MCSL 025 2016-2017 session Ignou Study Helper

www.ignousite.blogspot.com

PART-1: MCS-021

Q.1.Write a program in C language for multiplication of two sparse matrices using Pointers?

A.1. A matrix in which number of zero entries are much higher than the number of non zero entries is called sparse matrix. The natural method of representing matrices in memory as two-dimensional arrays may not be suitable foe sparse matrices. One may save space by storing for only non zero entries. For example matrix A (4*4 matrix) represented below

where first row represent the dimension of matrix and last column tells the number of non zero values; second row onwards it is giving the position and value of non zero number.

```
 #include <stdio.h>

#include <conio.h>

#include <alloc.h>


#define MAX1 3

#define MAX2 3

#define MAXSIZE 20


#define TRUE 1

#define FALSE 2


struct sparse

{

int *sp ;

int row ;

int *result ;
```

```
} ;

voidinitsparse ( struct sparse * ) ;
voidcreate_array ( struct sparse * ) ;
int count ( struct sparse ) ;
void display ( struct sparse ) ;
voidcreate_tuple ( struct sparse*, struct sparse ) ;
voiddisplay_tuple ( struct sparse ) ;
voidprodmat ( struct sparse *, struct sparse, struct sparse ) ;
voidsearchina ( int *sp, int ii, int*p, int*flag ) ;
voidsearchinb ( int *sp, intjj, intcolofa, int*p, int*flag ) ;
voiddisplay_result ( struct sparse ) ;
voiddelsparse ( struct sparse * ) ;

void main( )
{
struct sparse s[5] ;
int i ;

clrscr( ) ;

for ( i = 0 ; i <= 3 ; i++ )
initsparse ( &s[i] ) ;

create_array( &s[0] ) ;

create_tuple( &s[1], s[0] ) ;
```

```
display_tuple( s[1] ) ;

create_array( &s[2] ) ;

create_tuple( &s[3], s[2] ) ;
display_tuple( s[3] ) ;

prodmat ( &s[4], s[1], s[3] ) ;

printf ( "\nResult of multiplication of two matrices: " ) ;
display_result( s[4] ) ;

for ( i = 0 ; i <= 3 ; i++ )
delsparse ( &s[i] ) ;

getch( ) ;
}

/* initialises elements of structure */
voidinitsparse ( struct sparse *p )
{
p ->sp = NULL ;
p -> result = NULL ;
}

/* dynamically creates the matrix */
voidcreate_array ( struct sparse *p )
```

```
{
int n, i ;

   /* allocate memory */

p ->sp = ( int * ) malloc ( MAX1 * MAX2 * sizeof ( int ) ) ;

   /* add elements to the array */
for ( i = 0 ; i < MAX1 * MAX2 ; i++ )
   {
printf ( "Enter element no. %d: ", i ) ;
scanf ( "%d", &n ) ;
         * ( p ->sp + i ) = n ;
   }
}

/* displays the contents of the matrix */
void display ( struct sparse s )
{
int i ;

   /* traverses the entire matrix */
for ( i = 0 ; i < MAX1 * MAX2 ; i++ )
   {
      /* positions the cursor to the new line for every new row */
if ( i % 3 == 0 )
printf ( "\n" ) ;
printf ( "%d\t", * ( s.sp + i ) ) ;
    }
}

/* counts the number of non-zero elements */
int count ( struct sparse s )
{
intcnt = 0, i ;

for ( i = 0 ; i < MAX1 * MAX2 ; i++ )
    {
if ( * ( s.sp + i ) != 0 )
cnt++ ;
    }
returncnt ;
}

/* creates an array that stores information about non-zero elements */
voidcreate_tuple ( struct sparse *p, struct sparse s )
{
int r = 0 , c = -1, l = -1, i ;

    /* get the total number of non-zero elements */

p -> row = count ( s ) + 1 ;

    /* allocate memory */
```

```
p ->sp = ( int * ) malloc ( p -> row * 3 * sizeof ( int ) ) ;

    /* store information about
total no. of rows, cols, and non-zero values */

    * ( p ->sp + 0 ) = MAX1 ;
    * ( p ->sp + 1 ) = MAX2 ;
    * ( p ->sp + 2 ) = p -> row - 1 ;

    l = 2 ;

    /* scan the array and store info. about non-zero values
in the 3-tuple */
for ( i = 0 ; i < MAX1 * MAX2 ; i++ )
    {
c++ ;

        /* sets the row and column values */
if ( ( ( i % 3 ) == 0 ) && ( i != 0 ) )
        {
