4. Write a program to sort elements in decending order.
a) Search the element using binary search.
b) Ask the user to enter two locations and print sum and product of that elements.

Program:-

```
#include<stdio.h>
int main()
{
int n,i,j,t;
printf("Enter the number of elements:");
scanf("%d",&n);
int a[n],a,b,S;first,last,mid
printf("Enter the elements");
for(i=0;i<n;i++)
scanf("%d",&a[i]);
/* sort in decending order */
for(i=0;i<n-1;i++)
{
for(j=0;j<n-i-1;<j++)
{
if(a[j+1]>a[j])
{
    t=a[j+1]
    a[j+1]=a[j]
    a[j]=t
}
```

```
}
}
printf("Sorted array in decending order is:");
for(i=0; i<n ; i++)
printf("%d", a[i]);
return 0;
}
/* Binary search */
}
printf("Enter the element to find:\n");
scanf("%d",&s);
first = 0;
last = n-1;
mid = (first + last)/2;
while (first <= last)
{
if(a[mid]<s)
first = mid+1;
else if(a[mid]==s)
printf("%d found at %d\n", s, mid);
break;
else
last = mid-1;
mid = (first + last)/2;
}
if(first > last)
printf("%d is not found in array", s);
}
/* To print sum and product of given values*/
{
printf("Enter the first element location");
scanf("%d",&a);
printf("Enter the second element location");
scanf("%d",&b);
```

```
if(a<0|| a>n-1 || b<0  || b>n-1)
printf("Wrong input");
else
printf("Sum of selected elements is =%d", a[a]+a[b]);
printf("Product of selected elements is=%d", a[a]*a[b]);
return 0;
}
```

2) Sort the array using merge sort. Find the product of kth elements from first and last where k is taken from user.

~~#include<stdio.h>~~
~~int main()~~

Program:-

```
#include<stdio.h>
void mergesort(int a[], int a , int b);
void merge(int a[], int a1 ,int b1 , int a2 , int b2);
int main()
{
int a[30],n,a;
printf("Enter number of elements");
scanf("%d",&n);
printf("Enter the elements");
for(i=0;i<n;i++)
scanf("%d",&a[i])
```

```
mergesort(a[], 0, n-1);
printf("Sorted array is\n");
for(i=0; i<n; i++)
printf("%d", a[i]);
return 0;
}
void mergesort(int a[], int i, int j)
{
int mid;
if(i<j)
{
mid=(i+j)/2;
mergesort(a, i, mid);
mergesort(a, mid+1, j);
merge(a, i, mid, mid+1, j);
void merge(int a[], int i1, int j1, int i2, int j2)
{
int temp[50];
int i, j, k;
i=i1;
j=i2;
k=0;
while(i<=j1 && j<=j2)
{
if(a[i]<a[j])
temp[k++]=a[i++]
else
temp[k++]=a[j++]
}
while(i<=j1)
temp[k++]=a[i++];
while(j<=j2)
temp[k++]=a[j++];
for(i=i1; j=0; i<=j2; i++, j++)
a[i]=temp[j];
}
```

```
{
int k;
printf("Enter the value of k");
scanf("%d",&k);
if(k<=0 || k>n)
  printf("Invalid input);
else
  printf("Product =%d", a[k-1] * a[n-k]);
return 0;
}
```

