C - Assignment - 6

G. Gopi chand
AP19110010189
CSE - G

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;
  }
 }
}
printf ("descending order");
```

```
for (i=0 , i<n ; i++)
{
printf ("%d", a[i]);
}
int c, first , last, mid 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", 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 (" %d", 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<=hight; 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 product ();
{ int p=1;
 Int k=1;
 printf ("enter k")
 Scanf ("%d", &k);
```

↑

```
{
  p=p*i;
}
```

3) Insertion Sort :-

The data is sorted by insertion the data into an existing sorted file, the process followed is elements are known before while locating the place then searched.

Best case complementary is $O(n)$

eg of Insertion sort

7 4 5 2

4 7 5 2

4 5 7 2

2 4 5 7

eg selection sort

17 6 3 13 6

3 16 17 13 6

3 6 17 17 16

3 6 13 13 16

3 6 13 16 17

Selection sort

The data is sorted by inserting and placing the consecutive elements in sorted location

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

4) #include <stdio.h>

```
int main ( )
{
int a [100], n, c, d, swap;
printf ("enter size") ;
Scanf ("%d", &n ) ;
printf ("enter element ") ;
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 sort")
for (c=0 ; c<n; c++)
{
printf ("%d" , a [c]) ;
}
```

```
int sum=0 ', p=1;
```

2.)
```
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);
```

3.)
```
int m;
printf ("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] >. l )
{ return Bs (a, m+1 , l, e);
}
return -1 ;
intmain (void )
{ int a[] = { 1, 4, 3, 2, 9 }
int n = 5;
int e = 9
int p = Bs (a, n-1 , e );
if (p == -1)
{ printf ("not found" )
}
else
{ printf ("found at %. d ", p);
}
```