            r++ ;
            c = 0 ;
        }

        /* checks for non-zero element,
row, column and non-zero value
```

```
is assigned to the matrix */
if ( * ( s.sp + i ) != 0 )
        {
            l++ ;
            * ( p ->sp + l ) = r ;
            l++ ;
            * ( p ->sp + l ) = c ;
            l++ ;
            * ( p ->sp + l ) = * ( s.sp + i ) ;
        }
    }
}

/* displays the contents of the matrix */
voiddisplay_tuple ( struct sparse s )
{
int i, j ;

    /* traverses the entire matrix */

printf ( "\nElements in a 3-tuple: " ) ;

    j = ( * ( s.sp + 2 ) * 3 ) + 3 ;

for ( i = 0 ; i < j ; i++ )
    {
        /* positions the cursor to the new line for every new row */
```

```
if ( i % 3 == 0 )

printf ( "\n" ) ;

printf ( "%d\t", * ( s.sp + i ) ) ;

    }

printf ( "\n" ) ;

}


/* performs multiplication of sparse matrices */

voidprodmat ( struct sparse *p, struct sparse a, struct sparse b )

{

int sum, k, position, posi, flaga, flagb, i , j ;

    k = 1 ;


p -> result = ( int * ) malloc ( MAXSIZE * 3 * sizeof ( int ) ) ;


for ( i = 0 ; i < * ( a.sp + 0 * 3 + 0 ) ; i++ )

    {

for ( j = 0 ; j < * ( b.sp + 0 * 3 + 1 ) ; j++  )

        {

            /* search if an element present at ith row */


searchina ( a.sp, i, &position, &flaga ) ;

if ( flaga == TRUE )

            {

sum = 0 ;


                /* run loop till there are element at ith row
in first 3-tuple */
while ( * ( a.sp + position * 3 + 0 ) == i )
                {
                    /* search if an element present at ith col.
in second 3-tuple */

searchinb ( b.sp, j, * ( a.sp + position * 3 + 1 ),
&posi, &flagb ) ;

                    /* if found then multiply */
if ( flagb == TRUE )
sum = sum + * ( a.sp + position * 3 + 2 ) *
                                * ( b.sp + posi * 3 + 2 ) ;
position = position + 1 ;
                }

                /* add result */
if ( sum != 0 )
                {
                    * ( p -> result + k * 3 + 0 ) = i ;
                    * ( p -> result + k * 3 + 1 ) = j ;
                    * ( p -> result + k * 3 + 2 ) = sum ;
                    k = k + 1 ;
                }
            }
        }
    }
    /* add total no. of rows, cols and non-zero values */

    * ( p -> result + 0 * 3 + 0 ) = * ( a.sp + 0 * 3 + 0 ) ;
    * ( p -> result + 0 * 3 + 1 ) = * ( b.sp + 0 * 3 + 1 ) ;
    * ( p -> result + 0 * 3 + 2 ) = k - 1 ;
}

/* searches if an element present at iith row */

voidsearchina ( int *sp, int ii, int *p, int *flag )
{
int j ;
    *flag = FALSE ;
for ( j = 1 ; j <= * ( sp + 0 * 3 + 2 ) ; j++ )
    {
if ( * ( sp + j * 3 + 0 ) == ii )
        {
            *p = j ;
            *flag = TRUE ;
return ;
        }
     }
}

/* searches if an element where col. of first 3-tuple
is equal to row of second 3-tuple */
```

```
voidsearchinb ( int *sp, intjj, intcolofa, int *p, int *flag )
{
int j ;
    *flag = FALSE ;
for ( j = 1 ; j <= * ( sp + 0 * 3 + 2 ) ; j++ )
    {
if ( * ( sp + j * 3 + 1 ) == jj&& * ( sp + j * 3 + 0 ) == colofa )
        {
            *p = j ;
            *flag = TRUE ;
return ;
        }
    }
}

/* displays the contents of the matrix */
voiddisplay_result ( struct sparse s )
{
int i ;

    /* traverses the entire matrix */
for ( i = 0 ; i < ( * ( s.result + 0 + 2 ) + 1 ) * 3 ; i++ )
    {
        /* positions the cursor to the new line for every new row */
if ( i % 3 == 0 )
printf ( "\n" ) ;

printf ( "%d\t", * ( s.result + i ) ) ;

    }

}


/* deallocates memory */


voiddelsparse ( struct sparse *s )

{

if ( s ->sp != NULL )

free ( s ->sp ) ;

if ( s -> result != NULL )

free ( s -> result ) ;

}
```

