ASSIGNMENT-6

V.Nikitha
AP191100103
CSE-F

```
1)a)#include<stdio.h>
  #include <conio.h>
   Void main ()
   {
   clrscr ();
   int A[10],n, i, L=0,U=9,f=0, M;
   printf ("\nEnter 10 elements of an array in ascending
                                            order : \n");
   for(i=0; i<10; i++)
   scanf ("%d", &A[i]);
   printf ("\n enter the elements to be searched in an arr
   scanf(" %d", &num);
    while (L<=U)
    {
      M=(L+U)/2
     if(num >A[M])
       L=M+1;

     else
      if (num < A[M])
      U=M-1;
     else
     {
       f=1
       Break;
     }
   }
    if (f==0)
                  t present in array",num);
```

```
else
printf("\n%d is ~~not~~ present in array", p);
getch();
}
#include <stdio.h>
# include <conio.h>
void main()
{
  clrscr();
int A[10],p,n,i, L=0, U=9,f=0,M,sum=0,product=1;
printf("\nEnter 10 elements of an array in
                    ascending order:\n");

for (i=0; i<10;i++)
scanf("%d", &A[i]);
printf("\n enter the elements to be searched:");
scanf("%d",&n);
while (L<=U)
 {
    M=(L+U)/2
  if (n>A[M])
   L=M+1;
  else
   if (n < A[M])
   U=M-1;
  else
  {
   f=1
   break;
  }
}
```

```
a=~~array[(great~~
while (L<=U)
{
   M=(L+U)/2
  if (p> A[M])
   L=M+1;
   else
    if (p<A[M])
    U=M-1;
   else
   {
     f=1
     break;
   }
}
```

```
        if (f==0)
         printf("\n %d is not present in the array",[illegible]

      else
       printf("\n Enter the elements to be searched", n,p
                                                   M+1);

    }

      Sum= Sum+n+p;
       product= product*n*p;

    printf(" sum and product of the searched
                    elements", sum,product);

    getch();

  }
}

2)  #include<stdio.h>
    #include<conio.h>
    Void main()

  {
    clrscr();
    int A[5], B[5], C[10];
    int i,j,k,temp;
    printf("\n Enter 5 elements of first array : \n");
   for (i=0; i<5; i++)
     scanf("%d", &A[i])
    printf("\n Enter the 5 elements of 2nd array: \n"
   for(i=0; i<5; i++)
     Scanf("%d",&B[i]);
 for(i=0; i<4; i++);
 {
```

```
for(j=i+1; j<5;j++)
{
  if (A[i]>A[j])
  {
    temp= A[i];
    A[i]= A[j];
    A[j]=temp;
  }
  if (B[i]>B[j])
   {
     temp = B[i];
     B[i]= B[j];
     B[j]= temp;
   }
  }
}
for(i=0;,j=0;,k=0;i<10,i++)
{ if(A[j]<=B[k]) //*A[0]<=B[0]*
  { c[i]= A[j];
    j++;
  }
  else
   {
     c[i]=B[k];
     k++;
   }
   if (j==5||k==5)
   { i++;
     break;
   }
```

```
for(j<5);
{
    c[i]=A[j];
    i++;
    j++;
}
for(k<5);
{
    c[i]= B[k];
    i++;
    k++;
}
printf("\n Sorted array using merge sort:\n");
for(i=0; i<10; i++)
   printf("%d\t", c[i]);
   getch();
}
```

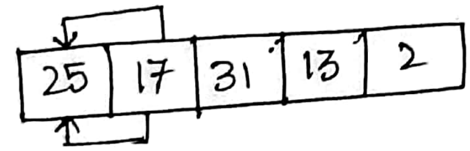
3) Insertion Sort: Insertion Sort is implemented by inserting a particular element at the appropriate position. In this method, the first iteration starts with Comparision of 1st element with 0th element. In the second iteration, 2nd element is compared with the 0th and 1st element. In general, in a every iteration an element is compared with elements. During Comparisi it is found that the element in ~~the~~ the given data can be inserted in the suitable position. This proce is repeated for all elements of the array.

) Selection Sort: This is the simplest method in the method of sorting, In this method, to sort the data in scending order, the 0th element is compared with all the other elements. If 0th element is found to be greater than the Compared element then it is interchanged. So after the overall iteration, the smallest element is placed at 0th position.

