Matrix Analysis using C

K Prasanna Kumar

Contents

***Abstract*—This module helps to learn Matrix Theory using C. Matrix Theory in C programming is required by a Communication Engineer, for the analysis of signal processing blocks in Openairinterface (OAI) software stack.**

1 Introduction to Matrix in C

Problem 1. Printing a matrix

```
#include<stdio.h>
int main()
{
// Decularation and Initialization
float A[3][2] = {{1.0, 1.2},
   {8,6.3},{4.5,0}};

// Printing matrix A
printf("A = \n");
for (int i=0; i<3; i++)
{
 for(int j=0; j<2; j++)
 {
  printf("%f \t", A[i][j]);
 }
 printf("\n");
}
return 0;
}
```

Problem 2. Transpose of a matrix

```
/*
Description : Printing the
   Transpose of a given matrix
Programmer : K. Prasanna Kumar
Last Modified Date : 27th May 2021
*/


#include<stdio.h>

// Decularation a function for
   printing a matrix
void print_matrix(int R, int C,
   int mat[R][C]);

void main()
{
// Runtime Initialization of rows
   and columns
int r, c;
printf("Number of rows r : ");
scanf("%d", &r);
printf("Number of column c : ");
scanf("%d", &c);

// Decularation of Matrix
int A[r][c], B[c][r];

// Runtime Initialization of
   Matrix A
for(int i = 0; i<r; i++)
{
 for(int j = 0; j<c; j++)
 {
  printf("Enter the value of A(%d
     %d) = ", i, j);
  scanf("%d", &A[i][j]);
 }
}

// Printing Matrix A
printf("A = \n");
```

Author is from 5G Testbed Project, IISc Bangalore, Email: kprasannakumar.iith@gmail.com

```
print_matrix(r, c, A);

// Transpose operation
for (int i = 0; i<r; i++)
{
 for(int j = 0; j<c; j++)
 {
  B[j][i] = A[i][j];
 }
}

// Printing Matrix A transpose
printf("\n\nA_transpose = \n");
print_matrix(c, r, B);
}

// Definination of print_matrix
   function
void print_matrix(int R, int C,
   int mat[R][C])
{
 for (int i = 0; i<R; i++)
 {
  for(int j = 0; j<C; j++)
  {
   printf("%d\t", mat[i][j]);
  }
  printf("\n");
 }
}
```

Problem 3. Equality of matrix : Two matrix $A = [a_{ij}]$ & $B = [b_{ij}]$ of the order nxm are said to be equal, if

$$[a_{ij}] = [b_{ij}]$$

```
/*
Description : Checking two matrix
   give are equal or not
Programmer : K. Prasanna Kumar
Last Modified Date : 27th May 2021
*/

#include<stdio.h>
// Defining function for runtime
   initilization of matrix
void scan_matrix(int R, int C, int
    mat[R][C])
{
 for(int i =0; i<R; i++)
 {
  for(int j =0; j<C; j++)
  {
   printf("Enter the value of(%d,
      %d) : ", i, j);
   scanf("%d", &mat[i][j]);
  }
 }
}

void main()
{
// Deculration and runtime
   initilzation of row and columns
int r,c;
printf("Number of rows r : ");
scanf("%d", &r);
printf("Numberr of columns c : ");
scanf("%d", &c);

// Deculration of matrixs
int A[r][c], B[r][c];

// Runtime Initilization of matrix
printf("\nEnter the values of
   matrix A = \n");
scan_matrix(r, c, A);
printf("\nEnter the values of
   matrix B = \n");
scan_matrix(r, c, B);

// Comparision
for (int i = 0; i<r; i++)
{
 for(int j =0; j<c; j++)
 {
  if (A[i][j] != B[i][j])
  {
   printf("\n --- Matrix A and B
      are not equal --- \n\n");
   return;
  }
 }
}
printf("\n ----- Matrix A = Matrix
   B ----- \n");
}
```

