Name ≠ Farhan Shah

ID ≠ 13180

Subject ≠ Data Structure and Algorithm

Teacher name ≠ Dr. Naeem Khan

Date ≠ 21/8/2020

Department ≠ B(EE)

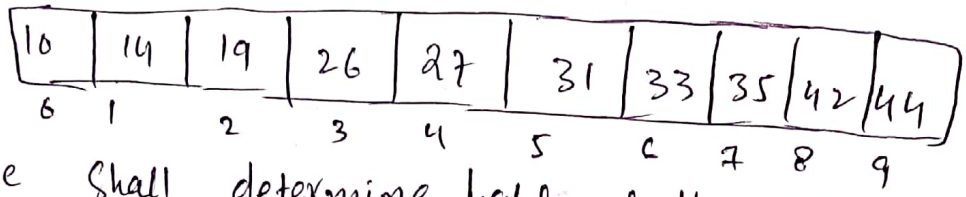
Q1: The following is your stored array and let assume that you need to search the location of value 31 using binary search.

Solution:-

This is the programe of Binary Search algorithm.

```
#Include<iostream.h>
# Include<conio.h>
Void Main()
{
 clear();
int a[9] = {10, 14, 19, 26, 27, 31, 33, 35, 42, 44},
   first =0 last, Mid , num;
num = 31;
do
 { Mid = (first + last )/2;
  if (a[Mid]==num)
  {
   cout<<"found at location"<< Mid+1;
     break;
else if (a[Mid]<num)
    first =Mid+1;
  else
        last=Mid-1;
}
while( first<= last;
```

Answer # 01 This is the answer of (Q1)

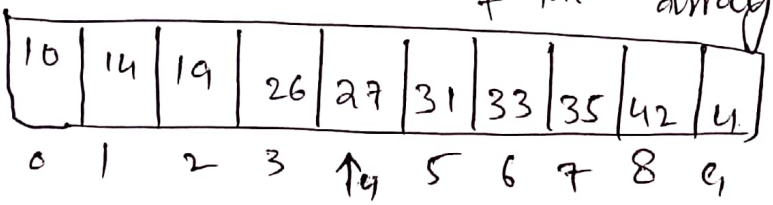


First we shall determine half of the array by using this formula.

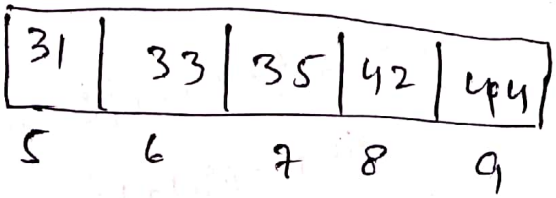
mid = low + (high - low)/2

Here it is 0 + (9 - 2)/2 = 4 (Integer value of 4.5)

So 4 is the mid of the array



Now we compare the value stored at location 4 with the value being searched i.e 31 we find the value at location 4 is 27 which is not a match. As the value is greater than 27 and we have a sorted array so we also known that the target value must be in the upper portion of the array.

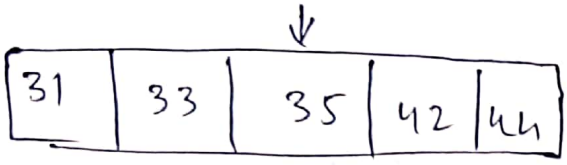


we changed our low to mid+1 and find the new mid value again

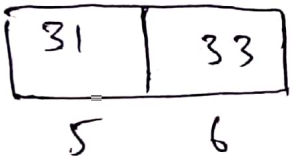
low = mid + 1

mid = low + (high - low)/2

our new mid 7 now. we compare the value stored at location 7 with our target value 31.

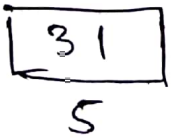


The value stored at location 7 is not a match rather it is more than what we are looking for. So the value must be in the lower part from this location.



Hence we calculate the mid again. This time it is 5

we compare value stored location 5 target value. we find that it is a match.



we ~~concul~~ conclude that the target value is 31 is stored at location 5.

Question #02

Answer #02

```
#Include <stdio.h>
main() {
Int LA[] = {1, 3, 5, 7, 8};
Int Item = 10, K = 3, n = 5;
Int i = 0, j = n;
printf ("The original array element are: \n");
for (i=0; i<n; i++) {
    printf(" LA[%d] = %d \n", i, LA[i]);
}

n = n+1;
while (j >= k) {
    LA[j+1] = LA[j];
    j = j - 1;
}
LA[K] = item
printf ("The array element after Insertion
: \n");
for (i = 0; i<n; i++) {
printf ("LA[%d] = %d \n", i, LA[i]);
}
}
```

Out put:-

Implemention of the above algorithm:-

The original array element are:

LA [0] = 1
LA [1] = 3
LA [2] = 5
LA [3] = 7
LA [4] = 8

The array element after insertion

LA [0] = 1
LA [1] = 3
LA [2] = 5
LA [3] = 70
LA [4] = 7
LA [5] = 8

Question #03

find a given target number (x = 61) using linear search from a list of number using C++.

[18, 36, 56, 61, 73, 87, 93]

Solution:-

```
# Include<iostream>
  using namespace std;
Void linear Search (int a[ ], int n)
{
```

```
int  tem = -1;

for (int i= 0; i<7 ; i++)
 {
   if (a[i] ==n)
   {
  Count<<"Element found at location"<<i<<endl;
    }
    }
    }

     int main( )
    int arr[7]= {18, 36, 56, 61, 73, 87, 93}
   cout<<"please enter an elements to search"<<
     endl;
    int num;
    cin >> num;
   linear search (arr,num);

        return 0;

      }
```