```
void [redacted] product();
{ int p=1;
  int k;

  printf("enter the k");
  scanf("%d", &k);

  for(i=0 ; i<=k ; i++)
   {
    p=p*a[i];
   }
  printf("%d", p);

}
```

3) Insertion sort - the data is sorted by inserting the data into an existing sorted one. the process followed is elements are moved higher while location to place them is searched.

best case complexity is $O(n)$

5) [illegible] sort:

13	6	3	13	6
		↓		
3	16	13	13	16
3	6	13	13	16

[illegible] insertion sort

3	4	5	2
4	3	5	2
4	5	3	2
2	4	5	3

```
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 one. the process followed is elements are moved before write location to place item is searched.

Best case complexity is $O(n)$.

5) Big oh notation sort -

1	6	3	13	6
m		1		
3	16	13	13	16
3	6	13	13	16

Big oh notation

3 4 5 2
4 3 5 2
4 5 3 2
2 4 3 5

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


Scanned with CamScanner

```
{
  mergesort(a, mid);
  mergesort(a, mid+1, high);
  merge(a, mid, high);
}
void merge(int a[], int mid, int high)
{
  int i, j, k;
  for(i=0, k=0, j=mid+1; i<=mid && j<=high; i++)
  {
    if(a[i] < a[j])
      b[k] = a[i++];
    else
      b[k] = a[j++];
  }
  while(i<=mid)
    b[k++] = a[i++];
  while(j<=high)
    b[k++] = a[j++];
  for(i=low; i<k; i++)
    a[i] = b[i];
}
```

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()
{
  int a[] = {1, 4, 3, 2, 7};
  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);

  }
}
```


Scanned with CamScanner

```
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("\n enter m");
scanf("%d",&m);
for(c=0; c<n; c++)
  {
    if (a[c]%m==0)
      {
        printf("%d", a[c]);
      }
    else
      printf("not found");
  }
```

3 6 13 12 16

3 6 13 16 12

* Selection sort - the data is sorted by selecting and placing the consecutive elements in sorted location.
the best case complexity is $O(n^2)$.

```
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++)
     {
```

9)
```
#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 partition();
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;
```


Scanned with CamScanner

```
printf("descending order");
for(i=0; i<n; i++)
  printf("%d", a[i]);
}

int c, first, last, mid, n, search, array[100], flag=0, f=1;
printf("Enter element");
scanf("%d", &n);
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 is found at %d", search, mid+1);
    break;
}
```

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;
       }
    }
}
```


Scanned with CamScanner

```
else
  beg = mid + 1;
  mid = (beg + end)/2;
}
if (beg > end)
{ printf("not found");
}
printf("\n Enter the numbers");
scanf("%d %d", &n1, &n2);
for(i = n1; i <= n2; i++)
{
    s = s + a[i];
    p = p * a[i];
}
printf("sum = %d", s);
printf("product = %d", p);
}
```


Scanned with CamScanner