1)

P. Manoj Sai
AP19110010109 / CSE-G

```
#include<stdio.h>
void desc()
{
  int myarr[100], myvalue, loop1, loop2, temp;
  Printf(" Input myvalue:");
  Scanf("%d", &myvalue);
  for(loop1=0; loop1<myvalue; loop1++)
  {
    Printf("value=%d:", loop1++);
    Scanf("%d", &myarray[loop1]);
  }
  for (loop2=0; loop2<(myvalue-1); loop2++)
  {
    for(loop1=0; loop1<(myvalue-1); loop1++)
    { if(myarr[loop1++]<myarr[loop1]);
      {
        temp=myar[loop1];
        myarr[loop1]=myarr[loop++];
        myarr[loop1+1]=tem;
      }
    }
  }
  Printf("Descending order:\n");
  for(loop1=myvalue; loop1>0; loop1--)
  {
    Printf("%d", myarr[loop1--]);
  }
  return0;
}
```

```
void binarysearch()
{
  int f, l, mid, n, search, arr[100];
  printf("Enter no. of elements:\n");
  scanf("%d", &n);
  Printf("Enter %d elements\n", n);
  For(c=0; c<n; c++)
    { scanf("%d", &arr[i]);
    }
    Printf("Enter value to be find.\n");
    scanf("%d", &search);
    f = 0
    l = n - 1
    mid = (f + l)/2
    }
while(f<l)
{
  if(arr[mid]<search)
  {
    f = mid + 1;
  }
  else if(arr[mid] == search)
  {
    Printf("%d found at location %d\n",
                                search, mid+1);
    break;
  }
  else
```

```
{
    f = mid + 1;
    mid = (f+l)/2;
  }
  if(f > l)
  {
    Printf("Not found ! %d isn't Present in first
                                    m", search);
  }
 }
}
void mul add()
{
   int a, b, add, mul;
   Printf("Enter a location:");
   Scanf("%d", &a);
   Print f("Enter b[ocation:");
   Scanf("%d", &b);
   add = arr[a] + arr[b];
   mul = arr[a] * arr[b];
   Printf("addition = %d", add);
   Print f("Multiplication = %d", mul);
}
 void desc ()
 void binary search()
 void mul add()
    main()
```

```
{
    int choice;
    while(1)
    {
        Printf("1. Descending order \n");
        Printf("2. Searching element in array\n");
        Printf("3. add & mul\n");
        Printf("4. Quit \n");
        Printf("Enter your choice:");
        Scanf("%d", &choice);
        Switch(choice)
        {
            Case 1:
            desc()
            break;

            Case 2:
            binarysearch()
            break;
            Case 3;
            mul[add()
            break;
            Case 4;
            exit(1);
            default:
                Printf("Wrong choice\n");
        }
    }
}
```

output

1. Descending order
2. Searching Element in array.
3. add & mul
4. Quit

Enter your choice: 1

Input many value: 5

Value-1: 3
value-2: 6
Value-3: 9
Value-4: 10
Value-5: 5

Descendin order.

10 9 6 5 3

Enter your choice: 2

Enter no of elements 5

enter 5 integers.