Problem 4. Header file for input & output opera-

tions of matrix during run-time

```
/*
Description : Hearder file for
   scan or print of a matrix
Programmer : K. Prasanna Kumar
Last Modified Date : 25th May 2021

Note: Save the header file with
   the name matrix_IO.h
*/

#include<stdio.h>

// User defined funtion for
   runtime initializtion of Matrix
void scan_matrix(int R, int C, int
    mat[R][C])
{
 for (int i =0; i<R; i++ )
 {
  for(int j=0; j<C; j++)
  {
   printf("Enter the value of(%d,
      %d) : ", i, j);
   scanf("%d", &mat[i][j]);
  }
 }
}

// User defined function to print
   a matrix
void print_matrix(int R, int C,
   int mat[R][C])
{
 for(int i=0; i<R; i++)
 {
  for(int j=0; j<C; j++)
  {
   printf("%d \t", mat[i][j]);
  }
  printf("\n");
 }
}
```

2 Types of Matrix

Problem 5. Null Matrix or Zero Matrix

```
/*
Description : Zero Matrix or
   Additve Identity Matrix
Programmer : K. Prasanna Kumar
Last Modified Date : 24th May 2020
*/

#include<stdio.h>
void main()
{
int r,c;
printf("Number of Rows : ");
scanf("%d", &r);
printf("Number of Columns : ");
scanf("%d", &c);

// Decularion of Zero Matrix
int O[r][c] ;

printf("Null Matrix or Zero Matrix
   \n");
printf("O = \n");
 for (int i = 0; i<r; i++)
 {
  for(int j = 0; j<c; j++)
  {
   O[i][j] = 0;
    printf("%d \t", O[i][j]);
  }
 printf("\n");
 }

}
```

Problem 6. Identity Matrix

```
/*
Description : Identity Matrtix is
   a multiplicative Identity of a
   matrix
Programmer : K. Prasanna Kumar
Last Modified Date : 24th April
   2020
*/

#include<stdio.h>
int main()
{
int n;
printf("Enter the Order of the
   Identity Matrix : ");
```

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


int I[n][n];

printf("Identity Matrix : I =\n"
   );
for(int i = 0; i<n; i++)
{
 for(int j = 0; j<n; j++)
 {
  if(i == j)
    I[i][j] = 1;
  else
    I[i][j] = 0;

   printf("%d\t", I[i][j]);
 }
 printf("\n");
}

return 0;
}
```

3 Matrix Algebra

Problem 7. Matrix Addition

```
/*
Description : Adding two matrix
   initilized at runtime
Programmer : K. Prasanna Kumar
Last Modified Date : 21st May 2020
*/

#include<stdio.h>
void main()
{
// Deculration
int r1,r2,c1,c2;
printf("Enter No. of rows of
   Matrix A: r1 =");
scanf("%d", &r1);
printf("Enter No. of colums of
   Matrix A: c1 =");
scanf("%d", &c1);

printf("Enter No. of rows of
   Matrix B: r2 =");
scanf("%d", &r2);
printf("Enter No. of colums of
   Matrix B: c2 =");
scanf("%d", &c2);

if(r1==r2 && c1==c2)
{

 float A[r1][c1];
 printf("-- Initilization of
    Matrix A ---\n");
 for(int i=0; i<r1; i++)
 {
  for(int j=0; j<c1; j++)
  {
   printf("Enter the value of(%d,
      %d) :", i, j);
   scanf("%f", &A[i][j]);
  }
 }

 float B[r2][c2];
 printf("-- Initilization of
    Matrix B ---\n");
 for(int i=0; i<r2; i++)
 {
  for(int j=0; j<c2; j++)
  {
   printf("Enter the value of(%d,
      %d) :", i, j);
   scanf("%f", &B[i][j]);
  }
 }

// Printing Matrix A and B
 printf("A =\n");
 for(int i=0; i<r1; i++)
 {
  for(int j=0; j<c1; j++)
  {
    printf("%f \t", A[i][j]);
  }
 printf("\n");
 }

 printf("B =\n");
 for(int i=0; i<r2; i++)
 {
  for(int j=0; j<c2; j++)
  {
   printf("%f \t", B[i][j]);
```

```
  }
 printf("\n");
 }

// Summation of Matrix A and B
 printf("\n -----Summation of
    Matrix A and B ----\n");
 float C[r1][c1];
 printf("C = \n");
 for(int i=0; i<r2; i++)
 {
  for(int j=0; j<c2; j++)
  {
   C[i][j] = A[i][j] + B[i][j];
    printf("%f \t", C[i][j]);
  }
 printf("\n");
 }

}
else
printf("\n\n Matrix Adition is not
   possible due to invalid input \
   n");

