**Binary Search**

**Algorithm for Binary Search:**

* Step 1: [INITIALIZE] SET BEG = lower bound  
  END = upper bound, POS = - 1
* Step 2: Repeat Steps 3 and 4 while BEG <=END
* Step 3: SET MID = (BEG + END)/2
* Step 4: IF A[MID] = VAL  
  SET POS = MID  
  PRINT POS  
  Go to Step 6  
  ELSE IF A[MID] > VAL  
  SET END = MID - 1  
  ELSE  
  SET BEG = MID + 1  
  [END OF IF]  
  [END OF LOOP]
* Step 5: IF POS = -1  
  PRINT "VALUE IS NOT PRESENT IN THE ARRAY"  
  [END OF IF]
* Step 6: EXIT

**Program for Binary Search using array:**

#include<stdio.h>

#include<conio.h>

int main()

{

int i, low, up, mid, n, key, array[30];

printf("Enter number of elements n");

scanf("%d",&n);

printf("Enter %d integers n", n);

for(i = 0; i < n; i++)

scanf("%d",&array[i]);

printf("Enter value to find n");

scanf("%d", &key);

low = 0;

up = n - 1;

mid = (low+up)/2;

while (low <= up) {

if(array[mid] < key)

low = mid + 1;

else if (array[mid] == key) {

printf("%d found at location %d.n", key, mid+1);

break;

}

else

up = mid - 1;

mid = (low + up)/2;

}

if(low > up)

printf("Not found! %d isn't present in the list.n", key);

return 0;

}

**Program for Binary Search Using linked list:**

#include<stdio.h>

#include<stdlib.h>

struct Node

{

int data;

struct Node\* next;

};

Node \*newNode(int x)

{

struct Node\* temp = new Node;

temp->data = x;

temp->next = NULL;

return temp;

}

struct Node\* middle(Node\* start, Node\* last)

{

if (start == NULL)

return NULL;

struct Node\* slow = start;

struct Node\* fast = start -> next;

while (fast != last)

{

fast = fast -> next;

if (fast != last)

{

slow = slow -> next;

fast = fast -> next;

}

}

return slow;

}

struct Node\* binarySearch(Node \*head, int value)

{

struct Node\* start = head;

struct Node\* last = NULL;

do

{

Node\* mid = middle(start, last);

if (mid == NULL)

return NULL

if (mid -> data == value)

return mid;

else if (mid -> data < value)

start = mid -> next;

else

last = mid;

} while (last == NULL ||

last != start);

return NULL;

}

int main()

{

Node \*head = newNode(1);

head->next = newNode(4);

head->next->next = newNode(7);

head->next->next->next = newNode(8);

head->next->next->next->next = newNode(9);

head->next->next->next->next->next = newNode(10);

int value = 7;

if (binarySearch(head, value) == NULL)

printf("Value not present\n");

else

printf("Present");

return 0;

}