Linear search

Q.linear search in 1d

```
#include<stdio.h>

int main()

{

   int n,j,i,item,y,found=0;

   scanf("%d",&n);

   int a[n];

   for(i=0;i<n;i++)

   {


      printf("enter");

      scanf("%d",&a[i]);






   }

   printf("enter element u want to search for");

   scanf("%d",&item);

   for(i=0;i<n;i++)
```

```
{

   if(item==a[i])
   {
      found=1;
      y=i;
      break;
   }



}
if(found==1)
{
    printf("element found at index =%d",y);

}
else

{
        printf("element not found");
    }
}
```

Q.linear search in 2d

```
#include <stdio.h>

int main()

{

    int r,c,j,i,item,found=0,row,column;

    scanf("%d",&r);

    scanf("%d",&c);

    int a[r][c];

    for(i=0;i<r;i++)

    {

     for(j=0;j<c;j++)

     {

       printf("enter");

       scanf("%d",&a[i][j]);


     }


    }

    printf("enter element u want to search for");

    scanf("%d",&item);
```

```
for(i=0;i<r;i++)
{
 for(j=0;j<c;j++)
 {
   if(item==a[i][j])
   {
       found=1;
       row=i;
       column=j;
       break;
   }

 }

}
if(found==1)
{
     printf("element found at row=%d and
coulmn=%d",row,column);
```

```
    }
    else


    {
        printf("element not found");
    }
}
```

Q.Bubble sort ascending

```
#include<stdio.h>

int main(){

int x[20],i;

int n,temp;



printf("entre limit = ");

scanf("%d",&n);


printf("enter array :\n");

for(i=0;i<n;i++){

  scanf("%d",&x[i]);

}


for(i = 0 ;i<n ;i++){

   for(int j = i+1;j<n;j++){

      if(x[i]>x[j]){

         temp = x[i];

         x[i] = x[j];

         x[j] = temp;

      }
```

```
    }
}
printf("array in ascending order \n");
for(i=0;i<n;i++){
   printf("%d\n",x[i]);
}
}
```

Q.Bubble sort descending

```
#include<stdio.h>
int main(){
int x[20],i;
int n,temp;


printf("entre limit = ");
scanf("%d",&n);

printf("enter array :\n");
for(i=0;i<n;i++){
  scanf("%d",&x[i]);
}

for(i = 0 ;i<n ;i++){
    for(int j = i+1;j<n;j++){
        if(x[i]<x[j]){
            temp = x[i];
            x[i] = x[j];
            x[j] = temp;
        }
    }
}
printf("array in desending order \n");
for(i=0;i<n;i++){
   printf("%d\n",x[i]);
}
}
```

Q.program to find smallest and largest element of array

```
#include <stdio.h>
int main()
{
   int i,j,big,small,n;
   printf("enter size of array u need");
   scanf("%d",&n);
   int a[n];
   printf("enter elements of array\n");
   for(int i=0;i<n;i++)
   {
      scanf("%d",&a[i]);
   }
   big=a[0];
   small=a[0];
   for(i=1;i<n;i++)
   {
      if(a[i]>big)
      {
         big=a[i];
      }
      if(a[i]<small)
      {
         small=a[i];
```

```
        }
    }
    printf("small element fo array is=%d and largest element of array is=%d",small,big);

}
```

Q.program to find largest and second largest element of array

```
#include <stdio.h>

int main()

{

   int size;

   int i;

   int largest = -1;

   int secondLargest = -1;

   printf("How many elements you want to enter : ");

   scanf("%d",&size);

   int array[size];

   for(i=0; i < size; i++){

      printf("Enter : ");

      scanf("%d", &array[i]);

   }

   for(i=0; i<size; i++)

   {


     if(array[i] > largest)

     {

      secondLargest = largest;
```

```
        largest = array[i];
      }
      else if(array[i] > secondLargest)
      {
        secondLargest = array[i];
      }
    }

    printf("First largest number is %d\n",largest);
    printf("Second largest number is %d\n",secondLargest);
}
```

Q.sum and avg in 1d

```
#include<stdio.h>

int main()

{

   int n,i,j,sum=0;

   float avg;


   printf("enter size of array u need");

   scanf("%d",&n);

   int a[n];

   printf("enter elements of array\n");

   for(int i=0;i<n;i++)

   {

      scanf("%d",&a[i]);

   }

   for(int i=0;i<n;i++)

   {

      sum=sum+a[i];


   }

   printf("%d\n",sum);

   avg=(float)sum/(float)n;

   printf("%0.2f",avg);




}
```

