Tutorials by Suman banerjee
(8013779048/8336808618)

C code for largest and smallest among three numbers:

```
#include<stdio.h>
int main()
{
        int a, b, c;
        printf("\n\n\t ENTER THREE NUMBERS a, b, c :\n");
        scanf("%d%d%d",&a,&b,&c);

        if((a==b)&&(b==c))
        {               printf("Numbers are equal ");
                        printf("\n");
        }

        else
                {
                printf("\n\n\t THE BIGGEST NUMBER Is:\n ");
                        if( (a > b) && (a > c) )
                                printf("a=%d",a );
                        else if(b > c)
                                printf("b=%d", b);
                        else
                                printf("c=%d", c);

                printf("\n\n\t THE SMALLEST NUMBER Is:\n ");
                        if( (a < b) && (a < c) )
                                printf("a=%d", a);
                        else if(b < c)
                                printf("b=%d", b);
                        else
```

```
                                printf("c=%d",c);
                                printf("\n");
                        }
return 0;
}
```

C code for largest and smallest among three numbers using Ternary operator:

```
# include <stdio.h>
#include<conio.h>
int main()
{
int a, b, c, big, small ;
printf("Enter three numbers : \n") ;
scanf("%d %d %d", &a, &b, &c) ;
big = a > b ? (a > c ? a : c) : (b > c ? b : c) ;
printf("\nThe largest number is : %d\n", big) ;
small = ( a < b ) ? ( ( a < c ) ? a : c ) : ( ( b < c ) ? b : c );
printf("\nThe smallest number is :%d\n",small);
return 0;
getch();
}
```

C code on BitWise operator:

```
#include <stdio.h>
main()
{
  unsigned int a = 60;     /* 60 = 0011 1100 */
  unsigned int b = 13;     /* 13 = 0000 1101 */
  int c = 0;
  c = a & b;       /* 12 = 0000 1100 */
```

```
   printf("Line 1 - Value of c is %d\n", c );

   c = a | b;       /* 61 = 0011 1101 */
   printf("Line 2 - Value of c is %d\n", c );

   c = a ^ b;       /* 49 = 0011 0001 */
   printf("Line 3 - Value of c is %d\n", c );

   c = ~a;          /*-61 = 1100 0011 */
   printf("Line 4 - Value of c is %d\n", c );

   c = a << 2;     /* 240 = 1111 0000 */
   printf("Line 5 - Value of c is %d\n", c );

   c = a >> 2;     /* 15 = 0000 1111 */
   printf("Line 6 - Value of c is %d\n", c );
}
```

C code for Sizeof operator:

```
#include <stdio.h>
main()
 { int a = 4;
   short b;
   double c;
   int* ptr;
   float f;

   /* example of sizeof operator */
   printf("Line 1 - Size of variable a = %d\n", sizeof(a) );
   printf("Line 2 - Size of variable b = %d\n", sizeof(b) );
```

```
  printf("Line 3 - Size of variable c= %d\n", sizeof(c) );
  printf("Line 4- Size of variable f= %d\n", sizeof(f) );
  /* example of & and * operators */
  ptr = &a;         /* 'ptr' now contains the address of 'a'*/
  printf("value of a is  %d\n", a);
  printf("*ptr is %d.\n", *ptr);

  /* example of ternary operator
  a = 10;
  b = (a == 1) ? 20: 30;
  printf( "Value of b is %d\n", b );

  b = (a == 10) ? 20: 30;
  printf( "Value of b is %d\n", b );
  */
}
```

C codes on Switch Case:

```
-----Vowel-Consonant----
#include<stdio.h>
int main()
{
        char ch;
        printf("ENTER THE LETTER : ");
        scanf("%c",&ch);
        switch (ch)
{
case 'A':
case 'a':
case 'E':
```

```
case 'e':
case 'I':
case 'i':
case 'O':
case 'o':
case 'U':
case 'u':
{
printf("\n %c is a vowel\n",ch);
break;
}
default :
printf("\n %c is a consonant\n",ch);
return 0;
}
}
```

