M. Rohan Bhaskar

DSA - Assignment -6

AP19110010466

CSE-H

1.

```
# include<stdio.h>
void main()
{
      int a[40];
      int i,j,a,n;
      printf(" Enter size:\n");
       scanf("%d",&n);
      printf(" Enter elements:\n");
      for (i=0; i<n; i++)
      {
            scanf("%d", & a[i]);


      for (j= i+1; j<n; ++j)
       {
          if(a[i] < a[j])
          {
                a = a[i];
                a[i] = a[j];
                a[j] = a;
          }
      }
      }
      printf ("The descending order is :\n");
for (i=0; i<n ; i++)
{
     printf ("%d", a[i]);

}

ints, first, last, mid, l1, , l2 t, sum=0; P=1;
printf(" enter elements ");
scanf(" %d", &s);
 first = 0;
  last = n-1;
  mid = (first * last)|2;
  while ( first < = last)
 {
       if (a [mid] < search)
       {
              first = middle+1;
       }
       else if (a[mid] ==search)
       {
             printf(" %d found at %d", s, mid+1);
             break;
       }
       else
 }
 else
   last = mid-1;
   mid = (first + last)|2
}
```

```
if (first > last)
{
        printf(" Not found \n");

}
printf (" Enter two locations\n");
scanf(" %d %d", &l1, &l2);
for( i = l1,  i <= l2 ; i++)
{
        p = p * a[i] ;
}
printf(" sum = %d", sum);
printf(" product = %d", p);

}
```

2. Merge sort

```
# include < stdio.h>
# define size 100
int   a1[size]
int   a2 [size]
void merge (int first, int mid, int last)
{
      int  i,j , k, l;
      for(i= first ; j= mid+1; k = first ; i<mid &&
                                    j<= last, k++)
    {
          if (a1[i] < a2[j])
          {
                a2[k] = a1[i++]
          }
```

```
        else
        {
                a2[ k] = a1 [ j++ ];
        }
    }
    while (i<= mid)
    {
            a2[ k++] = a1[i++];
    }
    while (j <= last)
    {
            a2 [k++] = a1 [j++];
    }
    for( int l=0 ; l< last+1 ; ++l)
    {
            a1[l] = a2[l];
    }
}
void sort (int first, int last)
{
        if ( first < last)
        {
                int mid = (first + last)/2 ;
                sort (first, mid);
                sort (mid+1, last);
                merge (first, mid, last);
        }
    else
    {
        return;
    }
}

int main (void)
{
    int n, i ;
    printf(" Enter no.of elements :\n");
    scanf(" %d", &n);
    printf(" Enter elements :\n");
    for (i=0; i<n; i++)
    {
        ~~scanf(" Array after sorted:\n");~~
        scanf(" %d", &a[i]);
    }
    printf(" Array after sorted:\n");
    sort(0, n-1);
    for( i=0; i<n; i++)
    {
        printf("%d", a[i]) ;

    }
    int k, x=1;
    printf(" Enter k value\n");
    scanf(" %d", &k);
    for( i=0; i<k; i++)
    {
        x=x*i;
    }
```

```
printf(" product of k elements is %d", x);
}
```

3. Insertion sort:- The data is sorted by insertion the data into an exciting sorted file, the proces followed is elements are known before while locating to place then is searched.

Best case complexity is o(n).

Eg:-

7 4 5 2

4 7 5 2

4 5 7 2

2 4 5 7

Selection sort:- The data is sorted by inserting and placing the consecutive elements in sorted location. The best case complexity is $o(n^2)$.

Eg:-

```
17    6    3    13  6
↓          ↓
m          l
```

3 16 17 13 6

3 6 17 13 16

3 6 13 17 16

3 6 13 16 17

4.

```
# include < stdio.h>
int main()
{
    int a[100], c, n, d, swap;
    printf(" Enter size");
    scauf (" %d", &n);
    printf (" Enter elements \n");
    for (c=0; c<n; c++)
    {
        scauf(" %d", & a[c]);
    }
    for (c=0; c <n-1; c++)
    {
        for (d=0 ; d<n-c-1; d++)
        {
            if(a[d]> a[d+1])
            {
                swap = a[d];
                a [d] = a[d+1];
                a [d+1] = swap;
            }
        }
    }
    printf(" Bubble sort \n");
```

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

for (c=0; c<=n; c+=2)
{
        printf(" %d", a[c]);

}

int sum=0, p=1;

for (c=1; c<=n; c+=2)
{
        p= p * a[c];
}

for( c=0; c<=n; c+=2)
{
        s= s+a [c];
}
printf(" sum & product =%d %d", sum, p);

int m;

printf(" Enter m\n");

scanf (" %d", &m);

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

            printf(" Not found\n");
        }
    }
```

5.

```
# include <stdio.h>

 int  BS(int a[ ], intf* intl, inte)

       if (l >= f)
       {
           int m = (f+l)/2;
           if (a[m] ==e)
           {
               return m;
           }
           if (a[m]> l)
           {
              return  BS(a, f, m-1, e);

           }
           return  BS(a, m+1 , l, e);

          }
          return -1;
     }
   int  main (void)
   {    int a[ ] = { 1,4, 3, 2,9 }

        int n=5;
        int e =9;

        int p = BS(a,0, n-1,e);

        if ( p==-1)
        {
            printf(" Not found");

        }
        else
        {
            printf(" found at %d", p);

       }
}
```