Q.2. Write a program in C language that will accept a Graph as input and will perform a Depth First Search on it. Make necessary assumptions.

A.2.

DFS:-

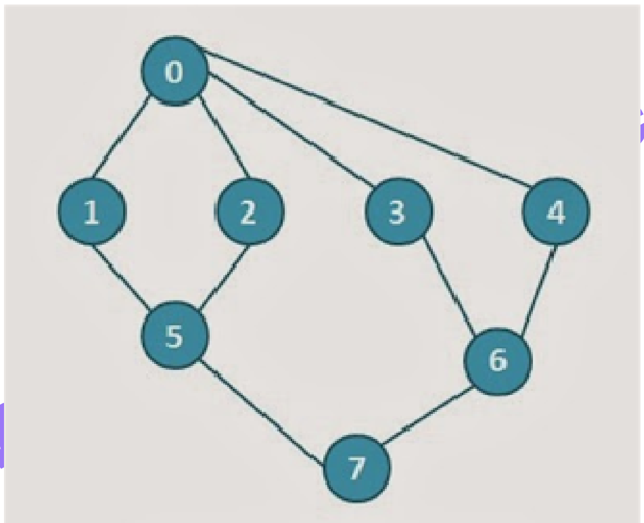
Depth First Search is an algorithm used to search the Tree or Graph. DFS search starts from root node then traversal into left child node and continues, if item found it stops otherwise it continues.

The advantage of DFS is it requires less memory compare to Breadth First Search(BFS).

n← number of nodes

Initialize visited[ ] to false (0)

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

    visited[i] = 0;

