| **Name** | Jhaveri Varun Nimitt |
| --- | --- |
| **UID no.** | 2023800042 |
| **Experiment No.** | 5 |

| **AIM:** | Demonstrate the use of two-dimensional arrays to solve a given problem. |
| --- | --- |
| **Program 1** | |
| **PROBLEM STATEMENT :** | Write a program to perform Matrix Addition, Subtraction. |
| **ALGORITHM:** | 1. Start 2. Define a void function read with the arguments : dimensions m and n of a 2D array ,array. 3. Declare variables i and j. 4. Iterate i from 0 to m and iterate j from 0 to n,incrementing i and j by 1 each iteration:   - - Read the value of arr[i][j] using scanf.   1. Define the function print with the arguments : dimensions m and n of a 2D array and the array itself.   - a. Declare variables i and j.  - b. Iterate i from 0 to m and iterate j from 0 to n,incrementing i and j by 1 each iteration:  - - Print the value of arr[i][j] \t  - print a new line   1. Define a void function addMatrice with the arguments : dimensions m and n of two arrays and their elements, result array. 2. Declare variables i and j. 3. Iterate i from 0 to and iterate j from 0 to n,incrementing i and j by 1 each iteration:   - - Calculate the sum of corresponding elements of arr1 and arr2 and store it in res[i][j].   1. Print "result: ". 2. Call the print function with parameters m, n, and res. 3. Define a void function subtractMatrix with the arguments : dimensions m and n of two arrays and their elements, result array. 4. Declare variables i and j. 5. Iterate i from 0 to m and iterate j from 0 to n,incrementing i and j by 1 each iteration:   - - Calculate the difference of corresponding elements of arr1 and arr2 and store it in res[i][j].   1. Print "result: ". 2. Call the print function with parameters m, n, and res. 3. Define the main function. 4. Declare integer variables m, n, and choice. 5. Print "rows: " and read the value of m using scanf. 6. Print "columns: " and read the value of n using scanf. 7. Declare 2D arrays arr1, arr2, and res, each with dimensions m x n. 8. Print "Enter elements first:" and call the read function with parameters m, n, and arr1. 9. Print "Enter elements second:" and call the read function with parameters m, n, and arr2. 10. Print "First matrix is:" and call the print function with parameters m, n, and arr1. 11. Print "Second matrix is:" and call the print function with parameters m, n, and arr2. 12. Print the menu options for addition and subtraction. 13. Read the value of choice using scanf. 14. Using a switch-case statement, perform the chosen operation based on the user's choice.  * Case 1 :      1. Print “addition of matrix is:” 2. call the addMatrice function with parameters m, n,arr1,arr2 and res.  * Case 2:      1. Print “subtraction of matrix is:” 2. call the subtractMatrix function with parameters m, n,arr1,arr2 and res.  * Default case :  1. Print "Invalid choice".      1. End. |
| **PROGRAM:** | #include <stdio.h>  void read (int m,int n, int arr[m][n]) {  int i,j;  for (i=0;i<m;i++)  {  for (j=0;j<n;j++)  {  printf("Enter the value of arr[%d][%d]: ",i,j);  scanf("%d",&arr[i][j]);  }  } } void print(int m,int n, int arr[m][n]) {  int i,j;  for (i=0;i<m;i++)  {  for (j=0;j<n;j++)  {  printf("%d\t",arr[i][j]);  }  printf("\n");  } }  void addMatrice(int m, int n ,int arr1[m][n],int arr2[m][n],int res[m][n]) {  int i,j;  for (i=0;i<m;i++)  {  for (j=0;j<n;j++)  {  res[i][j]=arr1[i][j]+arr2[i][j];  }  }  printf("result: \n");  print(m,n,res); }  void subtractMatrix(int m, int n ,int arr1[m][n],int arr2[m][n],int res[m][n]) {  int i,j;  for (i=0;i<m;i++)  {  for (j=0;j<n;j++)  {  res[i][j]=arr1[i][j]-arr2[i][j];  }  }  printf("result: \n");  print(m,n,res); }  int main() {  int m,n;  printf(" rows: ");  scanf("%d",&m);  printf(" columns: ");  scanf("%d",&n);  int arr1[m][n],arr2[m][n],res[m][n];  printf("Enter elements first : \n");  read(m,n,arr1);  printf("Enter elements second : \n");  read(m,n,arr2);  printf("first matrix is: \n");  print(m,n,arr1);  printf(" second matrix is: \n");  print(m,n,arr2);  printf("1. Add\n2. Subtract\n");  printf("Enter your choice: ");  int choice;  scanf("%d",&choice);  switch(choice)  {  case 1: printf("addition of matrix is: \n");  addMatrice(m,n,arr1,arr2,res);  break;  case 2:printf("subtraction of matrix is: \n");  subtractMatrix(m,n,arr1,arr2,res);  break;  default: printf("Invalid choice");  }    return 0; } |
| **RESULT:**  **Addition :**    **Subtraction:** | |
| **Program 2** | |
| **PROBLEM STATEMENT :** | Write a program to perform Matrix Multiplication |
