

# ASSIGNMENT-6

N. Vaishnavi  
API9110010487

```
#include <stdio.h>
void main()
{
    int array[10], sumpla, Propla;
    int a, b, c, d, num, temp, keynum;
    int small, cen, rise;
    printf("Enter value of sort 10");
    scanf("%d", &num);
    printf("enter the elements\n");
    for (a=0; a<num, a++)
    {
        scanf("%d", &array[a]);
    }
    printf("Array elements 10");
    for (a=0; a<num; b++)
    {
        for (b=0; b<(num-a-1); b++)
        {
            if (array[b] < array [b+1])
            {
                temp = array[b];
                array[b] = array[b+1];
                array[b+1] = temp;
            }
        }
    }
    printf("The sorted array\n");
    for (a=0; a<num, a++)
    {
        printf("%d\n", array[a]);
    }
}
```

```

printf("enter the element that need to Search in");
scanf("%d", &keynum);
small = 1;
rise = num;
do
{
    cen = (small + rise)/2;
    if (keynum > array[cen])
        rise = cen - 1;
    else if (keynum < array[cen])
        small = cen + 1;
}
while (keynum != array[cen] && small <= rise);
if (keynum == array[cen])
{
    printf("Search success and %d found location %d", keynum, mid);
}
else
{
    printf("Search failed in");
}
printf("enter the location in sorted array in");
scanf("%d %d", &c, &d);
c--;
d--;
for(a=0, a<num, a++)
{
    sumloc = array[z] + array[k];
    proloc = array[z] * array[k];
}
printf("In sum of the locations is %d", sumla);
printf("In product of the location %d", propla);
}

```

Output :-

Enter the value of sort

3

Enter the elements

2

4

5

The sorted array

5

4

2

Enter the element that need to be search

4

search success 4 is found at location 2

Enter the location in sorted array

4

sum of the locations 6

product of the locations 8

### 3) Insertion Sort :-

Insertion sort in C is a simple and efficient algorithm, that creates the final sorted array one element at a time.

Insertion sort works in a similar manner as we arrange a deck of cards.

Avg & worst-case complexity of this algorithm is  $O(n^2)$ .

Insertion sort is not good for large data sets.

Eg:- Initial Array

130	92	120	140	110
-----	----	-----	-----	-----

130	130	120	140	110
-----	-----	-----	-----	-----

92	130	120	140	110
----	-----	-----	-----	-----

92	130	130	140	110
----	-----	-----	-----	-----

92	120	130	140	110
----	-----	-----	-----	-----

92	120	130	140	110
----	-----	-----	-----	-----

92	120	120	130	140
----	-----	-----	-----	-----

Sorted Array  $\rightarrow$  [92 110 120 130 140]

Selection Sort :-

In Selection sort, the smallest element is exchanged with the first element of the unsorted list of elements. Then the second smallest element is exchanged with the second element of the unsorted list of elements and so on until all the elements are sorted.

Avg f worst-case complexity of the algorithm is  $O(n^2)$

Eg :-

6 12 9 4 5  
↑ scan 6, smallest 4  
exchange

4 12 9 6 5  
↑ scan 12, small 5  
exchange

4 5 9 6 12  
↑  
exchange

4 5 6 9 12

↑  
smallest - exchange

4 5 6 9 12

5)

```
#include <stdio.h>
#include <stdlib.h>
int Binary Search (int arr[], int num, int first, int last)
{
    if (first > last)
        printf("number you have entered is not found ");
    }
else
{
    int mid;
    mid = (first + last)/2 ;
    if (arr[mid] == num)
    {
        printf("element you have asked for is found at index %d", mid);
        exit(0);
    }
    else if (arr[mid] > num)
    {
        Binary Search (arr, num, first, mid - 1);
    }
    else
    {
        Binary Search (arr, num, mid + 1, last);
    }
}
int main()
{
    int arr[] = {110, 140, 160, 180, 120};
    int num = 140;
    int first = 0, last = (size of (arr) / size of (arr[0])) - 1;
    Binary Search (arr, num, first, last);
}
```

