

```
#include<stdio.h>

#include<conio.h>

void sumofmatrix(int matrix[3][3],int matrix2[3][3],int result[3][3])
{
    int i;

    int j;

    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            result[i][j]= matrix[i][j] + matrix2[i][j];
        }
    }
}

void subtract(int matrix[3][3],int matrix2[3][3],int result[3][3])
{
    int i;

    int j;

    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            result[i][j]= matrix[i][j] - matrix2[i][j];
        }
    }
}
```

```
}  
  
void multiply(int matrix[3][3],int matrix2[3][3],int result[3][3])  
{  
    int i;  
    int j;  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            result[i][j]= matrix[i][j] * matrix2[i][j];  
        }  
    }  
}  
  
void transpose(int matrix[3][3],int matrix2[3][3],int result[3][3])  
{  
    int i;  
    int j;  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            result[i][j]= matrix[j][i];  
        }  
    }  
}
```

```
void display( int result[3][3])
{
    int i;
    int j;
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            printf("%d\t",result[i][j]);
        }
    }
}
```

```
int main()
{
    int i;
    int j;
    int choice;
    int matrix[3][3];
    int matrix2[3][3];
    int result[3][3];
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
```

```
        scanf("%d", &matrix[i][j]);  
    }  
}
```

```
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        printf("%d \t", matrix[i][j]);  
    }  
    printf("\n");  
}
```

```
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        scanf("%d", &matrix2[i][j]);  
    }  
}
```

```
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        printf("%d \t", matrix2[i][j]);  
    }  
}
```

```
    }  
    printf("\n");  
}  
printf("1.addition,2.subtraction,3.multiplication,4.transpose");  
printf("enter your choice");  
scanf("%d", &choice);  
switch(choice)  
{  
    case 1:sumofmatrix(matrix,matrix2,result);  
        display(result);  
        break;  
    case 2:subtract(matrix,matrix2,result);  
        display(result);  
        break;  
    case 3:multiply(matrix,matrix2,result);  
        display(result);  
        break;  
    case 4:transpose(matrix,matrix2,result);  
        display(result);  
        break;  
    default:  
        printf("invalid");  
}
```

