

```
#include<stdio.h>

int main(){
    int m, n, p, i, j, k, u, sum;
    scanf("%d %d", &m, &n);

    // Find the smaller of the dimensions
    // Careful: S will have only p diagonal entries
    p = (m < n)? m : n;
    int U[m][m], S[p], V[n][n];

    for(i = 0; i < p; i++)
        scanf("%d", &S[i]);

    for(i = 0; i < m; i++)
        for(j = 0; j < m; j++)
            scanf("%d", &U[i][j]);

    for(i = 0; i < n; i++)
        for(j = 0; j < n; j++)
            scanf("%d", &V[i][j]);

    // Find the (i,j) entry of the product matrix A
    // A = U * S * V'
    for(i = 0; i < m; i++){
        for(j = 0; j < n; j++){
            sum = 0;
            for(k = 0; k < n; k++) {
                // The matrix B = U * S is m x n in size
                // The index k column of matrix B is simply the
                // index k column of U multiplied by S_k if k < m
                // else the index k column of B is zero for k >= m
                if (k < m)
                    u = U[i][k] * S[k];
                else
                    u = 0;
                sum += (u * V[j][k]);
            }
            printf("%d", sum);
            if(j < n-1) printf(" "); // No trailing spaces
        }
        if(i < m-1) printf("\n"); // No trailing newlines
    }
    return 0;
}
```