10
9
6
5
3

Enter value to find

6

6 found at location 3

Enter your choice: 3

Enter a location: 2

Enter b location: 3

addition = 15

multiplication = 54

Enter your choice: 4

EXIT.

```
#include <stdio.h>
#define MAX 100
int arr1[MAX]
int arr2[MAX]
void merge (int low, int mid, int high)
{ int i, j, k;
  for(i=low; j=mid+1; k=low; i<=mid && j<=high;
                                        k++)
    {
      if(arr1[i]<arr1[j]
      {
        arr2[k] =arr1[i++];
      }
      else
      {
        arr2[k]= arr1[j++];
      }
    }
  while(i<=mid)
  {
    arr2[k++]= arr1[i++];
  }
  while(j<=high)
  {
    arr2[k++]= arr1[j++];
  }
  for (int l=0; l<high+1; ++l)
  {
    arr1[l] = arr2[l];
  }
}
void sort (int low, int high)
{
```

```
if( low < high)
 { int mid = (low+high) /2;
    sort( low, mid );
    sort( mid+1, high);
    merge( low, mid, high )
  }
else
   return;
 }
int main( void )
 {
    int n;
    Printf(" \n Enter no. of elements:");
    scanf("%d", &n);
    Print f(" \n Enter %d elements", n);
    for( int i=0; i<n; i++ )
      {
        scanf("%d", & arr[i]);
      }
     Printf(" \n Array after sorting is: ");
     Sort( 0, n-1);
     for( int i=0; i<n; i++ )
      { printf("%d", arr1[i]);
      }
      int k, mul=1;
      Printf(" Enter k value \n");
      scanf("%d", &k);
      for( i=0; i<k; i++)
       { mul = mul * i;
       }
      Printf(" Product of k th elements is: %d ", mul);
 }
```

· Output:-

Enter Elements: 5

8

11

10

4

Array after sorting is 4 5 8 10 11

Enter k value

3

Product of kth Elements is 160

3.iInsertion sort

Let us take

0	1	2	3	4	5
7	3	5	4	2	6

Step-1 -> temp=3

compare 7&3 ->swap

3	7	5	4	2	6

Step-2 -> tem=5

Compare 7 & 5.

5 less than 7 -> swap

3	5	7	4	2	6

Step-3 -> temp=4

Compare 7 & 4

4 is less than 7 -> swap

4 is less than 5 -> swap

3	5	4	7	2	6

3	4	5	7	2	6

Step-4 temp=2

Compare 7 & 2

2 is less than 7

2 is less than 5

2 is less than 4

2 is less than 3

2	3	4	5	7	6

Step-5 -> temp=6

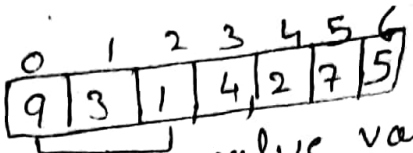
2	3	4	5	6	7

Code

```
for(i=1, i<n, i++)
{   key=a[i];
    j=i-1;
while(j>=0 && a[j]>key)
   {
     a[j+1]=a[j];
     j=j-1;
   }
a[j+1]=key;
}
```

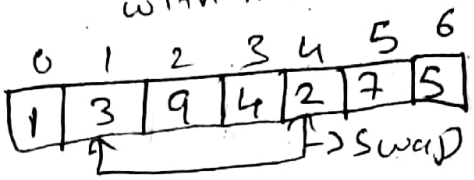
ii) Selection sort

Let us take



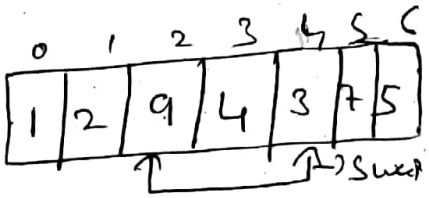
Step-1: index value value compare with min elem & swap

i=0



Step-2:

i=1

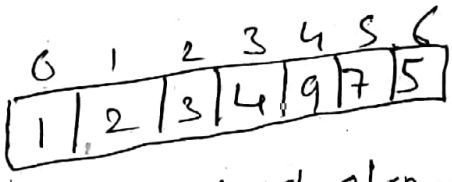


Code

```
int i, j, min, tem;
for(i=0; i<n-1; i++)
{
    min=i;
  for(j=i+1; j<n; j++)
  {
    if(a[j]<a[min])
    {
      min=j;
    }
  }
  temp= a[i]
   a[i]= a[min]
   a[min]= temp
}
```

