

Assignment : 6

Searching and Sorting.

Q1. Take the elements from the user and sort them in descending and do the following:

- (a) Using Binary Search, find the element and the location in the array where the element is asked from the user.
- (b) Ask the user to enter any two locations print the sum and product of values at those locations in the sorted array.

Ans.

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

```
void sorting (int a[], int n) /* function for sorting in
                                descending order */
{
    int i, j, k;
    for (i=0; i<n, i++)
    {
        for (j=0; j<n; j++)
        {
            if (a[i] < a[j])
            {
                k = a[i];
                a[i] = a[j];
                a[j] = k;
            }
        }
    }
}
```

Date _____ / _____ / _____

3 3 3 3 3 3 3 3 3 3 3 3 3 3 3

```
int binarySearch (int a[], int p, int n)
```

{

```
int i, j, middle;
```

```
i = 0;
```

```
j = n - 1;
```

```
while (i <= j)
```

{

```
    middle = (i + j) / 2;
```

```
    if (a[middle] == p)
```

```
        return middle + 1;
```

```
    else
```

{

```
        if (p < a[middle])
```

```
            j = middle - 1;
```

```
        else
```

```
            i = middle + 1;
```

{

{

```
    if (i > j)
```

{

```
        return 0;
```

{

3 3 3 3 3 3 3 3 3 3 3 3 3 3

```
int main()
```

{

```
    int n, i, a[100], x, p, j, K;
```

Date ___ / ___ / ___

```

printf (" Enter the number of elements in array.");
scanf ("%d", &n);
printf (" Enter the elements in array.");
for (i=0; i<n; i++)
{
    scanf ("%d", &a[i]);
}
Sorting (a,n);
for (i=0; i<n; i++)
{
    printf ("%d", a[i]);
}
printf (" Enter the element to be searched.");
scanf ("%d", &p);
x = binarySearch (a,p,n);
if (x!=0)
{
    printf (" Element found at position %d", x);
}
else
{
    printf (" Element not found");
}
printf (" Enter the position of two elements.");
scanf ("%d %d", &j, &k);
j=j-1;
k=k-1;
printf (" The sum of the elements is %d",
        a[j] + a[k]);
printf (" The product of the elements is %d",
        a[j] * a[k]);
return 0;
}

```

Output:

Enter the number of elements in array: 4

Enter the elements in array

1

2

3

4

4 3 2 1

Enter the element to be searched: 5

Element not found

Enter the position of two elements: 1 2

1

2

The sum of the elements is 7

The product of the elements is 12

Q2: Sort the array using Merge sort where the elements are taken from the user and find the product of Kth elements from first and last where K is taken from the user.

Ans.

#include <stdio.h>

void mergesort (int a[], int i, int j);

{

int middle;

if (j > i)

{

middle = (i+j)/2;

mergesort (a,i, middle);

mergesort (a, middle⁺¹, j);

mergesort (a,i, middle, middle+1,j);

}

}

Date ___ / ___ / ___

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

```
    int t[50];
```

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

```
    i = i1;
```

```
    j = j2;
```

```
    k = 0;
```

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

```
{
```

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

```
{
```

```
        t[k++] = a[i++];
```

```
}
```

```
else
```

```
{
```

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

```
}
```

```
}
```

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

```
{
```

```
        t[k++] = a[i++];
```

```
}
```

```
    while (j <= j2)
```

```
{
```

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

```
}
```

```
for (i = i1, j = 0; i <= j2; i++, j++)
```

```
{
```

```
    a[i] = temp[j];
```

```
}
```

```
}
```

Page No. _____

```

int main()
{
    int a[100], n, i, k, product;
    product = 1;
    printf (" Enter the number of elements in array.");
    scanf ("%d", &n);
    printf (" Enter the elements in array.");
    for (i=0; i<n; i++)
    {
        scanf ("%d", &a[i]);
    }
    printf (" In Enter the kth value : ");
    scanf ("%d", &k);
    mergesort (a, 0, n-1);
    printf (" Sorted array is : ");
    for (i=0; i<n; i++)
    {
        printf ("%d", a[i]);
    }
    if (i==k-1 || i==n-k)
    {
        product = product * a[i];
    }
    printf (" In Product. is %d ", product);
    return 0;
}

```

Date ___ / ___ / ___

Output:

Enter the number of elements in array: 6

Enter the elements in array:

1

2

3

4

5

6

Enter the kth value: 2

Sorted array is 1 2 3 4 5 6

Product is 10

Q3. Discuss Insertion sort and Selection sort with examples.

Ans.

Selection sort: The selection sort algorithm, sorts an array by repeatedly finding the minimum of array which is unsorted and putting it at the beginning. It maintains two subarrays in the array.

- The sorted one at the left.
- The unsorted one at right.

