Name	**PRATIK PUJARI**
UID no.	**2020300054**
Experiment No.	**5**

AIM:	
Program 1	
PROBLEM STATEMENT :	**Write a program to perform matrix multiplication. Dimensions of matrices will be decided by user.**

PROGRAM:

```
ALGORITHM:
(Initialize the variables r1,c1,r2,c2)
void main()
STEP 1: START.
STEP 2: Printf the dimension of the first martix and input them into r1 and c1.
STEP 3: Printf the dimension of the second matrix and input then into r2 and c2.
STEP 4: If c1 not equal to r2 then printf The matrix multiplication of the array is not possible else go to step 5.
STEP 5: Initialize the array arr1[r1][c1], arr2[r2][c2] and res[r1][c2].
STEP 6: Call the predefined function input (arr1,arr2) and mult(arr1,arr2)

void input (int arr[r1][c1] ,int arr2 [r2][c2])
STEP 1: START.
STEP 2: Printf "Enter the elements of the first array"
STEP 3: Initialize the loop counter i and j.
STEP 4: For I equal to zero and less than r1 ,Repeat the steps 4.1 and 4.2 else if the condition is false go to Step 5.
STEP 4.1: For j equal to zero and less than c1 ,Repeat the steps 4.1.1,4.1.2 or else if the  condition is false go to step 4.2.
STEP 4.1.1: Input the element of the array arr1[i][j] from the user.
STEP 4.1.2: Increment the loop counter by one(j).
STEP 4.2: Increment the loop counter by one (i).
STEP 5: Printf "Enter the elements of the second array"
STEP 6: For I equal to zero and less than r2, Repeat the steps 6.1 and 6.2 else if the condition is false go to Step 7.
STEP 6.1: For j equal to zero and less than c2, Repeat the steps 6.1.1,6.1.2 or else if the condition is false go to step 6.2.
STEP 6.1.1: Input the element of the array arr2[i][j] from the user.
STEP 6.1.2: Increment the loop counter by one(j).
STEP 6.2: Increment the loop counter by one (i).
STEP 7: END.
```

void mult(int arr1[r1][c1], int arr2[r2][c2])

```
STEP 1: START.
STEP 2: Initialize the array res to res[r1][c2] and the loop counters i , j and k.
STEP 3: For I equal to zero and less than r1 ,Repeat the steps 3.1 and 3.2 else if the condition is false go to Step 4.
    STEP 3.1: For j equal to zero and less than c2 ,Repeat the steps 3.1.1,3.1.2 or else if the  condition is false go to step 3.2.
        STEP 3.1.1: Do res[i][j] equal to zero.
        STEP 3.1.2: Increment the loop counter by one(j).
    STEP 3.2: Increment the loop counter by one (i).
STEP 4: For I equal to zero and less than r1 ,Repeat the steps 4.1 and 4.2 else if the condition is false go to Step 5.
    STEP 4.1: For j equal to zero and less than c2 ,Repeat the steps 4.1.1,4.1.2 or else if the  condition is false go to step 4.2.
        STEP 4.1.1: For k=0 and less than c1, Repeat the steps 4.1.1.1,4.1.1.2 or else if the condition fails go to step 4.1.2.
            STEP 4.1.1.1: Do res[i][j]=res[i][j]+arr1[i][k].
            STEP 4.1.1.2: Increment the loop counter by one (j).
        STEP 4.1.2: Increment the loop counter by one(j).
    STEP 4.2: Increment the loop counter by one (i).
STEP 5: Call the predefined function display(res).
STEP 6: END.
```

void display (int res[r1][c2])

```
STEP 1: START.
STEP 2: Initialize the loop counter i and j to zero.
STEP 3: For I equal to zero and less than r1 ,Repeat the steps 3.1 and 3.2 else if the condition is false go to Step 4.
    STEP 3.1: For j equal to zero and less than c2 ,Repeat the steps 3.1.1,3.1.2 or else if the  condition is false go to step 3.2.
        STEP 3.1.1: Printf %d ,res[i][j].
        STEP 3.1.2: Increment the loop counter by one(j).
    STEP 3.2: Increment the loop counter by one (i).
STEP 4: END.
```