-----Temperature Conversion-----

```
#include<stdio.h>
int main()
{       int choice;
        float c,f;
        printf("\n 1. Press 1 to convert Fahrenheit temperature to Centigrade\n 2. Press 2 to convert Centigrade temperature to Fahrenheit.\n Enter your choice (1/2)...\n");
        scanf("%d",&choice);
        switch(choice)
 {
                case 1:
                        {
```

```
                printf("Enter the temperature in Fahrenheit: ");
                scanf("%f",&f);
                c= (f-32)*(5.0/9.0);
                printf("The corresponding Centigrade temperature is: %f \n",c);
                break;
                }

        case 2:
        {
                printf("\nEnter the temperature in Centigrade: ");
                scanf("%f",&c);
                f=c*(9.0/5.0)+32;
                printf("\nThe corresponding Fahrenheit temperature is: %f \n",f);
                break;
        }
        default:
        {
                printf("\n Wrong choice\n");
        }
    }
    printf("\nThank you");
    return 0;
}
```

C code on number system conversion:

----Decimal to Binary----

```
#include<stdio.h>
#include<math.h>
int main()
{
```

```
 int n,c,i=0,b=0;
 printf("ENTER THE DECIMAL NUMBER : ");
 scanf("%d",&n);
 while(n!=0)
        {
          c=n%2;
          n=n/2;
          b=b+c*pow(10,i);
          i++;
        }
       printf("THE BINARY FORM= %d",b);
 return 0;
}
```

----Decimal to Octal----

```
#include<stdio.h>
#include<math.h>
int main()
{
 int n,c,i=0,b=0;
 printf("ENTER THE DECIMAL NUMBER : ");
 scanf("%d",&n);
 while(n!=0)
        {
          c=n%8;
          n=n/8;
          b=b+c*pow(10,i);
          i++;
        }
```

```
        printf("THE OCTAL FORM= %d",b);
 return 0;
}
```

----Decimal to Hexadecimal----

```
#include<stdio.h>
#include<conio.h>
#include<math.h>

void dec_hex(long int num)   // Function Definition
{
long int rem[50],i=0,length=0;

while(num>0)
  {
     rem[i]=num%16;
     num=num/16;
     i++;
     length++;
  }

printf("Hexadecimal number : ");
for(i=length-1;i>=0;i--)
 {
   switch(rem[i])
   {
     case 10:
        printf("A");
        break;
```

```
      case 11:
         printf("B");
         break;
      case 12:
         printf("C");
         break;
      case 13:
         printf("D");
         break;
      case 14:
         printf("E");
         break;
      case 15:
         printf("F");
         break;
      default :
        printf("%ld",rem[i]);
    }
  }
}

main()
{
long int num;


 printf("Enter the decimal number : ");
 scanf("%ld",&num);
```

```
   dec_hex(num);   // Calling function
}
```

C codes on numbers:

-----Armstrong Number-----

```
#include<stdio.h>
#include<math.h>
main()
{
 int c=0,a,n,s=0,y,x;
 printf("ENTER THE NUMBER = ");
 scanf("%d",&a);
 x=a;
 y=a;
 while(a!=0)
                {
                  a=a/10;
                  c=c+1;
                }
                while(x!=0)
                        {
                                n=x%10;
                                s=s+pow(n,c);
                                x=x/10;
                        }
                        if(s==y)
                                {
                                        printf("ARMSTRONG NUMBER");
                                }
```

```
                else
                        {
                           printf("NOT A ARMSTRONG NUMBER");
                        }

}
```

----Krishnamurti Number----

```
#include<stdio.h>
#include<math.h>
main()
{
 int a,x,n,i;
 long int s=1,k=0;
 printf("ENTER THE NUMBER = ");
 scanf("%d",&a);
 x=a;
 while(a!=0)
                {
                  n=a%10;
                  a=a/10;
                  s=1;
                  for(i=1;i<=n;i++)
                                {
                                 s=s*i;
                                }
                                k=k+s;

                }
```

```
        if(k==x)
                {
                printf("KRISHNAMURTI NUMBER");
                }
        else
                {
                printf("NOT A KRISHNAMURTI NUMBER");
                }
}
```

-----Madam Number or Palindrome of a number-----

```
#include<stdio.h>
int main()
{
 int x,n,r=0,a;
 printf("ENTER THE NUMBER : ");
 scanf("%d",&n);
 x=n;
 while(n!=0)
       {
        a=n%10;
        r=r*10+a;
        n=n/10;
       }
        if(x==r)
          {
            printf("THE NUMBER IS PALINDROME");
          }
          else
```

```
        {
          printf("THE NUMBER IS NOT A PALINDROME NUMBER");
        }

 return 0;
}
```

