

C pro to find maximum element in row & column

Karunakar
6AUA05063

Algorithm

Flowchart

Step 1: Start

Step 2: read r, c

Step 3: Enter the elements

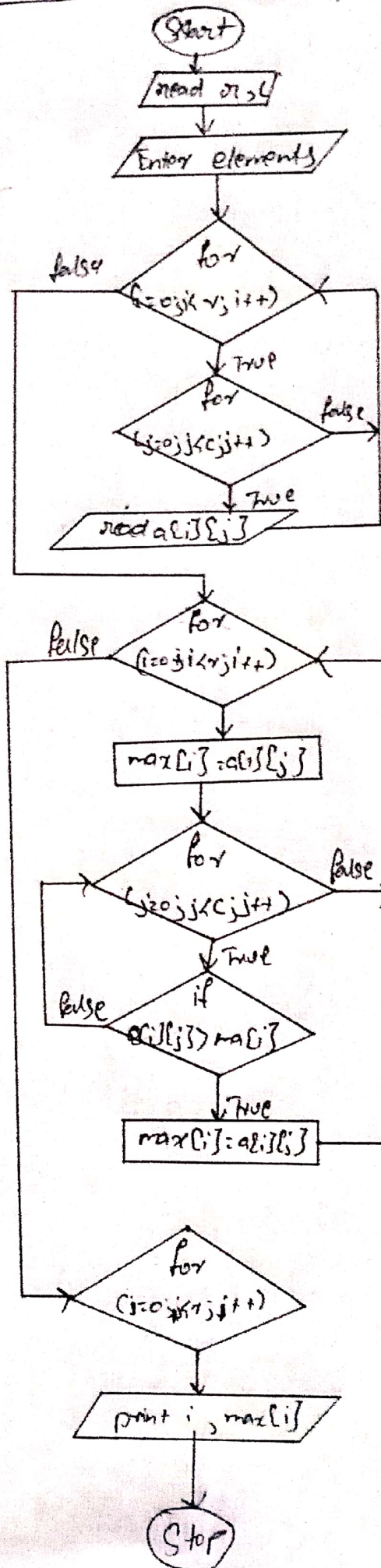
```
for(i=0; i<r; i++)
{
    for(j=0; j<c; j++)
        read a[i][j]
}
```

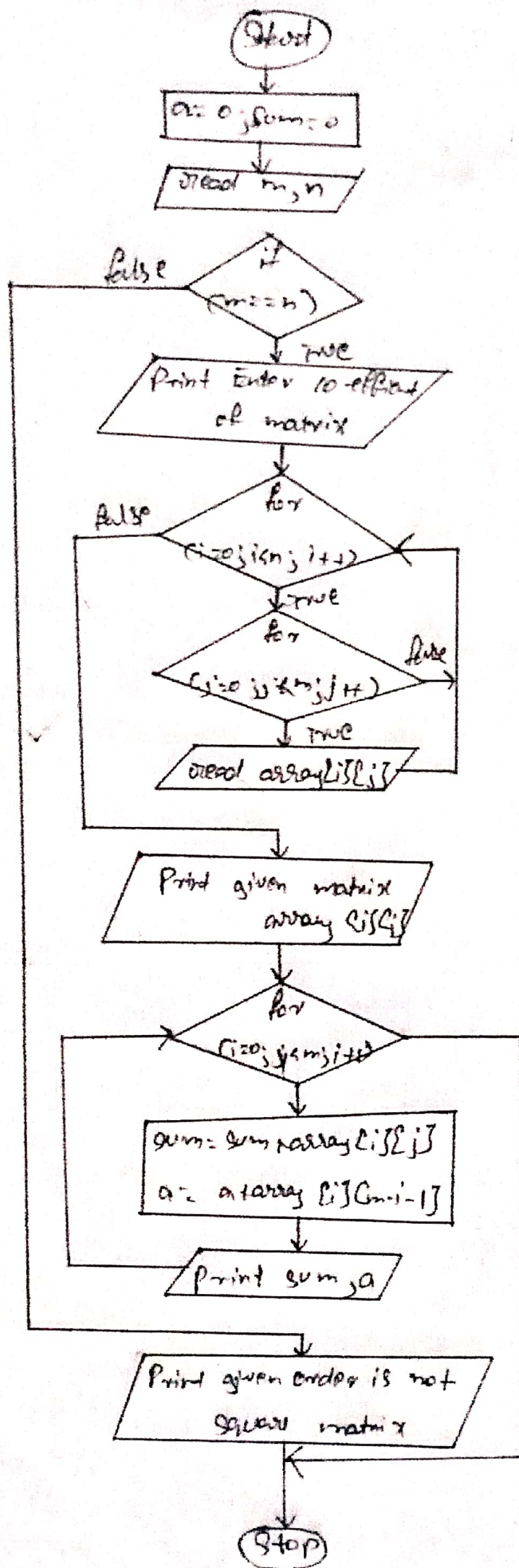
Step 4: print the entered element
 $a[i][j]$

```
Steps: for(i=0; i<r; i++)
{
    max = 0;
    for(j=0; j<c; j++)
    {
        if(a[i][j] > max)
        {
            max = a[i][j]
        }
    }
}
```