void DFS(vertex i) [DFS starting from i]
{
    visited[i]=1;
    for each w adjacent to i
        if(!visited[w])
            DFS(w);
}
```

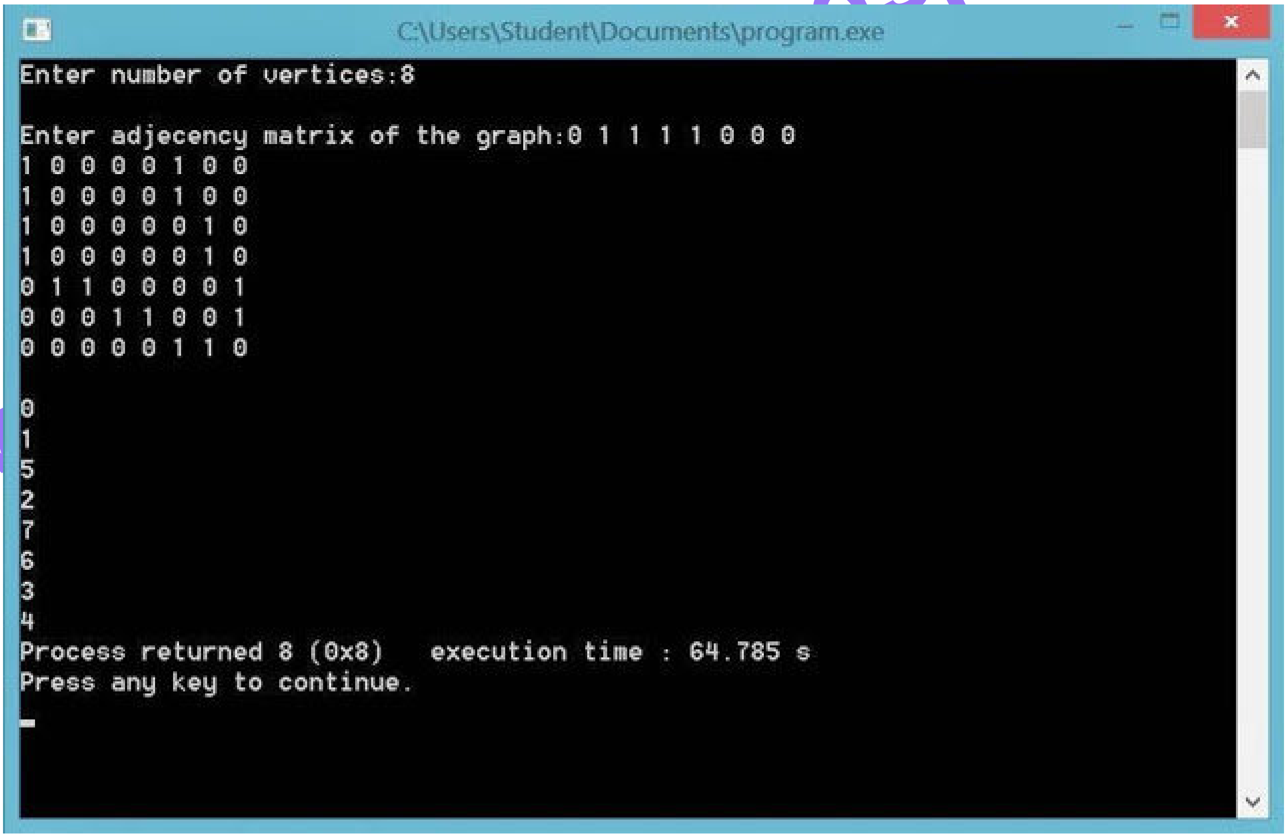


Graph used for input

Depth First Search (DFS) Program in C [Adjacency Matrix]

```
#include<stdio.h>

void DFS(int);
int G[10][10],visited[10],n;    //n is no of vertices and graph is sorted in array G[10][10]

void main()
{
inti,j;
printf("Enter number of vertices:");

         scanf("%d",&n);

   //read the adjecency matrix
         printf("\nEnteradjecency matrix of the graph:");

         for(i=0;i<n;i++)
for(j=0;j<n;j++)
                         scanf("%d",&G[i][j]);

   //visited is initialized to zero
for(i=0;i<n;i++)
visited[i]=0;

DFS(0);
}
```

```
void DFS(int i)
{
int j;
          printf("\n%d",i);
visited[i]=1;

          for(j=0;j<n;j++)
if(!visited[j]&&G[i][j]==1)
DFS(j);
}
```



```
Enter number of vertices:8

Enter adjecency matrix of the graph:0 1 1 1 1 0 0 0
1 0 0 0 0 1 0 0
1 0 0 0 0 1 0 0
1 0 0 0 0 0 1 0
1 0 0 0 0 0 1 0
0 1 1 0 0 0 0 1
0 0 0 1 1 0 0 1
0 0 0 0 0 1 1 0

