = find(55);
   // if element was found
   if(location != -1)
      printf("\nElement found at location: %d" ,(location+1));
   else
      printf("Element not found.");
}
```