Step 6 :- for(i=0; i<r; i++)
{
 print i, max[i]
}

Step 7: Stop





C pro to implement sum of principle diagonal and secondary diagonal
element. LAUG/2003

Algorithm

Step 1: Start

Step 2: $sum = 0$; $qum = 0$

Step 3: Enter order of matrix

read m, n

Step 4: if $(m \neq n)$ // if false goto Step 4.5

print "Enter the co-efficient of matrix"

for $(i = 0; i < m; i++)$

{

for $(j = 0; j < n; j++)$

read array $[i][j]$

}

4.1. print "The given matrix is"

output matrix element array $[i][j]$

4.2. for $(j = 0; j < m; j++)$

{

$sum = sum + array[i][j]$

$q = q + array[j][m-i-1]$

}

4.3. print sum of main diagonal

print sum

4.4. print sum of diagonal

print q

4.5. else

print "The given matrix is not square matrix"

Step 5: Stop

C pro to find transpose of matrix

Karurathar
GALIA CS063

Algorithm

Step 1 :- start

Step 2 :- Read the value for m, n

Step 3 :- Enter element of matrix
for $(i=0; i < m; i++)$

for $(d=0; d < n; d++)$
read matrix $[c][d]$

Step 4 :- for $(c=0; c < m; c++)$
for $(d=0; d < n; d++)$

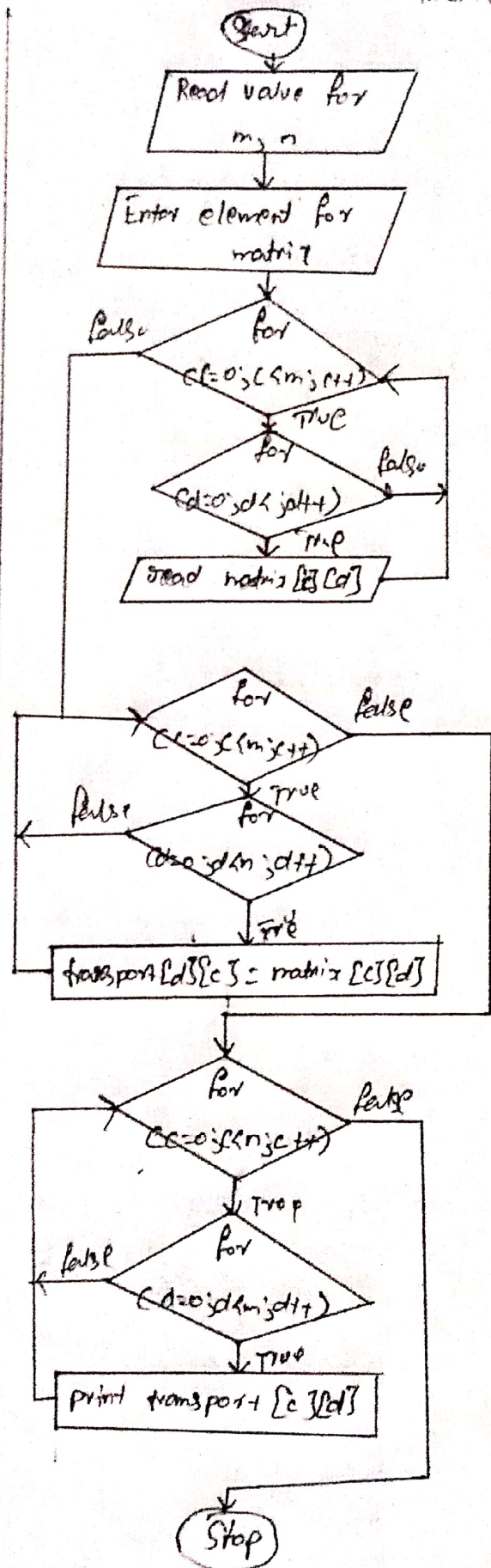
transpose $[d][c] = \text{matrix}[c][d]$

Step 5 :- for $(c=0; c < n; c++)$

for $(d=0; d < m; d++)$

print transpose $[c][d]$

Step 6 :- Stop



C program to find 2 array are similar.

Flowchart

Karunakar
CALCULUS

Algorithm

Step 1: Start

Step 2:- Read value for n

Step 3:- Enter the array element

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

read $a[i]$

Step 4:- Enter the 2nd array element

for $(j=0; j < n; j++)$

read $b[j]$

Step 5:- for $(i=0; i < n; i++)$

{ for $(j=0; j < n; j++)$

{ if $(a[i] == b[j])$

k++

}

}

Step 6:- if $(k == n)$

print similar

else
print not similar

Step 7: Stop