C code to find multiplication of two numbers without using star (*) operator:

```
#include<stdio.h>
int main()
{
  int a,s=0,b,c=0;
 printf("ENTER THE FIRST NUMBER : ");
 scanf("%d",&a);
printf("ENTER THE SECOND NUMBER : ");
 scanf("%d",&b);

 while(c!=b)
                        {
                                s=s+a;
                                c=c+1;
                        }

printf("%d",s);
return 0;
}
```

C codes on 1D Array:

---Array insertion---

```
#include <stdio.h>

int main()
{
  int array[100], position, c, n, value;

  printf("Enter number of elements in array\n");
  scanf("%d", &n);

  printf("Enter %d elements\n", n);

  for (c = 0; c < n; c++)
    scanf("%d", &array[c]);

  printf("Enter the location where you wish to insert an element\n");
  scanf("%d", &position);

  printf("Enter the value to insert\n");
  scanf("%d", &value);

  for (c = n - 1; c >= position - 1; c--)
    array[c+1] = array[c];

  array[position-1] = value;

  printf("Resultant array is\n");
```

```
  for (c = 0; c <= n; c++)
    printf("%d\n", array[c]);

  return 0;
}
```

---Array element deletion---

```
#include <stdio.h>

int main()
{
  int array[100], position, c, n;

  printf("Enter number of elements in array\n");
  scanf("%d", &n);

  printf("Enter %d elements\n", n);

  for ( c = 0 ; c < n ; c++ )
    scanf("%d", &array[c]);

  printf("Enter the location where you wish to delete element\n");
  scanf("%d", &position);

  if ( position >= n+1 )
    printf("Deletion not possible.\n");
  else
  {
    for ( c = position - 1 ; c < n - 1 ; c++ )
      array[c] = array[c+1];

    printf("Resultant array is\n");

    for( c = 0 ; c < n -1 ; c++ )
      printf("%d\n", array[c]);
```

```
  }

  return 0;
}
```

---Max and Min element in an Array---

```
#include <stdio.h>

int main()
{
   int arr[100];
   int i, max, min, size;

   /*
    * Reads size array and elements in the array
    */
   printf("Enter size of the array: ");
   scanf("%d", &size);
   printf("Enter elements in the array: ");
   for(i=0; i<size; i++)
   {
      scanf("%d", &arr[i]);
   }

   /* Supposes the first element as maximum and minimum */
   max = arr[0];
   min = arr[0];

   /*
    * Finds maximum and minimum in all array elements.
    */
   for(i=1; i<size; i++)
   {
      /* If current element of array is greater than max */
```

```
        if(arr[i]>max)
        {
           max = arr[i];
        }

        /* If current element of array is smaller than min */
        if(arr[i]<min)
        {
           min = arr[i];
        }
    }

    /*
     * Prints the maximum and minimum element
     */
    printf("Maximum element = %d\n", max);
    printf("Minimum element = %d", min);

    return 0;
}
```

---Occurrence of elements in an Array---

```
#include<stdio.h>
int main()
{
	int n,i,j,c=0,k,flag=0,a[100];
	printf("ENTER THE NUMBER OF ELEMENTS OF THE ARRAY : ");
	scanf("%d",&n);
	for(i=0;i<n;i++)
	{
		printf("ENTER THE ELEMENT[%d] : ",i);
		scanf("%d",&a[i]);
	}
	for(i=0;i<n;i++)
	{
		flag=0;
		c=0;
		for(j=0;j<i;j++)
		{
			if(a[i]==a[j])
			{
				flag=1;
				break;
			}
		}
		if(flag==0)
		{
			for(k=i;k<n;k++)
			{
				if(a[k]==a[i])
```

```
                {
                        c=c+1;
                }
            }
            printf("%d occures %d times \n",a[i],c);
        }
    }
    return 0;
}
```

---Bubble Sort ---

```
#include <stdio.h>
#include<stdlib.h>
int main()
{
	int *array, n, c, d, swap;
	printf("\nENTER SIZE OF ARRAY\n");
	scanf("%d", &n);
	array=(int *) malloc(n*sizeof(int));
	printf("ENTER %d INTEGERS\n", n);
		for (c=0;c<n; c++)
			scanf("%d", &array[c]);
		for (c=0;c<(n-1);c++)
		{
			for (d=0;d<n-c-1; d++)
			{
				if(array[d]>array[d+1])
					{
						swap=array[d];
						array[d]=array[d+1];
						array[d+1]=swap;
					}			}
		}
	printf("\nSORTED LIST IN ASCENDING ORDER:\n");
	for (c=0;c<n;c++)
		printf("%d\n", array[c]);
return 0;
}
```

