Assignment - 6

B·udayshankar
AP19110010176
CSE - G

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

a) using Binary search find the element and the location in the array where the elements 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>
int main ( )
 {
  int  i, low, high, mid, n, key , arr[100], temp, i,
    one, too, sum, product ;

   printf("enter the no.of elements inarray");
   scanf("%d", &n);
    printf("enter %d integers. ",n);
    for (i=0; i<n; i++)
     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
             decending  order : \n");
for (i=0; i<n; i++)
  {
    printf("%d", arr[i]);
  }
  printf ("enter value to find");
  scanf ("%d",  &key);
  low = 0 ;
  high = n-1;
   mid = (low +high)/2;
   while (low < high)
   {
```

```
if( arr[mid]> key)
 {
  low = mid+1;
 }
 elseif (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 not present in
            list", key);
 }

printf ("\n");
```

```
printf("enter two locations to find sum and
        product of the elements");
scanf("%d", &one);
scanf("%d", &two);
sum = (arr[one] + arr[two]);
product = (arr[one] * arr[two]);
printf(" The sum of elements = %d", sum);
printf(" The product of elements = %d", product);
return 0;

}
```

<u>out put</u> :-

```
enter no.of elements in array5
enter 5 integers
11
8
3
2
1
element of array is sorted in decending
order
```

```
11 8 3 2 1    enter value to find 3

3 found at location 2

enter two locations to find  sum and product
of elements

1

3

The sum of elements = 10

The product of elements = 16
```

2) Sort the array using merge sort where elements are taken from product of kth elements from first and last where k is taken from the user.

```
#include <stdio.h>
#include <conio.h>
# define size 5

void merge sort [size];

void merge array (int ,int, int, int) ;

int arr[size] ;

int main()
{
```

```
int i ,k, pro = 1;
printf("sample merge sort enample functions
            and array\n");
printf("\n enter %d elements forsorting\n",
                                   size );
for ( i=0 ;  i< size ;i++)
 {
  scanf("%d", &arr[i]);
  ~~printf("\n0~~
 }
 for (i=0 ; i< size ; i++)
 {
   printf ("\t%d", arr[i]);
 }
  mergesort (0, size -1);
  printf (" \n sorted data");
   for (i=0; i< size ;i++)
   {
    printf("\t %d", arr[i]);
printf( "find the product of the kth element
        from   first and last  where k \n");
 scanf("%d", &k);
 pro  =  arr(k] * arr[ size - k - 1];
 printf(" ~~procedure~~ = %d", pro);
 getch();
3
void mergesort (int i, int j)
 {
    int m
    if (i<j)
     {
        m= (i+j)/2;
        merge sort ( i, m);
        merge sort (m+1, j);
         merge array (i, m, m+1, j);
     3
 3
```

```
void merge array ( int a, int b, int c, int d)
 {
   int t [60];
    int i = a , j = c, k = 0;

   while (i<b && j<=d)
    { if (arr[i] < arr[j])
         t [k++] = arr[i++];
     else
         t [k++] = arr [j++];
   }

  while ( j<=b)
        t [k++] = arr [j++]

     for( i=a, j=0, i<=d ; i++, j++)
     arr [i] = t [j] ;
  }
```

output:

```
sample merge sort example functions and
array
enter 5 elements for sorting
11
8
3
2
1
Your data : 11 8 3 2 1
sorted data : 1 2 3 8 11
Find the product of kth elements
from first and last where k=2
product = 9
```

2) i) <u>Insertion sort</u> :-

let us take

0	1	2	3	4	5
7	3	5	4	2	6

step :- 1 → temp = 3

compare 7 & 3 → swap

3	7	5	4	2	6

step :- 2 → temp = 5

compare 7 & 5

5 less than 7 → swap

3	5	7	4	2	6

step :- 3 → temp = 4

compare 7 & 4

4 is less than 7 → swap

4 is less than 5 → swap

3	5	4	7	2	6

3	4	5	7	2	6

step :- 4 temp = 2

<u>code</u>