Q.Updation in 1d

```
#include<stdio.h>
int main()
{
    int i,pos,ele,n;
    printf("enter size of array=");
    scanf("%d",&n);
    int a[n];
    printf("array elements are=");
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    for(i=0;i<n;i++)
    {
        printf(" %d \n",a[i]);
    }
    printf("enter position");
    scanf("%d",&pos);
     printf("enter element");
    scanf("%d",&ele);
    a[pos-1]=ele;
    printf("after updation the array\n");
    for(i=0;i<n;i++)
    {
        printf(" %d ",a[i]);
    }
}
```

Q.Updation in 2d

```
#include<stdio.h>
int main()
{
   int r,c,r1,c1,ele,j,i,item,found=0,row,column;
    scanf("%d",&r);
    scanf("%d",&c);
    int a[r][c];
    for(i=0;i<r;i++)
    {
     for(j=0;j<c;j++)
     {
       printf("enter");
       scanf("%d",&a[i][j]);


     }


    }
    printf("enter row");
    scanf("%d",&r1);
    printf("enter column");
    scanf("%d",&c1);
    printf("enter element");
    scanf("%d",&ele);
    a[r1-1][c1-1]=ele;

    printf("after updation the array\n");

    for(i=0;i<r;i++)

    {

     for(j=0;j<c;j++)

     {

       printf("%d",a[i][j]);



     }


    }
}
```

Q.To insert element in 1d array

```
#include<stdio.h>
int main()
{
    int n,i,x,poss;

    printf("enter size of array u need=");
    scanf("%d",&n);
    int a[n];



    printf("enter elements of array=\n");
    for(int i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    for(int i=0;i<n;i++)
    {
        printf("%d\t",a[i]);
    }
    printf("\nenter number you want to insert=");
    scanf("%d",&x);
    printf("\nenter position you want to insert number
at=");
    scanf("%d",&poss);
    for(i=n-1;i>=poss-1;i--)
    {
        a[i+1]=a[i];
    }
    a[poss-1]=x;
    n++;
    for(i=0;i<n;i++)
    {
        printf("%d\t",a[i]);
    }


}
```

Q.To delete element in 1d array

```
#include<stdio.h>
int main()
{
    int n,i,x,poss;

    printf("enter size of array u need=");
    scanf("%d",&n);
    int a[n];



    printf("enter elements of array=\n");
    for(int i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    for(int i=0;i<n;i++)
    {
        printf("%d\t",a[i]);
    }
    printf("\nenter position u want to delete=");
    scanf("%d",&poss);
    for(i=poss-1;i<n-1;i++)
    {
        a[i]=a[i+1];
    }
    n--;
    for(i=0;i<n;i++)
    {
        printf("%d\t",a[i]);
    }


}
```

Q.To find diagonal elements and there sum in 2d array

```
#include <stdio.h>

int main()

{

    int r,c,j,i,item,found=0,row,column,sum=0;

    scanf("%d",&r);

    scanf("%d",&c);

    int a[r][c];

    for(i=0;i<r;i++)

    {

     for(j=0;j<c;j++)

     {

       printf("enter");

       scanf("%d",&a[i][j]);


     }


    }


    for(i=0;i<r;i++)

    {

     for(j=0;j<c;j++)
```

```
 {
   printf("%d",a[i][j]);



 }


}
printf("\ndiagonal elements");
for(i=0;i<r;i++)
{
 for(j=0;j<c;j++)
 {
   if(i==j)
   {
   printf("%d",a[i][i]);
   }



 }


}
 for(i=0;i<r;i++)
```

```
    {
     for(j=0;j<c;j++)
     {
        if(i==j)
      {
      sum=sum+a[i][i];
      }




     }


    }


    printf("\nsum of elements=%d",sum);



}
```

Q.Transpose of matrix (in 2d)

```
#include <stdio.h>
int main()
{
    int r,c,j,i,item,found=0,row,column,sum=0;
    scanf("%d",&r);
    scanf("%d",&c);
    int a[r][c];
    int b[r][c];
    for(i=0;i<r;i++)
    {
     for(j=0;j<c;j++)
     {
       printf("enter");
       scanf("%d",&a[i][j]);

     }

    }

    for(i=0;i<r;i++)
    {
```

```
 for(j=0;j<c;j++)
 {
   printf("%d\t",a[i][j]);



 }
 printf("\n");
}
for(i=0;i<r;i++)
{
 for(j=0;j<c;j++)
 {
   b[i][j]=a[j][i];



 }


}
printf("transpose of matrix is\n");
for(i=0;i<r;i++)
{
 for(j=0;j<c;j++)
```

```
      {
        printf("%d\t",b[i][j]);



      }
      printf("\n");

     }


}
```