---Selection Sort---

```
#include <stdio.h>
#include<stdlib.h>
int main()
{
	int *A, n, c, d, position, swap;
	printf("\nENTER SIZE OF ARRAY\n");
	scanf("%d",&n);
	A=(int*) malloc (n*sizeof(int));
	printf("ENTER ELEMENTS IN ARRAY\n");
	for (c=0;c<n;c++)
		scanf("%d", &A[c]);
	for(c=0;c<(n-1);c++)
	{
		position=c;
		for(d=c+1;d<n;d++)
		{
			if(A[position]>A[d])
				position = d;
		}
		if(position!=c)
		{
			swap=A[c];
			A[c]=A[position];
			A[position]=swap;
		}
	}
	printf("THE SORTED ARRAY=\n");
	for(c=0;c<n;c++)
```

```
            printf("%d\n", A[c]);
        return 0;
}
```

---Insertion Sort ---

```
#include<stdio.h>
#include<stdlib.h>
void ins_sort(int ar[],int n)
{
    int i,j,temp;
    for(i=1;i<n;i++)
    {
            temp=ar[i];
            j=i-1;
            while((temp<ar[j]) && (j>=0))
            {
                ar[j+1]=ar[j];
                j--;
            }
           ar[j+1]=temp;
    }
}
int main()
{
   int ar[10],n,i;
           printf("Enter the size\n");
           scanf("%d",&n);
           for(i=0;i<n;i++)
           {
                   printf("\nEnter the %dth element:\n",(i+1));
                   scanf("%d",&ar[i]);
           }
           ins_sort(ar,n);
```

```
        printf("The sorted array is:\n");
        for(i=0;i<n;i++)
        {
            printf("%d ",ar[i]);
    }

        //system("PAUSE");
        return 0;
}
```

---Linear Search ---

```
#include<stdio.h>
#include<stdlib.h>
main()
{
	int *A,a,b,c=0,i;
	printf("\nENTER RANGE OF ARRAY\n");
	scanf("%d", &a);
	A=(int*) malloc(a*sizeof(int));
	printf("\nENTER ELEMENTS IN ARRAY\n");
	for(i=0;i<a;i++)
		scanf("%d", &A[i]);
	printf("\nENTER ELEMENT TO BE SEARCHED\n");
	scanf("%d", &b);
	for(i=0;i<a;i++)
	{
		if(A[i]==b)
		{
			printf("\nELEMENT %d FOUND IN %d POSITION\n",b,i+1);
			c++;
		}
	}
	if(c==0)
		printf("\nELEMENT %d NOT FOUND IN ARRAY\n",b);
}
```

---Binary Search---

```
#include<stdio.h>
#include<stdlib.h>
int bs(int *,int ,int ,int );
int main()
{
    int *p,i,f,n,c,s=0,e;
    printf("ENTER THE NUMBER OF ELEMENTS WANT TO INPUT : ");
    scanf("%d",&n);
    p=(int *)malloc(n*sizeof(int));
    printf("\n\nENTER THE ELEMENTS IN ASCENDING ORDER \n\n");
    for(i=0;i<n;i++)
    {
        printf("ENTER THE ELEMENT: ");
      scanf("%d",(p+i));
    }
    e=n-1;
    printf("ENTER THE ELEMENT WANT TO SEARCH : ");
    scanf("%d",&f);
    c=bs(p,s,e,f);
    if(c==-1)
    {
        printf("THE ELEMENT IS NOT FOUND");
    }
    else
    {
        printf("THE ELEMENT %d IS FOUND AT POSITION %d\n",f,c);
    }
    return 0;
```

```
}
int bs(int *a,int start,int end,int search)
{
  int mid;
  mid=(start+end)/2;
  if(start>end)
  {
        return -1;
  }
  if(*(a+mid)==search)
  {
        return mid+1;
  }
  if(search<*(a+mid))
  {
        end=mid-1;
    bs(a,start,end,search);
  }
  else if(search>*(a+mid))
  {
        start=mid+1;
    bs(a,start,end,search);
  }
}
```

C code on 2D Array:

