Assignment - 6

Ch. Avinash Reddy
AP-19110010513.

1.
```
#include <stdio.h>
void swap(int *xp, int *yp)
{
 int temp = *xp;
  *xp  = *yp;
   *yp = temp;
}
void sort(int a[], int n)
{
 int i, j, min_idx;
 for (i = 0; i<n-1; i++)
 {
  min_idx = i;
  for (j = i+1; j< n; j++)
  if(a[j] > a[min_idx])
     min_idx = j;
  swap(&a[min_idx], &a[i]);
   }
}
int binary(int a[], int e, int n)
{
 int i=0, j= n-1, mid;
 while (1<=j)
{
 mid = (i+j)/2;
  if(a[mid]==e)
        return mid+1;
    else
     { if(e<a[mid])
      { j=mid-1
      }
```

```
else
 {
  i=mid+1;
 }
}
}
if(i>j)
{
return 0;
}
}
int main()
{
int n,i,a[20],f,e,a1,a2,k,j
printf("No of elements to use \n");
scanf("%d",&n);
printf("Enter the values of array: \n");
for(i=0;i<n;i++)
    scanf("%d",&a[i]);
sort(a,n);
for(i=0;i<n;i++)
    printf("%d\t",a[i]);
printf("enter the element to find in array");
scanf("%d",&e);
k=binary[a,e,n];
if(k!=0)
{ printf("element is found at %d position",f);
}
```

```
else
{
printf("element not found\n");
}
printf("enter the position of array to find the sum and product");
scanf("%d%d",&a1,&a2);
a1--;
a2--;
printf("the sum is%d", a[a1]+a[a2]);
printf("The product is%d", a[a1]*a[a2]);
}
```

2.

```
#include<stdio.h>
void mergsort(int array[], int i, int j);
void mrg(int array[], int i1, int j1, int i2, int j2);
void main()
{
int array[30], n, i, k;
printf("Enter The value of sort");
scanf("%d", &n);
printf("Enter the values in array");
for(i=0; i<n; i++)
scanf("%d", &array[i]);
mergesort(array, 0, n-1);
printf("\n sorted Array is");
for(i=0; i<n; i++)
printf("%d", array[i]);
int bof pro=1, of pro>1;
printf("\n Enter ther value of k");
scanf("%d", &k);
k=k-1;
```

```
for (i=0;i<=k;i++)
{
botpro=botpro*array[i];
}
for(i=k;i<n;i++)
{
lotpror=lotpro*array[i];
}
printf("\n The product form start is equal to %d", botpro);
printf("\n The product from last is equal to %d", lotpro);
}
voidmergesort(int array[], int i, int j)
{
int mid;
if(i<j)
{
mid=(i+j)/2;
mergesort(array, i, mid);
mergesort(array, mid+1, j);
merge(array, i, mid, mid+1, j);
}
}
void merge(int array[], 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(array[i]<array[j])
temp[k++] = array[i++];
else
temp[k++] = array[j++];
}
while(i<=j1)
temp[k++]=array[i++];
while(j<=j2)
temp[k++] = array[j++];
~~while(i<=j)~~
for(i=i1,j=0; i<=j2; i++, j++)
array[i]= temp[j];
}.
```

3. Insertion Sort

=> It is a simple and efficient algorithm that will create the sorted array one element at a time which is final

=> It works in a similar manner as we arrange a deck of cards.

=> Average and worst cases -> Case complexity of this algorithm is O(n2); it is not good for large sets.