```
for(i=1; i<n; i++)
{ key = a[i];
    j = i-1;
  while( j>=0 && a[j] >key)
  {
      a[j+1] = a[j];
       j = j-1;
  }
     a[j+1] = key;
}
```

Compare 7 & 2
2 is less than 7
2 is less than 5
2 is less than 4
2 is less than 3

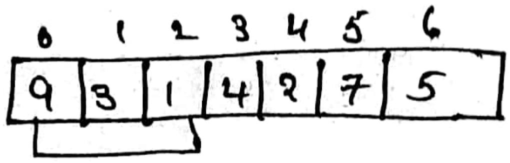
2	3	4	5	7	6

step :5 → temp = 6

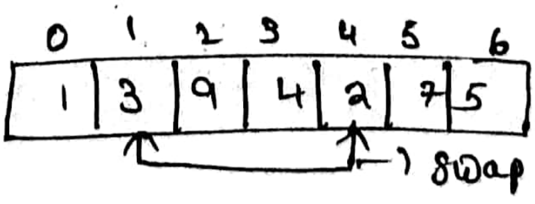
2	3	4	5	6	7

ii) <u>selection sort:</u>

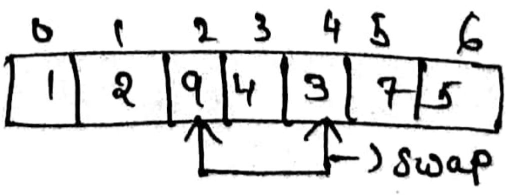
let us take



<u>step:1</u> i=0 index value compare with minimum element & swap



<u>step:2</u> i=1

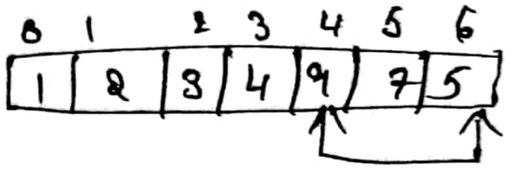


<u>step:3</u> i=2

0	1	2	3	4	5	6
1	2	3	4	9	7	5

Step : 4   it is least element swap with itself.
i = 3

Step : 5
i = 4



i = 5

0	1	2	3	4	5	6
1	2	3	4	5	7	9

Compare itself

i = 6   Compare itself

0	1	2	3	4	5	6
1	2	3	4	5	7	9

Code :

```
    int i, j, min, temp;
  for(i=0;  i<n-1; i++)
    {
          min = i;

       for ( j = i+1; j<n ; j++)
          {
               if ( a[j] <a[min])
                  {
                      min  =j
                  }
          }
            temp  = a[i]
            a[i]   =a[min]
            a[min]   = temp
    }
```

4) sort the array using bubble sort where elements are taken from the user and display the elements

i) alternate order

ii) sum of elements in odd positions and products of elements in even positions

iii) elements which are divisible by m where m is taken from the user

```
#include <stdio.h>
#include < Conio.h>
int main ()
{
 int arr [50] , i,j,n, temp, sum=0, product=1;
  Printf ("enter total number of elements to store");
  scanf ( "%d", &n);
  Printf ( "enter %d elements :",n)
   for(i=0; i<n; i++)
   scanf ( "%d", & arr[i]);
    printf ("\n sorting array using bubble sort");
     for (i=0; i< (n-1) ; i++).
      {
```

```
for(j=0;   j<(n-i-1)   ;j++)
 {
    if (arr[j]   > arr[j+1])
     {
        temp  = arr[j];
        arr[j] = arr[j+1];
        arr[j+1] = temp;
     }
}  }
printf("All array elements sorted");
printf("Array elements in ascending order:\n\n");
 for (i=0;  i<n ; i++)
  {  printf("%d",  arr[i]);
  }
  printf("array elements in alternateorder");
    for (i=0;  i<=n ; i=i+2)
    {
       printf("%d", arr[i]);
    }
    for(i=1 ; i<=n;  i=i+2)
     {
       sum = sum +arr[i];
     }
```

```
printf("The sum of odd position elements are
                    = %d \n", sum);
for (i=0; i<=n; i=i+2)
 {
   Product *= arr[i];
 }
 printf("The Products of even position
              elements are = %d \n", product);
 getch();
 return 0;
}
```

Output:-

```
enter total number of elements to
store 5
enter 5 elements
 8
 6
 4
 3
 2
sorting array using bubble sort
All array elements sorted
Array elements in ascending order
```

2
3
4
6
8

array elements in alternate order

2
4
8

The sum of odd position element is 9

The product of even position element are 6,4

6) write a recursive program to implement binary search !

```
#include < stdio.h>
~~#include < std~~
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);
  }
  elseif  (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  array");
   scanf('%.d", &n);
```

```
printf("enter the value in sorted");

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 a array 5
enter the value in sorted

 4
 5
 6
 7
 8
enter a value to search 5
element is found at index : 1
```