Q.Identical matrix(2d)

```
#include <stdio.h>
int main()
{
    int r,c,j,i,item,found=0,row,column,sum=0,flag=3;
    scanf("%d",&r);
    scanf("%d",&c);
    int a[r][c];
    int b[r][c];
    for(i=0;i<r;i++)
    {
     for(j=0;j<c;j++)
     {
       printf("enter");
       scanf("%d",&a[i][j]);

     }

    }

    for(i=0;i<r;i++)
    {
```

```
 for(j=0;j<c;j++)
 {
   printf("%d\t",a[i][j]);



 }
 printf("\n");
}
printf("\nenter second matrix\n");

for(i=0;i<r;i++)
{
 for(j=0;j<c;j++)
 {
   printf("enter");
   scanf("%d",&b[i][j]);

 }

}

for(i=0;i<r;i++)
```

```
{
 for(j=0;j<c;j++)
 {
   printf("%d\t",b[i][j]);



 }


 printf("\n");
}
for(i=0;i<r;i++)
{
 for(j=0;j<c;j++)
 {
   if(a[i][j]==b[i][j])
   {
       flag=1;
   }
   else
   {
       flag=0;
       break;
```

```
        }



      }



     }
     if(flag==1)
     {
        printf("identical");
     }
     else
     {
        printf("not identical");
     }

  }
```

Q.Symmetric matrix (in 2d)

```
#include <stdio.h>

int main()

{

    int r,c,j,i,item,found=0,row,column,sum=0,flag=3;

    scanf("%d",&r);

    scanf("%d",&c);

    int a[r][c];

    int b[r][c];

    for(i=0;i<r;i++)

    {

     for(j=0;j<c;j++)

     {

       printf("enter");

       scanf("%d",&a[i][j]);


     }


    }


    for(i=0;i<r;i++)

    {
```

```
 for(j=0;j<c;j++)
 {
   printf("%d\t",a[i][j]);



 }
 printf("\n");
}
  for(i=0;i<r;i++)
{
 for(j=0;j<c;j++)
 {

   b[i][j]=a[j][i];

 }

}




for(i=0;i<r;i++)
```

```
{
 for(j=0;j<c;j++)
 {
   if(a[i][j]==b[i][j])
   {
       flag=1;
   }
   else
   {
       flag=0;
       break;
   }



 }



}
if(flag==1)
{
    printf("symmetric matrix");
}
```

```
    else
    {
        printf("not symmetric");
    }

}
```

Q.not done(sorting 2d)

```
#include <stdio.h>

int main()

{

    int
r,c,i,j,j1,i1,i2,j2,t,item,found=0,row,column,sum=0,flag=3;

    scanf("%d",&r);

    scanf("%d",&c);

    int a[r][c];

    int b[r][c];

    for(i=0;i<r;i++)

    {

     for(j=0;j<c;j++)

     {

       printf("enter");

       scanf("%d",&a[i][j]);


     }


    }
```

```
for(i=0;i<r;i++)
{
 for(j=0;j<c;j++)
 {
   printf("%d\t",a[i][j]);



 }
 printf("\n");
}
for(i1=0;i1<r;i++)
{
    for(j1=0;j1<c;j++)
    {
        for(i2=0;i2<r;i++)
        {
            for(j2=0;j2<c;j++)
            {
                if(a[i1][j1]>a[i2][j2])
                {
                    t=a[i1][j1];
                    a[i1][j1]=a[i2][j2];
                    a[i2][j2]=t;
                  }
                }
              }
            }
          }
          for(i=0;i<r;i++)
          {
           for(j=0;j<c;j++)
           {
             printf("%d\t",a[i][j]);



           }
           printf("\n");
          }


        }
```

Q.WAP to find sum of even integers in an array(1D)

```
#include <stdio.h>
int main()
{
   int i,j,big,small,sum=0,n;
   printf("enter size of array u need");
   scanf("%d",&n);
   int a[n];
   printf("enter elements of array\n");
   for(int i=0;i<n;i++)
   {
      scanf("%d",&a[i]);
   }
   for(i=0;i<n;i++)
   {
   if(a[i]%2==0)
   {
      sum=sum+a[i];
   }
   }
   printf("sum of even numbers in ur array is=%d",sum);
}
```

Q.WAP to count +ve and -ve integers in an array(1D)

```
#include <stdio.h>
int main()
{
   int i,j,big,small,sum=0,n;
   printf("enter size of array u need");
   scanf("%d",&n);
   int a[n];
   printf("enter elements of array\n");
   for(int i=0;i<n;i++)
   {
      scanf("%d",&a[i]);
   }
   for(i=0;i<n;i++)
   {
   if(a[i]%2==0)
   {
      sum=sum+a[i];
   }
   }
   printf("sum of even numbers in ur array is=%d",sum);
}
```