

Assignment - 6

1. Take the elements from the user and sort them in descending order and do the following.

(a) Using binary search, find the element and the location in the array where the element is asked from user.

(b) Ask the user to enter any two locations print the sum and product of values at those locations in the sorted array.

```
#include <stdio.h>
Void binarysearch (int a[], int n)
{
    int item, first, last, mid, k=0;
    printf("Enter the element you want to search:");
    scanf("%d", &item);
    first=0;
    last=n;
    while (first<=last)
    {
        mid=(first+last)/2;
        if (a[mid]==item)
        {
            printf("Found at %d", mid+1);
            k=1;
            break;
        }
        else if (a[mid]>item)
        {
            first=mid+1;
        }
    }
}
```

```

        else
    {
        last = mid - 1;
    }
}

if (k == 0)
{
    printf("Item not found");
}

Void sum_product(int a[], int n)
{
    int i, j;
    printf("\nEnter the positions:");
    scanf("%d %d", &i, &j);
    printf("\nSum of %d and %d is %d", a[i-1],
           a[j-1], a[i-1] + a[j-1]);
    printf("\nProduct of %d and %d is %d",
           a[i-1], a[j-1], a[i-1] * a[j-1]);
}

int main()
{
    int a[100], n, temp;
    printf("Enter the no. of elements:");
    scanf("%d", &n);
    printf("Enter elements:\n");
    for (int i=0; i<n; i++)
    {
        for (int j=i+1;

```

```

for (int i=0 ; i<n; i++) {
    for (int j=i+1 ; j<n ; j++)
    {
        if (a[i] < a[j])
        {
            temp = a[i];
            a[i] = a[j];
            a[j] = temp;
        }
    }
    printf("The sorted array is :\n");
    for (int i=0 ; i<n; i++)
    {
        printf("%d ", a[i]);
    }
    binarysearch(a, n);
    sum_product(a, n);
}

```

Output:

Enter the number of elements : 5

Enter elements :

52

41

21

12

32

The sorted array is :

52 41 32 21 12

You want to search : 12

Enter the element that

Found at 5

Enter the positions : 4 5

Sum of 21 and 12 is 33

Product of 21 and 12 is 252

Q. Sort the array using Merge sort where elements are taken from the user and find the product of k^{th} elements from first and last where k is taken from the user.

```
#include <stdio.h>
Void mergesort(int a[], int i, int j);
void merge(int a[], int i1, int j1, int i2, int j2);
int main()
{
    int a[100], n, i, k;
    printf("Enter no. of elements: ");
    scanf("%d", &n);
    printf("Enter elements: \n");
    for (i=0; i<n; i++)
    {
        scanf("%d", &a[i]);
    }
    mergesort(a, 0, n-1);
    printf("Sorted array is: \n");
    for (int i=0; i<n; i++)
    {
        printf("%d ", a[i]);
    }
    printf("\nEnter k value: ");
    scanf("%d", &k);
    printf("Product is: %d", a[k-1]*a[n-k]);
}
```

```
Void mergesort (int a[], int i, int j)
```

```
{ int mid;
```

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

```
{ mid = (i+j)/2;
```

```
mergesort (a, i, mid);
```

```
mergesort (a, mid+1, j);
```

```
merge (a, i, mid, mid+1, j);
```

```
}
```

```
}
```

```
Void merge (int a[], int i1, int j1, int i2, int j2)
```

```
{ int temp[50];
```

```
int i, j, k;
```

```
i = i1;
```

```
j = j2;
```

```
k = 0;
```

```
while (i <= j1 && j <= j2)
```

```
{ if (a[i] < a[j])
```

```
{ temp[k++] = a[i++];
```

```
}
```

```
else {
```

```
temp[k++] = a[j++];
```

```
}
```

```
}
```

```
while (i <= j1)
```

```
{ temp[k++] = a[i++];
```

```
}
```

```
while (j <= j2)
{
    temp[k++] = a[j++]
}

for (i = i1, j = 0; i <= j2; i++, j++)
{
    a[i] = temp[j];
}
```

3.

Output:-

Enter no. of elements: 5

Enter array elements:

16

2

9

1

15

Sorted array is :

1 2 9 15 16

Enter k value: 2

Product is 30.