}
```

Problem 8. Matrix Difference

```
/*
Description : Forming a matrix
   with the differnce of two Matrix
Programmer : K. Prasanna Kumar
Last Modified Date : 21st May 2021
*/

#include<stdio.h>
// Deculration or Prototypes
void print_matrix(int R, int C,
   int mat[R][C]);

int main()
{
// Runtime Initializtion of matrix
 int r1, c1, r2, c2;
 printf("No. of rows of Matrix A :
    r1 = ");
 scanf("%d", &r1);
 printf("No. of colums of Matrix A
    : c1 = ");
 scanf("%d", &c1);
 printf("No. of rows of Matrix B :
    r2 = ");
 scanf("%d", &r2);
 printf("No. of colums of Matrix B
    : c2 = ");
 scanf("%d", &c2);

if (r1 == r2 && c1 == c2)
{
// Deculration of Marix
int A[r1][c1], B[r2][c2], C[r2][c2
   ];

// Initialization of Matrix
 for (int i =0; i<r1; i++ )
 {
  for(int j=0; j<c1; j++)
  {
   printf("Enter the value of A(%d
      ,%d) = ", i, j);
   scanf("%d", &A[i][j]);
  }
 }

 for (int i =0; i<r1; i++ )
 {
  for(int j=0; j<c1; j++)
  {
   printf("Enter the value of B(%d
      ,%d) = ", i, j);
   scanf("%d", &B[i][j]);
  }
 }

// Computation
 for (int i =0; i<r1; i++ )
 {
  for(int j=0; j<c1; j++)
  {
   C[i][j] = A[i][j] - B[i][j];
  }
 }

// Output
 printf("A = \n");
 print_matrix(r1, c1, A);

 printf("B = \n");
 print_matrix(r1, c1, B);
```

```
 printf("C = \n");
 print_matrix(r1, c1, C);
}
else
printf("\n --- Invalid Input ----- \n");

}

// Defining Function
void print_matrix(int R, int C,
   int mat[R][C])
{
 for(int i=0; i<R; i++)
 {
  for(int j=0; j<C; j++)
  {
   printf("%d \t", mat[i][j]);
  }
  printf("\n");
 }
}
```

Problem 9. Matrix Multiplication

```
/*
Description : Product of two
   matrix
Programmer : K. Prasanna Kumar
Last Modified Date : 21st May 2021
*/

#include<stdio.h>
// Deculration or Prototypes
void print_matrix(int R, int C,
   int mat[R][C]);
void scan_matrix(int R, int C, int
    mat[R][C]);

int main()
{
// Runtime Initializtion of matrix
 int r1, c1, r2, c2;
 printf("No. of rows of Matrix A :
    r1 = ");
 scanf("%d", &r1);
 printf("No. of colums of Matrix A
    : c1 = ");
 scanf("%d", &c1);
 printf("No. of rows of Matrix B :
    r2 = ");
 scanf("%d", &r2);
 printf("No. of colums of Matrix B
    : c2 = ");
 scanf("%d", &c2);

if (c1 == r2)
{
// Deculration of Marix
int A[r1][c1];
int B[r2][c2];
int C[r1][c2];
// Initialization of Matrix
printf("Enter the value of Matrix
   A : (i, j) \n");
scan_matrix(r1, c1, A);
printf("Enter the values of Matrix
   B : (i, j) \n");
scan_matrix(r2, c2, B);

// Computation
 int P;
 for (int i = 0; i<r1; i++)
 {
 for(int k = 0; k<c2; k++)
 {
 // Note C1 is equal to r2
 for(int j = 0; j<r2; j++)
 {
  P = A[i][j]*B[j][k];
  if (j == 0)
  C[i][k] = P;
  else
  C[i][k] = C[i][k] + P;
 }
 }
 }

// Output
 printf("A = \n");
 print_matrix(r1, c1, A);

 printf("B = \n");
 print_matrix(r2, c2, B);

 printf("\n\nProduct of A & B is :
    ");
 printf("C = \n");
 print_matrix(r1, c2, C);
```

```
}
else
printf("\n --- Invalid Input ----- \n");
}

// Function Definition :--
// User defined funtion to runtime initialize Matrix
void scan_matrix(int R, int C, int mat[R][C])
{
 for (int i =0; i<R; i++ )
 {
  for(int j=0; j<C; j++)
  {
   printf("Enter the value of(%d, %d) : ", i, j);
   scanf("%d", &mat[i][j]);
  }
 }
}