----Lower Triangular matrix-----

```
#include <stdio.h>


int main(){
   int rows, cols, size, rowCounter, colCounter;
   int inputMatrix[50][50];
   printf("Enter size square matrix\n");
   scanf("%d", &size);
   rows = cols = size;


   printf("Enter Matrix of size %dX%d\n", rows, cols);
   /*  Input matrix*/
   for(rowCounter = 0; rowCounter < rows; rowCounter++){
      for(colCounter = 0; colCounter < cols; colCounter++){
         scanf("%d", &inputMatrix[rowCounter][colCounter]);
      }
   }
   /*
    Printing lower triangular matrix
    */
   printf("Lower triangular Matrix\n");
   for(rowCounter = 0; rowCounter < rows; rowCounter++){
      for(colCounter = 0; colCounter < cols; colCounter++){
         if(rowCounter < colCounter){
            /* Upper triangle element*/
            //printf("%d ", 0);
            printf("_\t");
         } else {
```

```
            /* Lower triagle element*/
            printf("%d \t", inputMatrix[rowCounter][colCounter]);
          }
        }
        printf("\n");
    }

    return 0;
}
```

----Upper Triangular matrix-----

```
#include <stdio.h>

int main(){
   int rows, cols, size, rowCounter, colCounter;
   int inputMatrix[50][50];
   printf("Enter size square matrix\n");
   scanf("%d", &size);
   rows = cols = size;

   printf("Enter Matrix of size %dX%d\n", rows, cols);
   /*  Input matrix*/
   for(rowCounter = 0; rowCounter < rows; rowCounter++){
      for(colCounter = 0; colCounter < cols; colCounter++){
         scanf("%d", &inputMatrix[rowCounter][colCounter]);
      }
   }
   /*
    Printing upper triangular matrix
    L[i,j] = 0, If i > j and L[i,j] = l[i,j], If i <= j
   */
   printf("Upper triangular Matrix\n");
   for(rowCounter = 0; rowCounter < rows; rowCounter++){
      for(colCounter = 0; colCounter < cols; colCounter++){
         if(rowCounter > colCounter){
            /* Lower triangle element*/
            //printf("%d ", 0);
            printf("_\t");
         } else {
```

```
            /* Upper triagle element*/
            printf("%d \t", inputMatrix[rowCounter][colCounter]);
          }
        }
        printf("\n");
    }

    return 0;
}
```

----Transpose of a Matrix----

```
#include<stdio.h>
int main()
{
	int m,n,i,j,c[100][100],a[100][100];
	printf("Enter the number of rows and columns of matrix :  ");
	scanf("%d%d",&m,&n);
	for(i=0;i<m;i++)
	{
		for(j=0;j<n;j++)
		{
			printf("Enter the element_[%d][%d] : ",i,j);
			scanf("%d",&c[i][j]);
		}
	}
	printf("\nTHE ORIGINAL MATRIX IS : \n");
	for(i=0;i<m;i++)
	{
		for(j=0;j<n;j++)
		{
			printf("%d\t",c[i][j]);
		}
		printf("\n");
	}
	for(i=0;i<m;i++)
	{
		for(j=0;j<n;j++)
		{
			a[j][i] = c[i][j];
```

```
            }
        }
        printf("\nTRANSPOSE OF THE GIVEN MATRIX IS GIVEN BELOW :\n");
        for(i=0;i<n;i++)
        {
            for(j=0;j<m;j++)
            {
                printf("%d\t",a[i][j]);
            }
            printf("\n");
        }
        return 0;
}
```

-----Max element in a matrix----

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

main()
{
  int m, n, c, d, matrix[10][10], maximum;

  printf("Enter the number of rows and columns of matrix\n");
  scanf("%d%d",&m,&n);
  printf("Enter the elements of matrix\n");

  for( c = 0 ; c < m ; c++ )
  {
    for( d = 0 ; d < n ; d++ )
    {
      scanf("%d",&matrix[c][d]);
    }
  }

  maximum = matrix[0][0];

  for( c = 0 ; c < m ; c++ )
  {
    for( d = 0 ; d < n ; d++ )
    {
      if ( matrix[c][d] > maximum )
        maximum = matrix[c][d];
    }
```

```
  }

  printf("Maximum element in matrix is %d\n", maximum);
  return 0;
}
```

-----Min Element in a matrix----