3. Discuss Insertion Sort and Selection Sort with examples.

Insertion Sort:- The array get sorted item by item.

Comparsion of the elements in array takes place from the right side. The worst time complexity is $O(n^2)$.

Code:-

```
#include <stdio.h>
Void main()
{
    int n, array[100], t, c, d;
    printf ("Enter no. of elements: ");
    scanf ("%d", &n);
    printf ("Enter elements:\n");
    for (int i=0; i<n; i++)
    {
        scanf ("%d", &a[i]);
    }
    for (c=0; c<=n-1; c++)
    {
        d = c;
        while (d>0 && array[d-1] > array[d])
        {
            t = array[d];
            array[d] = array[d-1];
            array[d-1] = t;
            d--;
        }
    }
    printf ("Sorted array is :\n");
    for (int i=0; i<n; i++)
    {
        printf ("%d ", array[i]);
    }
}
```

Let us take an example

5	62	32	4	2
↑	↑			

5	62	32	4	2
↑	↑			

5	32	62	4	2
↑	↑			

c=1 : temp=1, Comparing 0th, 1st indexes, Smallest will be placed first.

c=2 : temp=2, Comparing 1st & 2nd indexes, swapping takes place. Again it will compare 0th & 1st indexes (temp=1), remains same.

c=3 : temp=3, Compares 3rd & 2nd then 2nd & 1st then 0th & 1st elements. we use while loop until temp gets zero to sort. "4" will be arranged in the first position of array.

5	32	62	4	2
↑	↑			

→

5	32	4	62	2
↑	↑			

→

5	4	32	62	2
↑	↑			

→

4	5	32	62	2
↑				

Swapping
takes place

Similarly, at c=4, we will get a sorted array after shifting 2 to the first position one by one.

4	5	32	62	2
↑	↑			

→

4	5	32	2	62
↑	↑			

→

4	5	2	32	62
↑	↑			

→

4	2	5	32	62
↑	↑			

Sorted Array :-

2	4	5	32	62
↑	↑			

Selection Sort :- The program repeatedly find the smallest value to arrange the array in increasing order. It takes less time when compare with bubble sort as few swaps takes place. Swapping takes place after finding the minimum value.

Code:-

```
#include <stdio.h>
void main()
{
    int a[100], n, c, d, temp, position;
    printf("Enter no. of elements:");
    scanf("%d", &n);
    printf("Enter elements:\n");
    for (c=0; c<n; c++)
    {
        scanf("%d", &a[c]);
    }
    for (c=0; c<n-1; c++)
    {
        position = c;
        for (d=c+1; d<n; d++)
        {
            if (a[position] > array[d])
            {
                position = d;
            }
        }
        if (position != c)
        {
            temp = a[c];
            array[c] = a[position];
            a[c] = temp;
            a[position] = temp;
        }
    }
    printf("The sorted array is:\n");
    for (c=0; c<n; c++)
    {
        printf("%d ", a[c]);
    }
}
```

```

for (c=0; c<n; c++)
{
    printf ("%d ", a[c]);
}

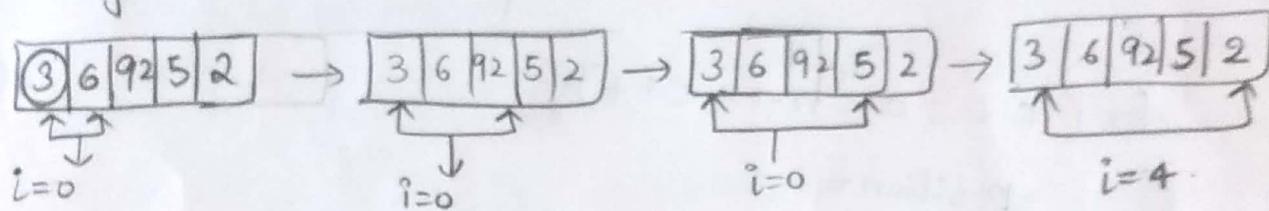
```

f.