// User defined function to print a matrix
void print_matrix(int R, int C, int mat[R][C])
{
 for(int i=0; i<R; i++)
 {
  for(int j=0; j<C; j++)
  {
   printf("%d \t", mat[i][j]);
  }
  printf("\n");
 }
}
```

```
void main()
{
// Decularation and runtime initializtion of rows & columns
int r, c;
printf("Number of Rows : ");
scanf("%d", &r);
printf("Number of Columns : ");
scanf("%d", &c);

// Deculration matrix A and its addative inverse B
int A[r][c], B[r][c];

// Initalization of A
printf("Enter the value of matrix A  :-- \n");
scan_matrix(r, c, A);
printf("\nA = \n");
print_matrix(r, c, A);

// Compute & print addative inverse
printf("Additive inverse of A : B = \n");
for (int i = 0; i<r; i++)
 {
  for(int j= 0; j<c; j++)
  {
   B[i][j] = -A[i][j];
  // printf("%d \t", -A[i][j]);
  }
 // printf("\n");
 }

print_matrix(r, c, B);
}
```

Problem 10. Additive inverse of a matrix

```
/*
Description : Forming a matrix as an additive inverse of a given matrix
Programmer : K. Prasanna Kumar
Last Modified Date : 27th May 2021
*/

#include<stdio.h>
#include"matrix_IO.h"
```

Problem 11. Summation of row elements to form a column matrix

```
/*
Description : Froming a Colum martrix by the summation of rows elements in the given matrix
Programmer : K. Prasanna Kumar
Last Modified Date : 27th May 2021
*/
```

```
#include<stdio.h>
#include"matrix_IO.h"

/*
Defineing a pointer function which
    return a row matrix with
sum of the elements in the given
   matrix
*/

int *sum_row(int R, int C, int mat
   [R][C])
{
 static int arr[20];
 for(int i = 0; i<R; i++)
 {
 for(int j =0; j<C; j++)
  {
   if (j == 0)
     arr[i] = mat[i][j];
   else
     arr[i] = arr[i] + mat[i][j];
  }
 }
 return arr;
}

void main()
{
 // Decularation
 int r, c;
 printf("Enter the number of rows
    r : ");
 scanf("%d", &r);
 printf("Enter the number of
    colums c : ");
 scanf("%d", &c);
 int A[r][c];
 int *A_column;

 // Initialization
 scan_matrix(r, c, A);
 printf("A = \n");
 print_matrix(r, c, A);

 // Computation
 A_column = sum_row(r, c, A);

 // Display
 printf("A_column  = \n");
 for (int i = 0; i<r; i++)
 {
 printf(" %d \n", *(A_column+i));
 }
}
```

Problem 12. Summation of column elements to form a row matrix

```
/*
Description : Froming a row
   martrix by the summation of
   column elements in the given
   matrix
Programmer : K. Prasanna Kumar
Last Modified Date : 27th May 2021
*/


#include<stdio.h>
#include"matrix_IO.h"

/*
Defineing a pointer function which
    return a row matrix with
sum of the elements in the given
   matrix
*/

int *sum_column(int R, int C, int
   mat[R][C])
{
 static int arr[20];
 for(int j = 0; j<C; j++)
 {
 for(int i=0; i<R; i++)
  {
   if (i == 0)
     arr[j] = mat[i][j];
   else
     arr[j] = arr[j] + mat[i][j];
  }
 }
 return arr;
}

void main()
{
 // Decularation
 int r, c;
```

```
  printf("Enter the number of rows
     r : ");
  scanf("%d", &r);
  printf("Enter the number of
     colums c : ");
  scanf("%d", &c);
  int A[r][c];
  int *A_row;

  // Initialization
  scan_matrix(r, c, A);
  printf("A = \n");
  print_matrix(r, c, A);

  // Computation
  A_row = sum_column(r, c, A);

  // Display
  printf("A_row  = \n");
  for (int i = 0; i<c; i++)
  {
  printf(" %d \t", *(A_row+i));
  }
  printf("\n");
}
```

References

[1] Gaurav Bhorkar, "Programming Fundamentals in C (NU / MSBTE / Beginner Level)", youtube video playlist https://www.youtube.com/playlist?list=PL90FACD026D4959BE. Email id :gaurav.bhorkar@gmail.com