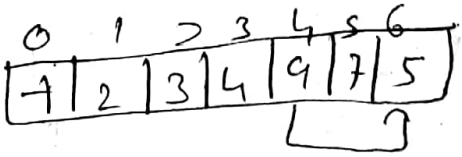
Step-3:

i=2



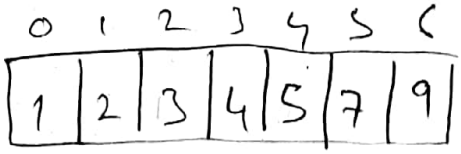
Step-4: 4 -> it is least elem and swap with itself

i=3



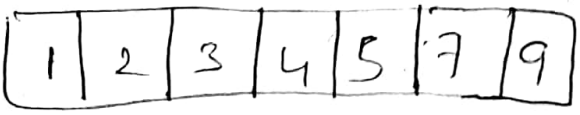
Step-5

i=4



i=5 Compare its-elf

i=6 Compares itself.



4)
```
#include<stdio.h>
int Bubblesort(int size, int *arr)
{
    int i, j, temp;
    for( i= size-2; i>=0; i--)
    {
        for (j=0; j<=i; j++)
        {
            if(arr[j] > arr[j+1])
            {
                temp= arr[j];
                arr[j]= arr[j+1];
                arr[j+1] = temp;
            }
        }
    }
    return 1;
}
int main(void)
{
    int size, i, arr[20], sum=0, mul=1, m;
    Printf("Enter no of elements :\n");
    Scanf("%d", &size);
    Printf("Enter the %d elements :", size);
    for(i=0; i<size; i++)
    {
        scanf("%d", &arr[i]);
    }
    Bubble sort (size, arr);
    Printf("After sorting");
    for(i=0; i<size; i++)
```

```
{ Printf("%d", arr[i]);
}
Printf("\n");
Printf("alternate elements after sorting \n");
for(i=0; i<size; i++)
  {
    Printf("%d", arr[i++]);
  }
Printf("\n");
Printf("Sum of elements in odd positions and mul of
                elements in even positions \n");
for(i=0; i<size; i++)
  {
    if(i%2==0)
      { mul = mul * arr[i]
      }
    else
      {
        sum = sum + arr[i]
      }
  }
Printf("Sum of Elements in odd position is %d", sum);
Printf("mul of Elements in even position is %d", mul);
Printf("Enter n value:\n");
scanf("%d", &n);
for(i=0; i<size; i++)
  {
    if(arr[i] % n == 0)
      {
        Printf("%d", arr[i]);
      }
  }Printf("\n"); return 0; }
```

Output:-

Enter no. of Elements: 5

Enter 5 Elements:

5

7

9

3

11

After sorting: 3 5 7 9 11

Alternate Elements after sorting

3 7 11

Sum of Elements in odd positions and mul of Elements in Even positions.

Sum of Elements in odd position is 21

mull of Elements in Even position is: 45

Enter m value: 5

5)
```
#include<stdio.h>
int binarysearch(int a[], int l, int h, int key)
{
    int mid=(l+h)/2;
    if(l>h)
        return -1;
    if(a[mid]==key)
        return mid;
    if(a[mid]<key)
        return binarysearch(a,mid+1,h,key);
    else
        return binarysearch(a,l,mid-1,key);
}
int main(void)
{
    int a[100];
    int n, pos,i,x;
    Printf("Enter length of array:\n");
    Scanf("%d", &n);
    Printf("Enter the elements:\n");
    for(i=0; i<n; i++)
        scanf("%d", &a[i]);
    Printf("Enter the element to search:\n");
    Scanf("%d", &x);
    Pos = binarysearch(a, 0, n-1, x);
```

```
if( pos<0)
    Printf(" cannot find %d in array \n", x);
else
    Print f(" Position of %d in array is %d \n",
                                    x, pos+1);
return 0;
}
```

Output

```
Enter the length of array: 5
Enter the array elements:
6
8
5
9
3
Eneter the element to search:
5
Position of 5 in array is 3
```