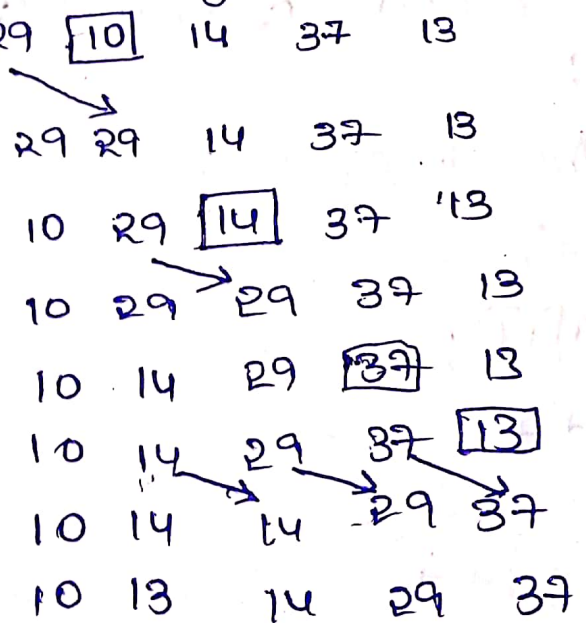
3) Discuss insertion sort and selection sort with examples

* Insertion sort: It is a simple sorting algorithm that builds the final sorted array one item at a time. It is much less efficient on large lists than more advanced algorithms such as quicksort, heapsort or mergesort.

Worst and average time complexity: $n^2$.

Best complexity: n

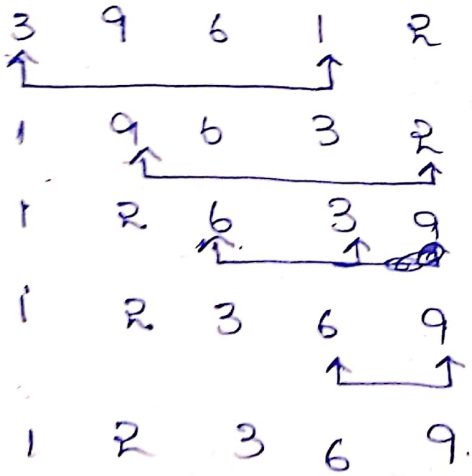
*ex:-



* Selection sort:- This sorting algorithm is an in place comparision-based algorithm in which the list is divided into two parts, the sorted part at the left and the unsorted part at right.

Time complexity : $O(n^2)$.

*ex:-



4) Sort the array using bubble sort where elements are taken from user, display them
i) in alternate order.
ii) sum of elements in odd position and product of elements in even position.
iii) Elements which are divisible by m.

program:-

```
#include<stdio.h>
void main()
{
int a[100],n,i,j,temp, s=0,p=1,m;
printf("Enter the number of elements:");
scanf("%d",&n);
printf("Enter the integers");
for(i=0;i<n;i++)
scanf("%d",&a[i]);
for(i=0;i<n-1;i++)
{
for(j=0;j<n-i-1;j++)
```

```
{
if(a[j]>a[j+1])
{
temp=a[j];
a[j]=a[j+1];
a[j+1]=temp;
}
}
}
printf("Sorted list is");
for(i=0;i<n;i++)
printf("%d",&a[i]);
printf("The alternate elements are");
for(i=0;i<n;i++)
{
if(i%2==0)
{
printf("%d",a[i]);
}
}
for(i=0;i<n;i++)
{
if(i%2!=0)
{
s=s+a[i]
}
}
printf("Sum of odd position elements is %d",s);
```

```
for(i=0; i<n; i++)
{
if(i%2==0)
{
p=p*a[i];
}
}
printf("Product of even position elements is %d", p);
printf("Enter value of m");
scanf("%d", &m);
for(i=0; i<n; i++)
{
if(a[i]%m==0)
{
printf("%d", a[i]);
}
}
}
```

5) Program of binary search function using recursion

program:-

```
#include<stdio.h>
int recBS(int a[], int si, int ei, int e){
if(ei>=si){
 int mid=si+(ei-si)/2;
 if(a[mid]==e)
  return mid;
 if(a[mid]>e)
   return recBS(a, si, mid-1, e);
   return recBS(a, mid+1, ei, e);
 }
return -1;
}
int main(void){
 int a[]={1,6,8,4,2,7}
 int n=6;
 int e=8;
int fi=recBS(a, 0, n-1, e);
if(fi==-1){
  printf("Element not found");
 }
else{
 printf("Element found at inded %d", fi);
 }
return 0;
}
```