```
# include <stdio.h>
main()
{
int mat[10][10] ;
int i, j, row, col, small ;
printf("Enter the row and column of the matrix : ") ;
scanf("%d %d", &row, &col) ;
printf("\nEnter the elements of the matrix : \n\n") ;
for(i = 0 ; i < row ; i++)
  for(j = 0 ; j < col ; j++)
    scanf("%d", &mat[i][j]) ;
small = mat[0][0] ;
for(i = 0 ; i < row ; i++){
  for(j = 0 ; j < col ; j++){
    if(mat[i][j] < small)
      small = mat[i][j] ;
  }
}
printf("\nThe smallest element in the matrix is : %d\n\n",small);
}
```

-------Sum of Diagonal elements of a Matrix------

```
#include <stdio.h>
int main(){
   int rows, cols, rowCounter, colCounter, diagonalSum = 0;
   int inputMatrix[50][50];
   printf("Enter Rows and Columns of Matrix\n");
   scanf("%d %d", &rows, &cols);

   printf("Enter first Matrix of size %dX%d\n", rows, cols);
   /*  Input first matrix*/
   for(rowCounter = 0; rowCounter < rows; rowCounter++){
      for(colCounter = 0; colCounter < cols; colCounter++){
         scanf("%d", &inputMatrix[rowCounter][colCounter]);
      }
   }
   /* Sum diagonal elements of input matrix. Diagonal elements are those
     elements whose row and column indexes are same. */
   for(rowCounter = 0; rowCounter < rows; rowCounter++){
      for(colCounter = 0; colCounter < cols; colCounter++){
         if(rowCounter == colCounter){
            diagonalSum += inputMatrix[rowCounter][colCounter];
         }
      }
   }

   printf("Sum of all diagonal elements of Matrix is: %d\n", diagonalSum);
   return 0;
}
```

-------Sum of Diagonal elements of a Matrix by not visiting all the matrix------

```
#include <stdio.h>
int main(){
   int rows, cols, rowCounter, colCounter, diagonalSum = 0;
   int inputMatrix[50][50];
   printf("Enter Rows and Columns of Matrix\n");
   scanf("%d %d", &rows, &cols);

   printf("Enter first Matrix of size %dX%d\n", rows, cols);
   /*  Input first matrix*/
   for(rowCounter = 0; rowCounter < rows; rowCounter++){
      for(colCounter = 0; colCounter < cols; colCounter++){
         scanf("%d", &inputMatrix[rowCounter][colCounter]);
      }
   }
   /* Sum diagonal elements of input matrix. Diagonal elements are those
     elements whose row and column indexes are same.
     For Example: Matrix[1][1], Matrix[4][4] */
   for(rowCounter = 0; rowCounter < rows; rowCounter++){
      //if(rowCounter <= cols-1) {
         diagonalSum += inputMatrix[rowCounter][rowCounter];
      //}
   }
   printf("Sum of all diagonal elements of Matrix is: %d\n", diagonalSum);
   return 0;
}
```

-----Print the matrix diagonally------

```
#include<stdio.h>
int main(){
   int rows, cols, rowCounter, colCounter, currentRow, currentCol;
   int inputMatrix[50][50];

   /*  Input matrix*/
   printf("Enter size of matrix\n");
   scanf("%d %d", &rows, &cols);

   printf("Enter the matrix of size %dX%d\n", rows, cols);
   for(rowCounter = 0; rowCounter < rows; rowCounter++){
      for(colCounter = 0; colCounter < cols; colCounter++){
         scanf("%d", &inputMatrix[rowCounter][colCounter]);
      }
   }
   printf("Printing matrix diagonally\n");
   // Print Upper half of matrix
   for(colCounter = 0; colCounter < cols; colCounter++)
          {
      currentCol = colCounter;
                    currentRow = 0;
      for(;currentCol >= 0 && currentRow < rows; currentCol--, currentRow++){
         printf("%d ", inputMatrix[currentRow][currentCol]);
      }
      printf("\n");
   }

   // Print Lower half of matrix
```

```
    for(rowCounter = 1; rowCounter < rows; rowCounter++){
        currentCol = cols -1;
                        currentRow = rowCounter;
        for(;currentCol >= 0 && currentRow < rows; currentCol--, currentRow++){
            printf("%d ", inputMatrix[currentRow][currentCol]);
        }
        printf("\n");
    }

    return 0;
}
```