Example for Selection Sort:-

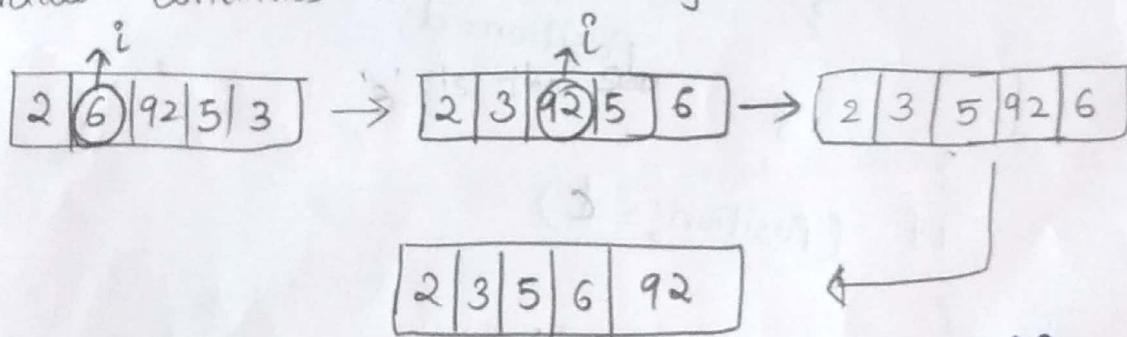
Let 'a' be an array $a = [3 | 6 | 92 | 5 | 2]$

Step 1:- It will consider the minimum value to be in the first position. It will find the minimum value among the array and replace it with $a[0]$.



Now Swapping takes place between $a[0]$ & $a[4]$.

Step 2:- The process of finding the position of minimum value continues until the array ends.



Swapping takes place only once in step 1. After comparing the elements and finding got found minimum value, swapping takes place.

The sorted array is : $[2 | 3 | 5 | 6 | 92]$

- Q(4) Sort the array using bubble sort where elements are taken from the user and display the elements
- i) in alternate order
 - ii) Sum of elements in odd positions and product of elements in even positions.
 - iii) Elements which are divisible by m where m is taken from the user.

```
#include <stdio.h>
Void bubbleSort(int a[], int n)
{
    int temp;
    for (int i=0; i<n; i++)
    {
        for (int j=0; j<n-i-1; j++)
        {
            if (a[j] > a[j+1])
            {
                temp = a[j];
                a[j] = a[j+1];
                a[j+1] = temp;
            }
        }
    }
    printf("The sorted array is given by : \n");
    for (int i=0; i<n; i++)
    {
        printf("%d", a[i]);
    }
}
int main()
{
    int a[100], n, sum=0, multiply=1, m;
```

```

printf("Enter no. of elements:");  

scanf("%d", &n);  

printf("Enter elements:\n");  

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

{  

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

}  

bubble sort(a);  

printf("In The alternate elements are:\n");  

for (int i=0; i<n; i+=2)  

{  

    printf("%d ", a[i]);  

}  

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

{  

    if (i%2==0)  

    {  

        sum+=a[i];  

    }  

    else  

    {  

        multiply*=a[i];  

    }  

}  

printf("In The sum of elements in odd  

positions is: %d", sum);  

printf("In The product of elements in  

even positions is: %d", multiply);  

printf("In Enter any number to know its  

multiples in array:");  

scanf("%d", &m);

```

```

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

```

Output:-

Enter no. of elements: 5.

Enter elements:

12 23 45 56 21

The sorted array is given by:

12 21 23 45 56

The alternate elements are:

12 23 56

The sum of elements in odd positions is : 91

The product of elements in even positions is : 945

Enter any number to know its multiples in array: 2

12 56

5) Write a recursive program to implement binary search?

```

#include <stdio.h>
Void binarysearch (int a[], int first, int last, int item)
{
    int f = first, l = last, mid, k=0;
    mid = (f+l)/2;
    if (a[mid] == item)
    {

```

```

        printf("Item found at %d", mid+1);
        k=1;
    }
    (a[mid]>item) ? (l=mid-1) : (f=mid+1);
    if (k==0)
    {
        if (first <= last)
        {
            binarysearch(a,f,l,item);
        }
        else
        {
            printf("Item not found");
        }
    }
}

int main()
{
    int a[10]={1,2,3,4,7,9}, item;
    printf("Enter the element you want to search:");
    scanf("%d", &item);
    binarysearch(a,0,6,item);
}

```

Output:-

Enter the element you want to search: 2
 for Item found at 2

Enter the element you want to search: 26

Item not found