DSA - Assignment - 6

T. Bhaskar

AP19110010486

1.,
```
# include <stdio.h >

void Main ( )

{

int a ( 30 ):

int i, J, a, n;
Print f ("enter size ")
scan f ("%d", & n);
 Print f (" Enter elements ");
  for (i=0 ; i <n ; i++);

    scanf (" %d ", k a(i));

for (i=0 , i<n; i++).

{
  for (J = i+1 ; J<n; ++J)

 {  if (a [ i] = a [J]]

  {  a = a[i];
   a [i] = a [J];
    a [J] = a;
   }
   }
   }
Printf (" descending order"),
 for (i=0, i<n; i++)

 {
  Print f("%d", a[i]);

 }
  Int c, first, last, mid ., l, l2 x ., sum=0, P=1
   Print f ("Enter element");
   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 ]:
     Print f ("%d found at %d", s, mid +1);
     break,

   }
   Else
    last = mid -1;
   mid = (first + last)/2
   }
    if (first > last)
```

```
{
  Printf (Not found");
}
  Printf (" Enter two locations");
  Scanf (" %d %d", &l1, &l2] ;
  for (i=l1, i<=l2 ; i++);
  {
   p= p* a[i];
  }
  Printf (" sum = %d ", sum);
  Printf (" Product = %d "P);

② # include <stdio.h>
  # include <conio.h>
  int a[20] ;n , J;
  void sort (int, int) , low ; high ,mid, b[20];
  void merge (int, int, int);
  void main ()
  {
   clrscr ();
   Printf (" Enter size")
   scanf (" %d", &n);
   Printf (" enter elements").;
   for (i=0 ; i<n ; i++).
```

```
scanf ("%d", &a[i]);
low = 0 , high = n-1;
sort (low, high)
printf ("After sorting ");
for (i=0 ; i<n; i++)
printf ("%4d", a[i]);
 product ();
 getch();
 }
 void  sort (int low ; int high);

 {
  mid : (low  + high)/2.
   if (low < high).
 {
 sort (low, mid).
 sort (mid+1, high);
 merge (low, mid, high);

 }
 void merge (int low, int mid, int high)

{
 int l1, l2.
for (l1=0, l2 = mid, i=0 , l1 <=mid && l2<= high, i++)
```

```
{
  if (a[l1]< a[l2])
    b[i] = a[l1++];
  else
    b[i] = a[l2++];
}
  while (l1<=mid)
   b[i++]  = a[l1++];
  while (l2 <=high)
   b[i++]= a[l2++];
  for [i=0 ; i<b; i++].
    a[i] = b[i]
 }
  void Poroduct ();
 {
   int p = 1 ;
  int  k = 1;
  Print f (" Enter k")
  scan f ("%d", &k);
  for (i=0, i<=k ; i++)
 {
  p= p* i;
```

3,, **Insertion soot:** The data is sorted by insertion the data into an exciting sorted life, the process followed is elements are known before while locating to place then is searched. Best case complexity is $O(n)$.

eg of insertion sort:-

7 4 5 2

4 7 5 2

4 5 7 2

2 4 5 7

eg of selection soot :-

17 6 3 13 6

↓ ↓

3 16 17 13 6

3 6 17 13 16

3 6 13 17 16

3 6 13 16 17

Selection soot:-

The data is sorted by inserting and placing the consecutive elements in sorted location the best case complexity is $O(n^2)$.

```
41.   # include <stdio.h >
        int main ( )
   {
    int a[100] ; n, c, d, swap;
    printf ("enter size );
   scanf ("%d", &n);
   printf ("enter elements"):
    for (c=0; c<n, c++)
   {
    scanf ("%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;
    }
   }
Print f  ("bubble sort")
for (i=0 ; i<n; i++)
{
  Print f ("%d", a[i]).
}
(i) Print f ("alternative elements");
  for (i=0, i<=n; i+=2).
  {
   Print f ("%d", a[i]);
  }
   int sum=0 ; P=1
2)  for(i=1; i<=n; i+=2)
    {
     P= P * a[i];
    }
  for (i=0,  i<=n, i+=2).
    {
     s = s + a[i];
    }
  Print f ("sum& Product = %d %d", sum, P);
```

```
4,   uint m;
     Printf ("Enter m");
    scanf ("%d", &m);
    for (i=0; i<=n; i++)
    {
     if (a[i] % m = = 0)
     {
     Printf ("%d",a [i]);
    }
    else
    Printf ("Not found");

5,      # include <stdio.h>
          int B.S (uint a[], uintf * uint l, uint e]
       if (l==f)
       {
       uint m= (f+l)/2
       if (a[m] ==e)
      {
       return, m;
    }
    if (a[m] >e)
   {
   return BS (a,f, m-1,e);
   }
return   BS (a, m+1, J, 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)

  {
    Print f (" notfound ")

   }
   else

  {
    Print f (" found  at   %d ", P).

         }

        }
```