Assignment-6

1)
```
# include < stdio.h >
void main( )
{
int a[30];
int i, j, a, n;
printf(" Enter size"];
scanf("%d", &n);
printf(" Enter elements");
for (i=0; i<n; ++i)
scanf("%d", &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;
}
}
}
print f(" descending order");
for( i=0; i<n; i++)
{
printf("%d", a[i]);
}
int c, first, last, mid, s, l1, l2, sum=0, P=1;
printf("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)
printf("%d" found at %d", c, mid+1);
break;
}
else
last = mid -1;
mid = (first + last)/2;
}
if(first >last)
{
printf ("Enter two locations");
scanf ("%d %d", &l1, &l2);
for(i= l1, i<= l2 ; i++)
{
sum =sum+ a[i];
P= P* a[i];
}
```

```
printf("sum= %d", sum);
printf("product = %d", p);
}
2) # include <stdio.h>
# include <conio.h>
int a[20], n, i;
void sort (int, int), low, high, mid, b[20];
void merge(int, int, int);
void product;
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 && l1 && l2<= high
                                          i+p)
{
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<n; i++)
 a[i] = b[i];
}
void product ();
{
int p = 1;
int k;
printf("Enter k");
scanf("%d",&k);
for(i=0; i<=k; i++)
{
p=p*i;
}
printf("%d",p);
}
```

3) Insertion sort - The data is sorted by inserting the data into an existing sorted file; the process follows is elements are known before while location to place them is searched.

best case complexity is O(n).

⇒ eg of selection sort:

```
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
```

⇒ eg of insertion so

```
7 4 5 2
4 7 5 2
4 5 7 2
2 4 5 7
```

Selection sort - The data is sorted by select and placing the consecutive elements. In sorted location.

The best case complexity is $O(n^2)$.

int .

```
1) # 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;
}
}
}
printf(" bubble sorted");
for(c=0; c<n; c++)
{
```

```
print f(" %d", a[c]);
}
(i) print f("alternate elements");
  for (c=0; c<=n; c+=2)
  {
  print f(" %d", a[c]);
  }
 int sum=0; p=1;
(ii) for (c=1; c<=n; c+ =2)
  {
   p=p * a[c];
  }
 for (c=0; c<=n; c+=2)
  {
  s=s+a[c];
  }
print f("sum & product = %d, %d", sum,
(iii) int m;
 print f("Enter m");
 scanf(" %d", &m);
 for (c=0; c<=n; c++)
 {
 if (a[c] % m ==0)
 {
```

```
printf("%d", a[c]);
}
else
printf("not found");
}
}
```

5)
```
# include <stdio.h>
int BS ( int a[], int f, int l int e)
  if (l >= f)
  {
   int 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, l,e);
 }
 return -1;
}
int main (void)
{
int a[]={1,4,3,2,9}
int n= b;
int e= 9;
int e = 9;
int P = BS(a, 0, n-1, e);
if (P = = -1)
{
 printf ("not found")
}
else
{
 printf ("found at %d", P);
}
}
```