PROGRAM:

```
#include<stdio.h>
int r1,c1,r2,c2;
void input(int arr1[r1][c1],int arr2[r2][c2]);
void mult(int arr1[r1][c1],int arr2[r2][c2]);
void display(int res[r1][c2]);
void main()
{
   printf("Enter the dimensions of the first array(row x column):");
   scanf("%dx%d",&r1,&c1);
   printf("Enter the dimensions of the second array(row x column):");
   scanf("%dx%d",&r2,&c2);
   if(c1!=r2)
         {
                  printf("The matrix multiplication of the arrays is not
possible");
   }
   else
   {
      int arr1[r1][c1], arr2[r2][c2];
      int res[r1][c2];
      input(arr1,arr2);
      mult(arr1,arr2);
   }
}
void input(int arr1[r1][c1],int arr2[r2][c2])
{
   printf("Enter the elements of the first array:");
   for(int i=0;i<r1;i++)
   {
      for(int j=0;j<c1;j++)
      {
         scanf("%d",&arr1[i][j]);
      }
   }
         printf("Enter the elements of the second array:");
         for(int i=0;i<r2;i++)
   {
      for(int j=0;j<c2;j++)
      {
         scanf("%d",&arr2[i][j]);
      }
   }
}
void mult(int arr1[r1][c1],int arr2[r2][c2])
{
   int res[r1][c2];
   for(int i=0;i<r1;i++)
   {
        for(int j=0;j<c2;j++)
        {
            res[i][j]=0;
        }
    }
    for(int i=0;i<r1;i++)
    {
        for(int j=0;j<c2;j++)
        {
            for(int k=0;k<c1;k++)
            {
                res[i][j]=res[i][j]+arr1[i][k]*arr2[k][j];
            }
        }
    }
    display(res);
}
void display(int res[r1][c2])
{
    printf("The multiplied matrix with %dx%d dimensions is :\n",r1,c2);
    for(int i=0;i<r1;i++)
    {
        for(int j=0;j<c2;j++)
        {
            printf("%d ",res[i][j]);
        }
        printf("\n");
    }
}
```

RESULT: Matrices are multiplied using iteration logic.

INPUT:

2x2 and 2x2.
{1 3}   {4 6}
{2 4}   {7 3}

OUTPUT:

```
Enter the dimensions of the first 1matrix (row x column):2x2
Enter the dimensions of the second matrix(row x column):2x2
Enter the elements of the first array:1
3
2
4
Enter the elements of the second array:4
6
7
3
```

	The multiplied matrix with 2x2 dimensions is : 25 15 36 24

Program 2	
PROBLEM STATEMENT :	Write a program which reads the current year followed by N followed by a list of N employee numbers and their current ages. Produce a list showing the years in which the employees retire (become 65 years old). If more than one employee retires in a given year then include them all under the same heading.
PROGRAM:	ALGORITHM: (Initialize the variable nemp) void main() STEP 1: START. STEP 2: Initialize the array emp[nemp][3] and nyear. STEP 3: Printf “Enter the number of employees and the current year:” STEP 4: Input the user input into the variable nemp and nyear. STEP 5: Call the predefined function employee(emp,nyear). void employee(int emp[nemp][3], int year) STEP 1: START. STEP 2: Initialize the loop counter i and j to zero. STEP 3:For i=0 and less than nemp, Repeat the steps 3.1,3.2,3.3,3.4,3.5and 3.6 or else if the condition fails go to step 4. STEP 4.1: Printf “Enter the Employee ID of EMP %d:” i+1. STEP 4.2: Input the user and store it to emp[i][j] STEP 4.3: Increment j by one. STEP 4.4: Printf “Enter the Employee age%d” i+1. STEP 4.5: Input from the user and store it to emp[i][j]. STEP 4.6: Increment the loop counter by one(i). STEP 5: Call the predefined function sort(emp,year) STEP 6: END