0
1
5
2
7
6
3
4
Process returned 8 (0x8)   execution time : 64.785 s
Press any key to continue.
```

PART-2: MCS-022

Q.1.Write a shell script in Linux/Unix that accepts a text file as input and prints the number of sentences in the file.

A.1. A shell script is a computer program designed to be run by the Unix shell, a command – line Interpreter .The various dialects of shell scripts are considered to be scripting languages.

Typical operations performed by shell scripts include file manipulation, program execution, and printing text. A script which sets up the environment, runs the program, and does any necessary cleanup, logging, etc. is called a wrapper.

```
echo Enter a text

read text


w=`echo $text | wc -w`

w=`expr $w`

c=`echo $text | wc -c`

c=`expr $c - 1`

s=0

alpha=0

j=` `

n=1

while [ $n -le $c ]

do

ch=`echo $text | cut -c $n`

if test $ch =  $j

then

s=`expr $s + 1`

fi
```

```
case $ch in

(a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z) alpha=`expr $alpha + 1`;;

esac

n=`expr $n + 1`

done

special=`expr $c - $s - $alpha`

echo Words=$w

echo Characters=$c

echo Spaces=$s

echo Special symbols=$special

sentences = Words+ Characters+ Spaces + Special Symbols

echo $Sentences
```

Q.2.Your PC is on a network. Make necessary settings in your PC so that it can Print to a Printer that is on the Network of PC but not directly connected.

A.2. STEPS To Connect Printer:-

1. Click on Start in the bottom left corner of your screen. A popup list will appear.
2. Select Control Panel from the popup list. Type the word network in the search box.
3. Click on Network and Sharing Center.
4. Click on Change advanced shared settings, in the left pane.
5. Click on the down arrow, which will expand the network profile.
6. Select File and printer sharing and choose Turn on file and printer sharing.
7. Click on Save changes.

You're now ready to share your printer.

1. Click on Start in the bottom left corner of your screen. A popup list will appear.
2. Click on Devices and Printers, from the popup list.
3. Right click the printer you want to share. A dropdown list will appear.

4. Select Printer properties from the dropdown list.

5. Click on the Sharing tab

