Assignment-6

1 Take the elements from the user and sort them in descending order to do the following
(a) Using binary search find the element and the location in the array where it is asked from user
(b) Ask the user to enter any two locations print the sum and product of values at those locations in sorted array

P. Vishal chowdary
AP19110010056
CSE-G

```
#include<stdio.h>
#define NUM 30
 void bubblesort (int array[], int size)
  {
   for(int i=0; i<size-1; ++i)
   for (int j=0, j<size-i-1, ++j)
    {
      if (array[j]< array[j+1])
      {
       int temp = array[j];
        array [j] = array[j+1];
        array [j+1]= temp;
      }
    }
 }
void display(int array[], int size)
 {
   for(int i=0; i< size; ++i) {
     printf("%d", array[i]);
  }
   printf("\n");
 }

  int binarysearch(int array[], int l, int r, int x) {
   if (r>=1) {
      int mid =l+(r-1)/2;
      if (array[mid]==x){
         return mid;
      }
     else if (array[mid]>x){
         return binarysearch(array, mid+1, r, x);
         }
     }
     return-1
```

```
void sumandproduct(int array[]) {
 int loc1,loc2;
  printf(" Enter the location 1:");
   scanf("%d", &loc1);
   printf (" Enter location 2: ");
    scanf("%d", &loc2);
     printf(" sum of elements in posishions %d and %d is:
         %d\n", loc1,loc2,array[loc1-1]+ array[loc2-1]);
}

int main( )
  {
   int a[NUM],size ,k,r,result;
    printf("Enter no.of elements of array: ");
  scanf("%d", &size);
    for(k=0; k<size ;k++)
    {
      printf(" Enter the %d th element:",k+1);
        scanf("%d", & a[k]);
     }
    printf("Given array :\n");
     display(a,size);
      bubblesort(a,size);
       printf("sorted array in Descending order:\n");
       display(a,size);
       printf("a\n");
       printf(" Enter the element to search:");
         scanf("%d", &r);
         result =binarysearch(a,0,size-1,r);
         if(result==-1) {
          printf("%d element is not found in sorted array",r);
           }
           else {
           printf("%d element is not found in Sorted array",r);
           }
             else {
printf ("%d element is found in Sorted array at locat
                   %d",result+1);
    }
    printf("b\n");
     sum and product(a);
      return 0;
```

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

```
#include <stdio.h>
#define ms 100
 int a[ms];
 void merge(int l1, int u1, int l2, int u2)
 {
   Int i, j, k, temp[ms];
    k=0;
    i=l1;
    j=l2;
  while((i<=u1) && (j<=u2)){
       if(a[i] < a[j]){
       temp[k]=a[i]; i++; k++;
     }
     else{
       temp[k]=a[j]; j++; k++;
     }
   }
   while(i<=u1){
       temp[k]=a[i]; i++; k++;
    }
    while(j<=u2){
      temp[k]=a[j]; j++; k++;
     }
    for(i=l1, k=0; j<=u2; i++, k++){
       a[i]=temp[k];
     }
  }
```

```
void mergesort(int lb, int ub){
  if (lb<ub)
 {
  int mid=(ub+lb)/2
  mergesort( lb,mid);
  mergesort(mid+1, ub),
  merge(lb, mid, mid+1, ub);
 }
}
int main(){
 int i,n, product =1, k;
 printf(" Enter the size of the array\n");
   Scanf("%d", &n);
 for(i=0; i<n ;i++){
    printf("a [%d]\t =",i);
    Scanf("%d", &a[i]);
 }
  mergesort (0, n-1);
  printf("Enter k\n");
  Scanf("%d",&k);
  for(i=0; i<k; i++){
     Product *= a[i];
  }
  Printf("The product till the kth element");
     return 0;
 }
```

3. Discuss insertion sort and selection sorry with examples :

Insertion sort:-

Insertion sort in c in a simple and efficient algorithm that creates the final sorted arry one element at a time

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

Average & worst case complexity of this algorithm in o(n2)

Insertion sort in not good for largedata sets:

Eg:-

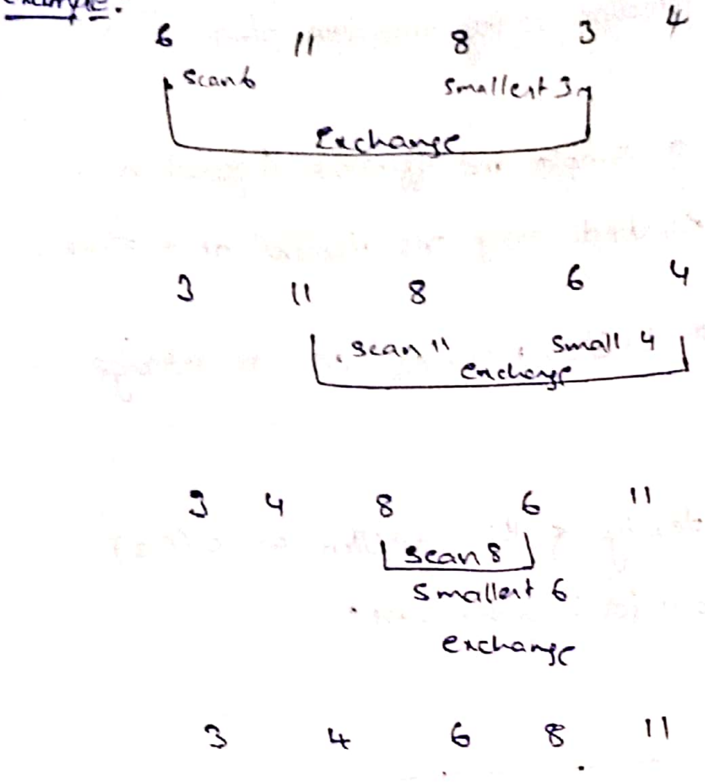
118	78	106	124	96
118	118	106	124	96
78	118	106	124	96
78	118	118	124	96
78	106	118	124	96
78	106	118	124	96
78	106	106	106	124
78	96	106	118	124

sorted → away

Selection sort:

The selection sort perform sorting by searching for the min value numbers and placing it into the first or last parsitions according to the order. The process of searching the minimeum key and placing it in the proper parsition in continued untill the all elements are placed at right parsition

Example:



4. Sort the array using bubble sort where elements taken from the user and display the elements

i. In alternative order

ii. Sum of elements in odd positions and product of element in even positions

iii. Elements which are divisible by m where m is taken from the user.

```
#include<stdio.h>
 int main( )
{
 int aray[70] ·j,c,x,i,m ,swap, f sum=0,f pro=1;
 printf("Enter the elements \n"):
 scanf("%d", &j);
 printf("Enter %d integers\n",j);
 for(c=0; c<j; c++)
  Scanf("%d", &array[c]);
  For(c=0; c<j-1; c++)
  {
   For(x=0; x<j-c-1; x++)
   { if(array[x]>array[x+1])
    {
     Swap = array[x];
array [x] = array [x+1];
 array [x+1]= swap ;
}
}
}
printf("Sorted Array in Ascending oader");
 for(c=0; c<j; c++)
 printf("%d", array[c]):
{ printf("The Alternative series in");
 for(i=0;i<j;i++)
  {
  if(i%2!=0)
  { printf("%d", array[i]);
  }
  }
  for(i=0, i<j; i++)
   { if(i%2!=0)
   { of sum= ofsum +array[i];
   }
   else
   {
   ofpro = ofpro* array [i];
    }
    }
  printf("\nsum IN ODD POSISTIONS IS %d", of sum);
  printf(" \n PRODUCT IN EVEN POSSISTION %d", ofpro);
  printf ("\n ENTER THE VALUE");
   scanf ("%d", &m);
  for (i=0 ; i<j ; i++)
  {
   if(array[i]%m==0)
   { printf("%d ", array[i]);
   }
```

OUTPUT

```
Enter the Elements 5
Enter 5 Integers
110
 90
  60
  80
  70
Sorted Array in ascending order.
  60
  70
  80
  90
  110
 The Alternative series is   70 90
 Sum in odd posistions  250
 Product in even posistion  6300
 Enter the value
  10
  60 70 80 90 ,110
```

5. Write a recursive program to implement binary search?

```
#include<stdio.h>
 void binary search(int[],int,int,int);
 void bubble.sort (int[],int);
 int main( )
{
  int key, size, i ;
  int list[25];
  printf("Enter size of a List:");
   Scanf("%d", &size);
  printf("Enter elements");
   for(i=0, i<size ; i++)
    {
     scanf ("%d", &List(i));
    {
     bubble_sort(list, size);
```

```
        printf("Enter key to search \n");
        scanf("%d ", &key);
        binary search (list, 0, size, key);
      }
      void bubble_sort(int list[], int size)
      { int temp, i, j;
        for(i=0, i<size; i++)
        {
         for(j=i; j<size; j++)
         { if (list[i] > list[j])
           {
             temp=list[i];
             list[i] = list[j];
             list[j] =temp;
           }
         }
       }
    }
}
```

output

```
Enter the key to search
 4
key to found.
```