

① #include <stdio.h>

int main()

{
int i, low, high, mid, n, key, arr[100], temp, one,
two, sum, product;

printf("enter the no. of elements in array");

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

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

{
if (j=i+1; j<n; j++)

{
if (arr[i] < arr[j])

{
temp = arr[j];

{
arr[i] = arr[j];

arr[j] = temp;

}

}

}

printf("In elements of array is sorted in descending
order");

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

{
printf("%d", arr[i]);

}

printf("enter value to find

```

search ("%.d", &key);
low = 0;
high = n - 1;
mid = (low + high) / 2;
while (low < high)
{
    if (arr[mid] > key)
    {
        low = mid + 1;
    }
    else if (arr[mid] == key)
    {
        printf ("%.d found at location %.d",
               key, mid + 1);
        break;
    }
    else
    {
        high = mid - 1;
        mid = (low + high) / 2;
    }
}

if (low > high)
{
    printf ("Not found! %.d isn't present in the list.",
           key);
}

printf ("\n");

printf ("Enter two locations to find sum & products
of the elements")

```

```

scanf ("%d", &done);
scanf ("%d", &d two);
sum = arr[one] + arr[two];
Product = (arr[one] * arr[two]);

Print f ("the sum of elements = %d", sum);
Print f ("the product of elements = %d", Product);

return 0;
}

```

Output:

Enter no. of elements in array 5

Enter 5 integers

9

7

5

4

2

Element of array is sorted in descending order.

9 7 5 4 2 Enter value to find 5

5 found at location 3

Enter two locations to find sum and product of elements.

2

4

The sum of elements = 7

The product of elements = 10

②

```
#include <stdio.h>
#include <stdlib.h>
#define max_size 5

void merge_sort [max_size];
void merge_array (int, int, int, int);

int arr_sort [max_size];

int main ()
{
    int i, k, pro=1;
    printf ("Sample merge sort example function and
            array/n");

    printf ("In enter %d elements for sorting In",
            max_size);

    for (i=0; i<max_size; i++)
    {
        scanf ("%d", &arr_sort [i]);
        printf ("In your data:");
    }
    for (i=0, i<max_size; i++)
    {
        printf ("%d", arr_sort [i]);
    }
    merge_sort (0, max_size-1);
}
```



```
Print f (" \n sorted data: ");
```

```
for (i=0, i < max_size; i++)
```

```
{
```

```
Print (" If %d," arr_sort[i]);
```

```
}
```

```
Print f ("Find the product of the kth element from  
first and last where k \n");
```

```
Scan f ("%d", &k);
```

```
Pro = arr_sort[k] * arr_sort[max_size - k - 1];
```

```
Print f ("Produce = %d", Pro);
```

```
getch();
```

```
}
```

```
void merge_sort (int i, int j)
```

```
{
```

```
int m;
```

```
if (i < j)
```

```
{
```

```
m = (i + j) / 2;
```

```
merge_sort (i, m);
```

```
merge_sort (m + 1, j);
```

```
merging two arrays
```

```
merge_array (i, m, m + 1, j);
```

```
}
```

```
}
```

```
void merge_array (int a, int b, int c, int d)
```

```

{
    int t[50];
    int i=a, j=c, k=0;
    while (i <= b & j <= d)
    {
        if (arr_sort[i] < arr_sort[j])
            t[k++] = arr_sort[i++];
        else
            t[k++] = arr_sort[j++];
    }
    // collect remaining elements
    while (i <= b)
        t[k++] = arr_sort[i++];
    for (j=c; j <= d; j++)
        arr_sort[j] = t[j];
}

```

output :-

enter 5 ~~elements~~ elements for sorting

9

7

4

6
2

your data : 9 7 4 6 2

sorted data : 2 4 6 7 9

k = 2

product = 36

```
① #include <stdio.h>
#include <stdio.h>
int main()
```

```
{ int arr[50], i, n, temp, sum=0, product=1;
```

```
Print f ("Enter total number of elements to store;")
```

```
scanf ("%d", &n);
```

```
Print f ("Enter %d elements:", n);
```

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

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

```
Print f ("\n sorting array using bubble sort technique),
```

```
for (i=0, i<(n-1); i++);
```

```
{ for (j=0; j<(n-1-i); j++)
```

```
{ if if arr[j] > arr[j+1]
```

```
{
```

```
temp = arr[j];
```

```
arr[j] = arr[j+1]
```

```
arr[j+1] = temp;
```

```
}
```

```
}
```

```
Print f ("all array elements sorted successfully; \n");
```

```

Print + ("Array elements in ascending order, \n");
for (i=0; i<n; i++)
{
    Print + ("%d \n", arr[i]);
}
Print + ("array elements in alternate order \n");
for (i=0; i<n; i=i+2)
{
    Print + ("%d \n", arr[i]);
}
for (i=1; i<n; i=i+2)
{
    sum = sum + arr[i];
}
Print + ("the sum of odd positive elements  
are = %d \n", sum);

for (i=0; i<n; i=i+2)
{
    product *= arr[i]
}
Print + ("the products of even position  
elements are = %d \n", Product);

getch();

return 0;
}

```


Output:-

Enter 5 elements

8

6

4

3

2

All array elements sorted successfully

Array elements in ascending order

2

3

4

6

8

Array elements in alternate order

the sum of odd position element is 9

the product of even position element are 6, 4.

⑤

```
#include <stdio.h>
```

```
#include <stdio.h>
```

```
void binary search (int arr[], int num, int first, int last)
```

```
{
```

```
    int mid;
```

```
    if (first > last)
```

```
    {
```

```
        printf ("number is not found");
```

```

    }
    else
    {
        mid = (first + last) / 2;
    }
    if (arr[mid] == num)
    {
        printf("Element is found at index %d",
               mid);
        exit(0);
    }
    else if (arr[mid] > num)
    {
        primary_search(arr, num, first, mid - 1);
    }
    else
    {
        Binary_search(arr, num, mid + 1, last);
    }
}

void main() {
    int arr[100], beg, mid, end, i, n, num;
    printf("Enter the size of an array");
    scanf("%d", &n);

    printf("Enter the values in sorted sequence (n);");
    for (i = 0; i < n; i++)
    {

```

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

```
}
```

```
beg = 0;
```

```
end = n-1;
```

```
printf ("Enter a value to be search: ");
```

```
scanf ("%d", &num);
```

```
Binary Search (arr, num, beg, end);
```

```
}
```

Output:

Enter the size of an array 5

Enter the values in sorted sequence

4

5

6

7

8

Enter a value to search: 5

Elements is found at index: 1