6. Select the Share this printer check box.

In order for other people to connect to the printer, they just have to add the network printer that you just opened for sharing to their computers. Here's how to do this.

1. Click on Start in the bottom left corner of your screen. A popup list will appear.

2. Click on Devices and Printers from the popup list.

3. Select Add a printer.

4. Click on Add a network, wireless or Bluetooth printer.

5. Click the shared printer.

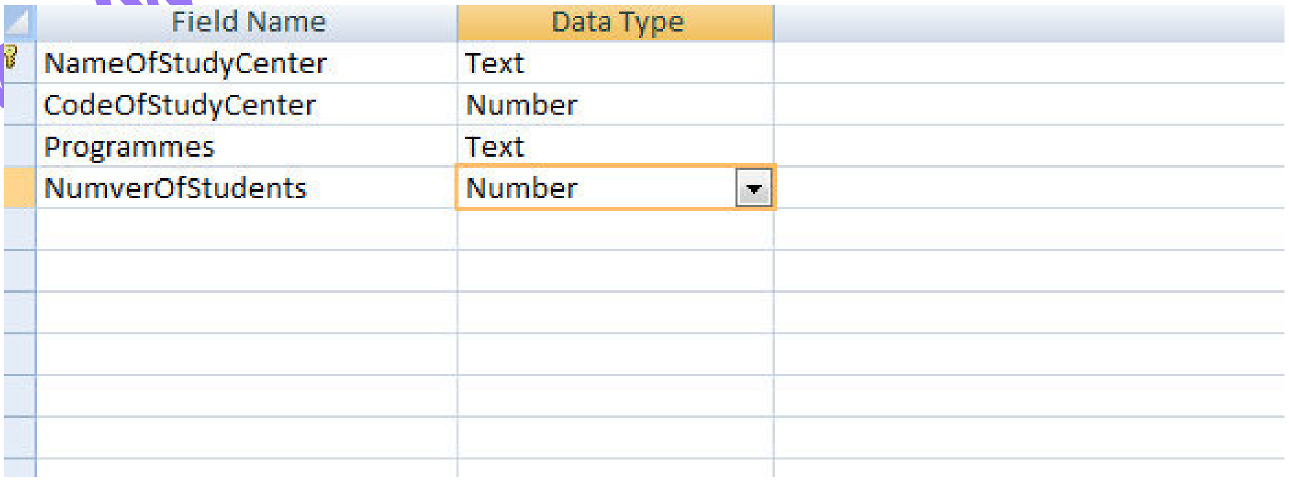
6. Click Next. Continue according to the instructions on the screen.

PART-3: MCS-023

Q.1.Create a database consisting of Name of Study Center, Code of Study Center, Programmes offered at Study Center, Number of Students enrolled Programme Wise.After creating the database, perform the following tasks:(i) List the number of Students who are enrolled for MCA across all Study Centers

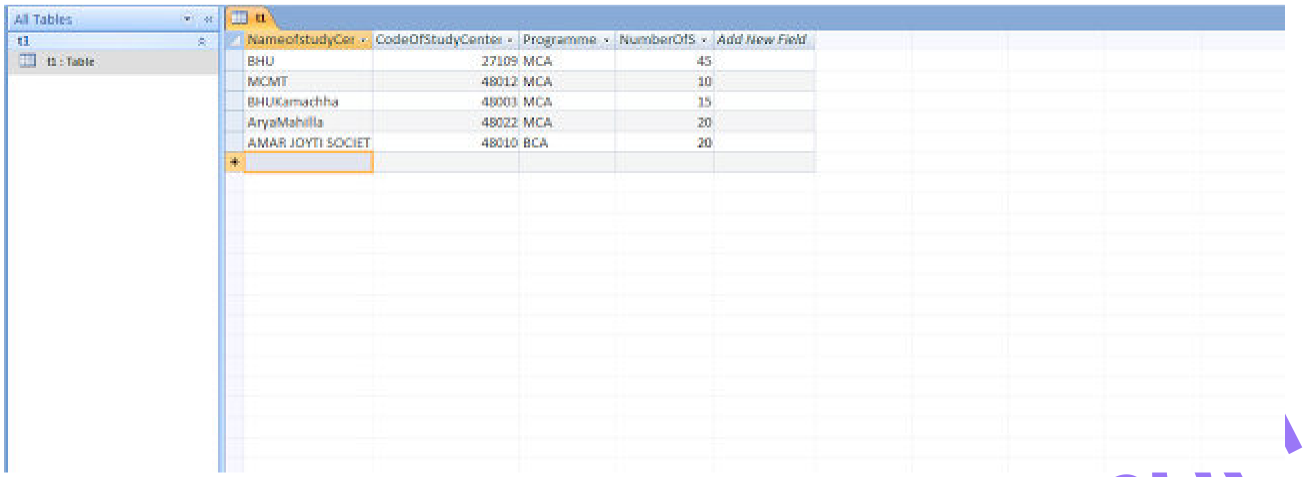
A.1. Step 1)

Create Table :-



Field Name	Data Type
NameOfStudyCenter	Text
CodeOfStudyCenter	Number
Programmes	Text
NumverOfStudents	Number

Step 2:- Inserting Values In Table:-



NameofstudyCer	CodeOfStudyCenter	Programme	NumberOfS	Add New Field
BHU	27109	MCA	45	
MCMT	48012	MCA	10	
BHUKamachha	48003	MCA	15	
AryaMahilla	48022	MCA	20	
AMAR JOYTI SOCIET	48010	BCA	20	

Step3:- Generate a Select Query Statment:-

Select * from t1 where Prorammes = 'MCA'

And Execute it.

Step 4:-

NameofStudyCenter	CodeOfStudyCenter	Prorammes	NumberOfStudent
BHU	27109	MCA	45
MCMT	48012	MCA	10
BHUKamachha	48003	MCA	15
AryaMahila	48022	MCA	20

Q.1.Write a program in Java for the addition of two matrices.

A.1.

Array equal to the number of rows of the matrix and the length of the sub arrays equal to the number of columns of the matrix. For example, a matrix of order 3*7 will be represented as a 2D array matrix[3][7]. A two level nested for loop will be used to read the input matrices from the keyboard. The outer loop counter, i ranges from 0 to the number of rows of the matrix while the inner loop counter, j ranges from 0 to the number of columns of the matrix. Within the inner loop, the input integers will be read using nextInt() method of the scanner class and stored at position [i][j] of the array.