void sort(int emp[nemp][3],int year)

```
STEP 1: START.
STEP 2: Initialize the variable temp and loop counters i and j
STEP 3: For i=0 and less than nemp ,Repeat the steps 3.1 and 3.2 or else if the condition fails go to step 4.
  STEP 3.1: Do emp[i][2]=0.
  STEP 3.2: Increment the loop counter (i) by one.
STEP 4: For i equal to zero and less than nemp ,Repeat the steps 4.1,4.2,4.3 and 4.4 or else if the condition fails go to step 5.
  STEP 4.1: Initialize the variable j to zero.
  STEP 4.2: Do temp=year-emp[i][1]+65.
  STEP 4.3: Do emp[i][2]=temp.
  STEP 4.4: Increment the loop counter(i) by one.
STEP 5: For i=0 and less than nemp , Repeat the steps 5.1 and 5.2 or else if the condition fails go to step 6.
  STEP 5.1: For j=i and less than nemp ,Repeat the steps 5.1.1 or else if the condition fails go to step 5.2
    STEP 5.1.1: If emp[i][2]>emp[j][2] and emp[i][2] not equal to 0 and emp[j][2] not equal to 0 go to Step 5.1.1.1 or else 5.1.2
      STEP 5.1.1.1: Initialize the variable swap
      STEP 5.1.1.2: For k=0 and less than 3 ,Perform the below three steps or if the condition fails go to step 5.1.2.
        STEP #1: swap=emp[i][k].
        STEP #2: emp[i][k]=emp[j][k].
        STEP #3: emp[j][k]=swap.
    STEP 5.1.2: Increment the loop counter j by one.
  STEP 5.2: Increment the loop counter i by one
STEP 6: Call the predefined function result(emp).
STEP 7: END.
```

void result(int emp[nemp][3])

```
STEP 1: START.
STEP 2: Printf “List of people retiring is:”.
STEP 3: Printf “Id\tAGE\tRetires in \n”.
STEP 4: Initialize the loop counter i and j to zero.
STEP 5: For i=0 and less than nemp ,Repeat the steps 5.1,5.2 and 5.3 or else if the condition fails go to step 6.
  STEP 5.1: For j=0 and less than 3, Repeat the steps 5.1.1 and 5.1.2 or else if the condition fails go to step 5.2
    STEP 5.1.1: If  emp[i][j] not equal to zero printf “%d \t” emp[i][j] or else if go to step  5.1.2
    STEP 5.1.2: Increment the loop counter j by one.
  STEP 5.2: Printf(“\n”)
  STEP 5.3: Increment the loop counter i by one.
STEP 6: END.
```

PROGRAM:

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

```
int nemp;
void employee(int emp[][nemp],int);
void sort(int emp[][nemp],int);
void result(int emp[][nemp]);
void main()
{
    int emp[3][nemp] ,nyear;
    printf("Enter the number of employees and current year:");
    scanf("%d %d",&nemp,&nyear);
    employee(emp,nyear);
}
void employee(int emp[][nemp], int year)
{
    for(int i=0;i<nemp;i++)
    {
        int j=0;
        printf("Enter the Employee Id of EMP%d :",i+1);
        scanf("%d",&emp[i][j]);
        j++;
        printf("Enter the Age of the EMP%d :",i+1);
        scanf("%d",&emp[i][j]);
    }
    sort(emp,year);
}
void sort(int emp[][nemp],int year)
{
    int temp;
    for (int i=0;i<nemp;i++)
    {
        emp[i][2]=0;
    }
    for(int i=0;i<nemp;i++)
    {
        int j=0;
        temp=year-emp[i][1]+65;
        emp[i][2]=temp;
    }
    for(int i=0;i<nemp;i++)
    {
        for(int j=i;j<nemp;j++)
        {
            if(emp[i][2]>emp[j][2] && emp[i][2]!=0 && emp[j][2]!=0)
            {
                int swap;
                swap=emp[i][2];
                emp[i][2]=emp[j][2];
                emp[j][2]=swap;
                swap = emp[i][1];
                emp[i][1] = emp[j][1];
                emp[j][1] = swap;
                swap = emp[i][0];
                emp[i][0] = emp[j][0];
                emp[j][0] = swap;
            }

        }
    }
    result(emp);
}
void result(int emp[][nemp])
{
    printf("\nList of people retiring:\n");
    printf("ID\tAGE\tRetires in \n");
    for (int i=0;i<nemp;i++)
    {
        for (int j=0;j<3;j++)
        {
            if(emp[i][j]!=0)
            {
                printf("%d \t", emp[i][j]);
            }
        }
        printf("\n");
    }
}
```

RESULT: The retirement ages of the employees are in ascending order.

INPUT:

```
3 2021
1 34
2 45
3 21
```

OUTPUT:

```
Enter the number of employees and current year:3 2021
Enter employee's ID and age:1 34
Enter employee's ID and age:2 45
Enter employee's ID and age:3 21

List of people retiring:
| EMPLOYEE ID | CURRENT AGE | RETIRES IN |
-----------------------------------------
|     2041    |    2     |    45    |
|     2052    |    1     |    34    |
|     2065    |    3     |    21    |
```