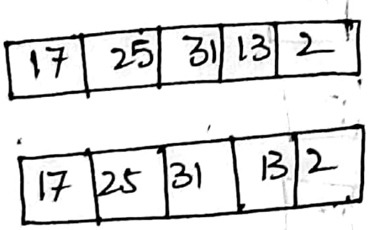
Examples:

Insertion sort:

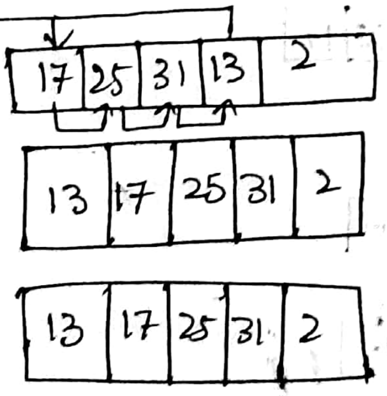
1st Iteration:



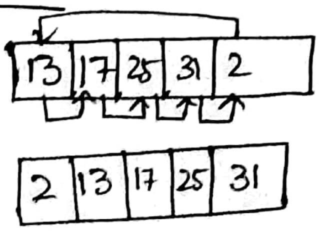
2nd Iteration:



3rd Iteration:

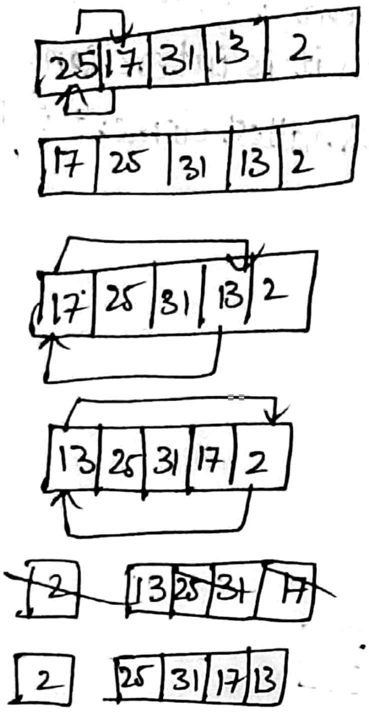


4th Iteration:



Selection Sort:

1st Iteration:



25 17 31 13 2

17 25 31 13 2

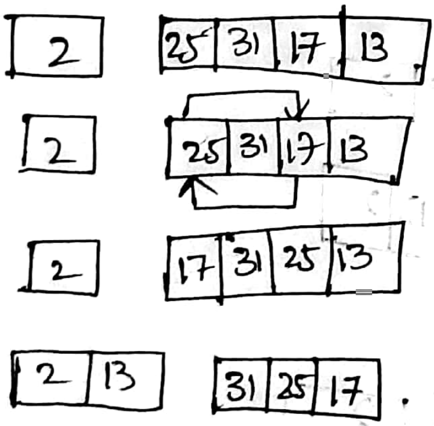
17 25 31 13 2

13 25 31 17 2

~~2 13 25 31 17~~

2 25 31 17 13.

2nd Iteration:



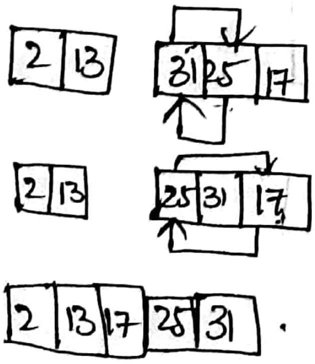
2 25 31 17 13

2 25 31 17 13

2 17 31 25 13

2 13 31 25 17.

3rd Iteration:



2 13 31 25 17

2 13 25 31 17

2 13 17 25 31.

```
5) #include <stdio.a>

 void binary_search (int [], int, int,int);
 Void bubble - sort (int[], int);

  int main ()
 {
   int Key, size,i;
   int list[25];
   printf ("Enter the size of a list : ");
   scanf ("%d", &size);
   printf ("Enter the elements \n");
   for (i=0; i<size; i++)
    {
       scanf ("%d", &list[i]);

    }
     bubble_sort (list, size);
       printf("\n");
       printf ("Enter key to search \n");
       scanf ("%d", & Key);
       binary_search (list, 0, size, key);
```

```
Void bubble_ sort (int list [], int size)
{   Int temp,i,j;
    for (i=0; i< size;i++)
    {
      for (j=0; j<size; j++)
      { if (list[i] > list[j])
         {
           temp= list[i];
           List[i]= list[j];
            List[j]=temp;
         }
       }
     }
 }
Void binary_search (int list [], int lo, int p, int ice cream)
{
   int mid;
   if (lo >p)
   {
     printf ("icecream not found\n");
     return;
   }
   mid= lo+ p/2
   if (list[mid]== key)
   {
    printf ("icecream found\n");
  }
```

```
    else if (list[mid]>Key)
    {
        binary_search( list, lo, mid-1, icecream);
    }
    else if (list[mid]<~~key~~icecream)
    {
        binary_search(list, mid+1, p, icecream);
    }
   }
  }

)  #include <stdio.h>
   #include <conio.h>

   {
   clrscr();
   static void twowaysort(int a[], int n)
    {
      int l=0,r=n-1;
        int=k=0;
          while (l<r)
            while (a[l]%2!=0){
              l++;
              k++;
          }
         while (a[r]%2==0 && l<r)
             r--;

         if (l<r){
           int temp = a[l];
                a[l]= a[r];
                  a[r]= temp;
          }
     }
```

```
array.sort(a,0,k);
array.reverse(a,k,n-k);
```

Input:

a[5] = {1, 2, 7, 9, 4}, k = 2

Output =

Sorted a[] = {1, 2, 4, 7, 9}

reverse a[] = {9, 7, 4, 2, 1}

{2, 4} is divisible by 2.