```
import java.util.Scanner;

public class MatrixAddition {

  public static void main(String[] args) {
     Scanner s = new Scanner(System.in);
     System.out.print("Enter number of rows: ");
     int rows = s.nextInt();
     System.out.print("Enter number of columns: ");
     int columns = s.nextInt();
     int[][] a = new int[rows][columns];
     int[][] b = new int[rows][columns];
     System.out.println("Enter the first matrix");
     for (int i = 0; i < rows; i++) {
        for (int j = 0; j < columns; j++) {
           a[i][j] = s.nextInt();
        }
     }
     System.out.println("Enter the second matrix");
     for (int i = 0; i < rows; i++) {
        for (int j = 0; j < columns; j++) {
           b[i][j] = s.nextInt();
        }
     }
     int[][] c = new int[rows][columns];
     for (int i = 0; i < rows; i++) {
        for (int j = 0; j < columns; j++) {
           c[i][j] = a[i][j] + b[i][j];
        }
     }
```

```
        System.out.println("The sum of the two matrices is");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < columns; j++) {
                System.out.print(c[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

Here is a sample execution.

```
Enter number of rows: 2
Enter number of columns: 3
Enter the first matrix
3 4 7
1 8 4
Enter the second matrix
3 2 1
1 0 4
The sum of the two matrices is
6 6 8
2 8 8
```

Q.2.Write a program in Java that connects to a database and generates a report consisting of the Programmes study center wise where the student enrollment is less than 50. Make assumptions wherever necessary.

A.2.

```
import java.sql.*;      // Use classes in java.sql package


                               // JDK 7 and above




public class JdbcSelectTest                 { // Save as "JdbcSelectTest.java"


public static void main(String[] args) {
try (
// Step 1: Allocate a database "Connection" object
```

```
Connection conn = DriverManager.getConnection(
"jdbc:mysql://localhost:8888/ebookshop", "myuser", "xxxx"); // MySQL


// Connection conn = DriverManager.getConnection(
// "jdbc:odbc:ebookshopODBC"); // Access


// Step 2: Allocate a "Statement" object in the Connection
Statement stmt = conn.createStatement();


 {



// Step 3: Execute a SQL SELECT query, the query result
// is returned in a "ResultSet" object.




String strSelect = "select title, price, qty from books";
System.out.println("The SQL query is: " + strSelect); // Echo For debugging
System.out.println();



ResultSet rset = stmt.executeQuery(strSelect);
// Step 4: Process the ResultSet by scrolling the cursor forward via next().



// For each row, retrieve the contents of the cells with getXxx(columnName).
System.out.println("The records selected are:");



int rowCount = 0;
while(rset.next()) { // Move the cursor to the next row}
String title = rset.getString("title");


double price = rset.getDouble("price");
int qty = rset.getInt("qty");


System.out.println(title + ", " + price + ", " + qty);++rowCount;
}
System.out.println("Total number of records = " + rowCount);
} catch(SQLException ex) {
ex.printStackTrace();}}}
```