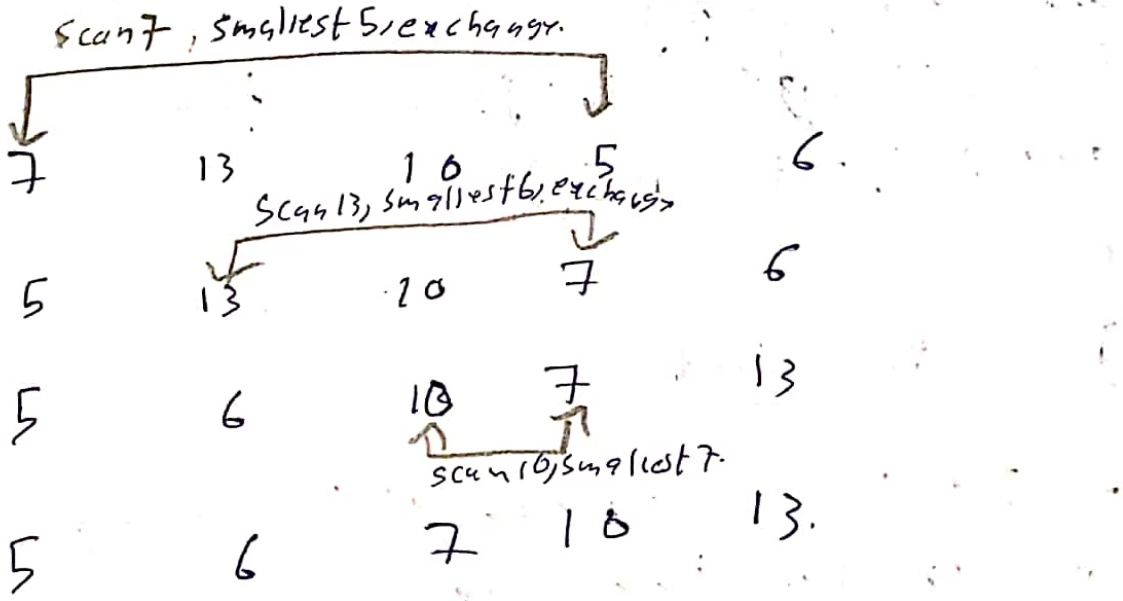
ex:- Initial Array.

122	84	112	130	102
84	122	112	130	102
84	112	122	130	102
84	116	122	130	102
84	102	112	122	130

=> sorted.

Selection Sort :- It is the smallest element is exchanged with the first element of unsorted list elements. Then the second smallest element is exchanged with the second element of the unsorted list of elements and so still they are sorted.

ex:-



4
```
#include <stdio.h>
void swap(int *xp, int *yp
{
int temp = *xp;
   *xp = *yp;
   *yp = temp;
}
void selection sort(int a[], int n)
{
 int i, j, min_idx;
 for (i = 0; i<n-1; i++)
    {
     min_idx = i;
      for (j = i+1; j<n; j++)
       if(a[j] > a[min_idx])
           min_idx = j;
      swap(&a[min_idx], &a[i]);
    }
}
void main()
{
int a[100], n, i, j, temp, sumodd=0, product=1, n;
printf("Enter no of elements\n");
scanf("%d", &n);
printf("Enter %d integers\n", n);
for (i=0; i<n; i++)
 { scanf("%d", &a[i]);
 }
```

```
selectionSort (a, n);
printf("Ascending order\n");
for (i=0; i<n; i++)
{
 printf("%d\t", a[i]);
}
printf("Alternate order:");
for (i=0; i<n; i++)
{
  if (i%2!=0)
    {
      printf("%d", a[i]);
    }
 }
for (i=0; i<n; i++)
 {if(i%2!=0)
    {
     sumodd=sumodd+a[i];
    }
    else
    { product=product*a[i];
    }
}
printf("\n Sum of odd Index is %d", sumodd);
printf("\n product of odd Index is %d", product);
printf("\n Enter the value of n\n");
scanf("%d", &n);
for (i=0; i<n; i++)
```

```
{
if(a[i] % m == 0)
 {
  printf("%d", a[i]);
 }
}
}
```

5.
```
# include <stdio.h>
# include <stlib.h>
int Binarysearch (int arr[], int num, int first, int last)
{
if (first > last)
printf ("Number entered is not found");
}.
else
{
int mid;
mid = (first + last)/2;
if (arr[mid] == num)
{
Printf ("element given by you is found in index %d", mid);
exit (0);
}
else if (arr[mid] > num)
{
Binaysearch(arr, num, first, mid-1);
}
```

```
else
{
BinarySearch(arr, num, mid+1, last);
}
}
}
int main()
{
int arr[] = {10, 82, 70, 130, 15};
int num = 130;
int first = 0, last = (sizeof(arr)/sizeof(arr[0])) - 1;
BinarySearch(arr, num, first, last);
}
```