Output:

element you have asked for is found at index 1.

```
#include <stdio.h>
int main()
{
    int array[100], n, a, b, i, m, swap, sum = 0, proo = 1;
    printf("enter the elements in ");
    scanf("%d", &n);
    printf("enter %d integers\n", n);
    for(a=0; a<n; a++)
        scanf("%d", &array[a]);
    for(a=0; a<n-1; a++)
    {
        for(b=0; b<n-a-1; b++)
        {
            if(array[b] > array[b+1])
            {
                swap = array[b];
                array[b] = array[b+1];
                array[b+1] = swap;
            }
        }
    }
    printf("sorted array is as follows");
    for(a=0; a<n, a++)
        printf("\n%d", array[a]);
    printf("Alternative series is ");
}
```

```

for (i=0; i<n; i++)
{
    if (i % 2 == 0)
    {
        printf ("%d", array[i]);
    }
}

for (i=0, i<n; i++)
{
    if (i % 2 != 0)
    {
        sum = sum + array[i];
    }
    else
    {
        prod = prod * array[i];
    }
}

printf ("The sum in odd position is %d", sum);
printf ("The product in even %d", prod);
printf ("Enter the value");
scanf ("%d", &m);

for (i=0; i<n; i++)
{
    if (array[i] % m == 0)
    {
        printf ("%d", array[i]);
    }
}

```

Output :-

Enter the elements 5

enter 5 integers

5

4

3

2

1

Sorted array in A0

1

2

3

4

5

The Alternative Series 1 3 5

sum in odd is 9

Product in even is 8

Enter the value.

2

2 4

Q) #include <stdio.h>

Void mergesort(int array[], int i, int j);

Void merg(int array[], int i1, int j1, int i2, int j2);

Void main().

{

int array[40], n, i, k;

printf("enter the value of n");

scanf("%d", &n);

printf("enter the value in array");

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

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

mergesort (array, 0, n-1);

```

printf("Is sorted arrays is");
for (i=0; i<n; i++)
    printf("%d", array[i]);
int prodL = 1, prodR = 1;
printf("In Enter the value of k");
scanf("%d", &k);
k = k - 1;
for (i=0, i<=k, i++)
{
    prodR = prod * array[i];
}
for (i=n-1; i>=k; i--)
{
    prodL = prodL * array[i];
}
printf("in the product from start is equal %d", prodR);
printf("in the product from last is equal %d", prodL);

}

void mergesort(int array[], int i, int j)
{
    int mid;
    if (i < j)
    {
        mid = (i+j)/2;
        mergesort(array, i, mid);
        mergesort(array, mid+1, j);
        merge(array, i, mid, mid+1, j);
    }
}

void merge(int array[], int i1, int j1, int i2, int j2)
{
    int temp[50];
    int i, j, k;

```

$i = i_1$ ,  
 $j = j_2$ ;

$k = 0$ ;

while ( $i \leq j_1$  & &  $j \leq j_2$ )

{  
if ( $\text{array}[i] < \text{array}[j]$ )

$\text{temp}[k++] = \text{array}[i++]$ ;

else

$\text{temp}[k++] = \text{array}[j++]$ ;

}

while ( $i \leq j_1$ )

$\text{temp}[k++] = \text{array}[i++]$ ;

while ( $j \leq j_2$ )

$\text{temp}[k++] = \text{array}[j++]$ ;

while ( $j \leq j_2$ )

$\text{temp}[k++] = \text{array}[j++]$ ;

for ( $i = i_1$ ,  $j = 0$ ,  $i \leq j_2$ ,  $i++, j++$ )

$\text{array}[i] = \text{temp}[j]$ ;

}

Output :-

enter the value of sort 5

enter the value of array 3

4

3

2

sorted array is 1 2 3 5

enter the value of k 2

product from start is equal to 2

product from last is equal to 24.