| **ALGORITHM:** | 1. Start 2. Define a void function read with the arguments : dimensions m and n of a 2D array , array. 3. Declare variables i and j. 4. Iterate i from 0 to m and iterate j from 0 to n,incrementing i and j by 1 each iteration:   - - Read the value of arr[i][j] using scanf.   1. Define the function print with the arguments : dimensions m and n of a 2D array ,array. 2. Declare variables i and j. 3. Iterate i from 0 to m and iterate j from 0 to n,incrementing i and j by 1 each iteration:   - - Print the value of arr[i][j] followed by a tab.  - print a newline.   1. Define a void function multiplyMatrices with the arguments : dimensions m, n, p, and q of two 2D arrays and their elements, result array. 2. Iterate i from 0 to m,incrementing i by 1 each iteration: 3. Iterate j from 0 to n ,incrementing j by 1 each iteration:   - - result[i][j] =0.  - - Iterate k from 0 to n,incrementing k by1 each iteration:  - - Update result[i][j] by adding the product of A[i][k] and B[k][j].   1. Print "Resultant matrix:". 2. Call the print function with parameters m, n, and result. 3. Define the main function. 4. Declare integer variables m, n, p, and q. 5. Print "row column for 1st:" and read the values of m and n using scanf. 6. Print "row column for 2:" and read the values of p and q using scanf. 7. If the number of columns in the first matrix is not equal to the number of rows in the second matrix:   - - Print "error"   1. Declare 2D arrays A, B, and res, each with dimensions m x n, p x q, and n x p, respectively. 2. Print "Enter elements first:" and call the read function with parameters m, n, and A. 3. Print "Enter elements second:" and call the read function with parameters p, q, and B. 4. Print "First matrix is:" and call the print function with parameters m, n, and A. 5. Print "Second matrix is:" and call the print function with parameters p, q, and B. 6. Call the multiplyMatrices function with parameters m, n, p, q, A, B, and res. 7. End. |
| **PROGRAM:** | #include <stdio.h>   void read (int m,int n, int arr[m][n]) {  int i,j;  for (i=0;i<m;i++)  {  for (j=0;j<n;j++)  {  printf("Enter the value of arr[%d][%d]: ",i,j);  scanf("%d",&arr[i][j]);  }  } } void print(int m,int n, int arr[m][n]) {  int i,j;  for (i=0;i<m;i++)  {  for (j=0;j<n;j++)  {  printf("%d\t",arr[i][j]);  }  printf("\n");  } }  void multiplyMatrices( int m, int n, int p,int q,int A[m][n], int B[p][q], int result[n][p])  {  for (int i = 0; i < m; i++)   {  for (int j = 0; j < n; j++)   {  result[i][j] = 0;  for (int k = 0; k < n; k++)   {    result[i][j] += A[i][k] \* B[k][j];  }  }  }  printf("Resultant matrix:\n");  print(m,n,result); }  int main()  {  int m, n, p, q;   printf("row column for 1st: ");  scanf("%d %d", &m, &n);   printf("row column for 2: ");  scanf("%d %d", &p, &q);  if (n != p)   {  printf("error\n");  return 1;  }  int A[m][n], B[p][q], res[n][p];   printf("Enter elements first : \n");  read(m,n,A);  printf("Enter elements second : \n");  read(m,n,B);    printf("first matrix is: \n");  print(m,n,A);  printf(" second matrix is: \n");  print(m,n,B);  multiplyMatrices(m,n,p,q,A,B,res);   return 0; } |
| **RESULT:** | |
| **Program 3** | |
| **PROBLEM STATEMENT :** | Write a program to perform Transpose of Matrix |
| **ALGORITHM:** | 1. Start 2. Define a void function read with arguments : dimensions m and n of 2D array , array. 3. Declare i and j. 4. Iterate i from 0 to m and iterate j from 0 to n,incrementing i and j by 1 each iteration:   - -Read the value of arr[i][j] using scanf.   1. Define a void function transpose with arguments : dimensions m and n of a matrix, original matrix, result matrix. 2. Iterate i from 0 to m,incrementing i by 1 each iteration:   - - Iterate j from 0 to n,incrementing j and j by 1 each iteration:  - - Assign the value of matrix[i][j] to result[j][i].   1. 4. Define a void function print with arguments : the dimensions m and n of a 2D array ,array. 2. Declare i and j. 3. Iterate i from 0 to m and iterate j from 0 to n,,incrementing i and j by 1 each iteration:   - - Print the value of arr[i][j] \t  - print a newline.   1. Define the main function. 2. Declare integer m and n. 3. Read the values of m and n using scanf. 4. Declare 2D arrays ogmatrix and result with dimensions m x n and n x m respectively. 5. Call the read function with parameters m, n, and ogmatrix. 6. Print "Original matrix:". 7. Call the print function with parameters m, n, and ogmatrix. 8. Call the transpose function with parameters m, n, ogmatrix, and result. 9. Print "Transposed matrix:". 10. Call the print function with parameters n, m, and result. 11. End. |
| **PROGRAM:** | #include <stdio.h>  void read (int m,int n, int arr[m][n]) {  int i,j;  for (i=0;i<m;i++)  {  for (j=0;j<n;j++)  {  printf("Enter the value of arr[%d][%d]: ",i,j);  scanf("%d",&arr[i][j]);  }  } }  void transpose(int m,int n,int matrix[m][n], int result[n][m])  {  for (int i = 0; i < m; i++)   {  for (int j = 0; j < n; j++)   {  result[j][i] = matrix[i][j];  }  } }  void print(int m,int n, int arr[m][n]) {  int i,j;  for (i=0;i<m;i++)  {  for (j=0;j<n;j++)  {  printf("%d\t",arr[i][j]);  }  printf("\n");  } }  int main()  {  int m, n;  printf("row column for matrix: ");  scanf("%d %d", &m, &n);  printf("Enter elements first : \n");  int ogmatrix[m][n] ;  int result[n][m];  read(m,n,ogmatrix);  printf("Original matrix:\n");  print(m,n,ogmatrix);  transpose(m,n,ogmatrix, result);  printf("\nTransposed matrix:\n");  print(n,m,result);  return 0; } |
| **RESULT:** | |
| **CONCLUSION:** | **I have understood how to use two-dimensional arrays to solve a given problem.** |