Following example explaining selection sort:

Given array : 4 3 5 6 1 2 7 (Unsorted array)

Step I: → 1 3 5 6 4 2 7

Step II: → 1 2 5 6 4 3 7

Step III: → 1 2 3 6 4 5 7

Step IV: → 1 2 3 4 6 5 7

Step V: → 1 2 3 4 5 6 7 (Completely sorted)

code for selection sort:

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

```
int main ()
```

```
{
```

```
int i,j,n, a[100], p,t;
```

```
printf (" Enter the number of elements : ");
```

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

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

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

```
{
```

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

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

```
{
```

```
position=i;
```

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

```
{
```

```
if (a[p]>a[j])
```

```
{
```

```
p=j;
```

```
}
```

```
}
```

```
if (p!=i)
```

```
{
```

```
t=a[i];
```

```
a[i]=a[p];
```

```
a[p]=t;
```

```
}
```

```
printf (" Elements are sorted in ascending order");
```

Date ___ / ___ / ___

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

```

```
        printf ("%d/t", a[i]);
    }
}
```

Insertion Sort: The strategy behind the insertion sort is similar to the process of sorting a pack of cards. We can take the card and move its position in sequence and move the remaining cards left or right as needed.

Following example explains Insertion sort

Given array : 90 80 100 50 60 (Unsorted)

$$\begin{array}{cccccc} \rightarrow & 80 & 90 & 100 & 50 & 60 \\ \rightarrow & 80 & 90 & 100 & 50 & 60 \\ \rightarrow & 80 & 90 & 100 & 50 & 60 \\ \rightarrow & 50 & 80 & 90 & 100 & 60 \\ \rightarrow & 50 & 60 & 80 & 90 & 100 \end{array}$$

(Sorted)

Code for Insertion Sort.

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

```
int main()
```

```
{
```

```
    int i, n, a[100], p, t;
```

```
    printf (" Enter number of elements : ");
```

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

```
    printf (" Enter %d number of elements ", n);
```

Page No. _____

```

for (i=0; i<n; i++)
{
    scanf ("%d", &a[i]);
}

```

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

```
{
```

```
p = i;
```

```
while (p>0 && a[p-1] > a[p])
{

```

```
t = a[p-1];
```

```
a[p-1] = a[p];
```

```
a[p] = t;
```

```
p--;
```

```
}
```

```
}
```

printf ("Elements are sorted in ascending order.");

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

```
{
```

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

```
{
```

```
}
```

- a4. 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 position and product of elements in even positions.
- (iii) Elements which are divisible by m where m is taken from the user.

Ans. # include <stdio.h>

```
int main()
```

```
{
```

```
int i, j, t, n, a[100], sum=0, product=1, m, c=0;
```

```
printf (" Enter the number of elements in array");
```

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

```
printf (" Enter elements : ");
```

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

```
{
```

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

```
}
```

```
printf (" Enter the value of m: ");
```

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

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

```
{
```

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

```
{
```

```
if (a[j] > a[j+1])
```

```
{
```

```
    t = a[j+1];
```

```
    a[j+1] = a[j];
```

```
    a[j] = t;
```

```
}
```

```
}
```

```
}
```

```
printf (" Elements are sorted in ascending order:");
```

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

```
{
```

```
    printf ("%d", a[i]);
```

```
}
```

```
printf ("\n Elements in alternate order are:");
```

Date ___ / ___ / ___

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

{

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

{

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

{

```
if (i%2==0)
```

{

```
sum = sum + a[i];
```

{

{

```
printf ("In Sum of elements in odd positions  
is %d", sum);
```

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

{

```
if (i%2 != 0)
```

{

```
product = product * a[i];
```

{

{

```
printf ("In Product of elements in even positions  
is %d", product);
```

```
printf ("In The numbers that are divisible by %d  
are : ", m);
```

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

{

```
if (a[i] % m == 0)
```

{

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

```
++c;
```

{

{

Date ___ / ___ / ___

```

if(c==0)
{
    printf (" None");
}

```

Output:

Enter the number of elements in array: 6

Enter elements:

5

3

6

2

1

4

Enter the value of m: 2

Elements are sorted in ascending order:

1 2 3 4 5 6

Elements in alternate order are: 1 3 5

Sum of elements in odd positions is 9

Product of elements in even position is 48

The numbers that are divisible by 2 are:

Q5. Write a recursive program to implement
binary search?

Ans.

#include <stdio.h>

void sort(int a[], int n)

{

int t, i, j;

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

2

```
for (j=i; j<n; j++)
```

2

```
if (a[i] > a[j])
```

2

```
t = a[i];
```

```
a[i] = a[j];
```

```
a[j] = t;
```

3

3

3

```
printf ("Sorted array : ");
```

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

2

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

3

3

```
void binary search (int a[], int x, int y, int e)
```

2

```
int middle;
```

```
if (x>y)
```

2

```
printf ("%d is not found at any position \n", e);
```

```
return;
```

3

```
middle = (x+y)/2;
```

```
if (a[middle] == e)
```

2

```
printf ("%d is found at position %d \n", e, middle + 1);
```

3

Date ___ / ___ / ___

else if ($a[middle] > e$)

{

binary-search ($a, x, middle - 1, e$);

}

else if ($a[middle] < e$)

{

binary-search ($a, middle + 1, y, e$);

}

}

*

Output:

Enter the size of array: 6

Enter elements

4

3

2

1

5

6

Sorted array: 1 2 3 4 5 6

Enter key to search: 4

4 is found at position 4

—x—

* int main()

```
int e, i, a[100];
printf ("Enter the size of array:");
scanf ("%d", &n);
printf ("Enter elements");
for (i=0; i<n; i++)
{
    scanf ("%d", &a[i]);
}
```

Sort (a, n)

```
printf ("Enter key to search");
scanf ("%d", &e);
```

binary-search ($a, 0, n, e$);

return 0;

y