

1.)Napisati program za množenje dviju matrica sa pohranom rezultata u treću matricu. Koliko promasaja L1 memorije očekujemo za množenje matrica A i B tipa **double** dimenzija A(8 x 8), B(8 x 16), C(8 x 16). Objasniti i izračunati.

Pretpostaviti da se matrice A,B i C inicijalno nalaze u L2 memoriji (početak svake matrice je poravnat a početkom linije), te da je L1 memorija dovoljno velika da se sve tri matrice mogu istovremeno nalaziti u njoj. Veličina linije u L1 i L2 je 64B.

Rješenje:

Pošto sve 3 matrice stanu u L1 i nalaze se u L2 onda ce se promasaji dogoditi samo kada prvi put učitavamo podatke u L1. Imamo double podatke, prva je 8x8, a druge dvije 8x16, to je znaci $64+128+128=320$ podataka, velicine 8B (double), znaci 2560B podataka, to podijelimo sa 64B koliko je jedna linija i dobijemo da cemo imati 40 redaka u PM i to ce biti 40 promasaja! Ostalo su sve pogotci.

```
#include <stdio.h>

int main()
{
    int m, n, p, q, c, d, k, sum = 0;
    int first[10][10], second[10][10], multiply[10][10];

    printf("Enter the number of rows and columns of first matrix\n");
    scanf("%d%d", &m, &n);
    printf("Enter the elements of first matrix\n");

    for (c = 0; c < m; c++)
        for (d = 0; d < n; d++)
            scanf("%d", &first[c][d]);

    printf("Enter the number of rows and columns of second matrix\n");
    scanf("%d%d", &p, &q);

    if (n != p)
        printf("Matrices with entered orders can't be multiplied with each other.\n");
    else
    {
        printf("Enter the elements of second matrix\n");

        for (c = 0; c < p; c++)
            for (d = 0; d < q; d++)
                scanf("%d", &second[c][d]);

        for (c = 0; c < m; c++) {
            for (d = 0; d < q; d++) {
                for (k = 0; k < p; k++) {
                    sum = sum + first[c][k]*second[k][d];
                }

                multiply[c][d] = sum;
                sum = 0;
            }
        }

        printf("Product of entered matrices:-\n");

        for (c = 0; c < m; c++) {
            for (d = 0; d < q; d++)
                printf("%d\t", multiply[c][d]);

            printf("\n");
        }
    }

    return 0;
}
```

char : 1 byte
short : 2 bytes

```
int    : 4 bytes
long   : 4 bytes
float  : 4 bytes
double: 8 bytes
```

Primjer 2.

Napisati program za transponiranje matrice. Potrebno je učitati dimenzije matrice i sve elemente matrice. Transponirana matrica je ona kojoj se zamijene reci i stupci.

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

#define MAX_RED  50
#define MAX_STUP 50

int main() {
    int i, j, m, n, pom, max;
    int mat[MAX_RED][MAX_STUP];

    // varijabla dim postavlja se na manju vrijednost od mogućih
    // maksimalnih broja redaka i stupaca

    int dim = (MAX_RED < MAX_STUP)? MAX_RED : MAX_STUP;

    // učitavanje velicine matrice
    do {
        printf("Upisite vrijednost za broj redaka < %d:", dim);
        scanf("%d", &m);
        printf("Upisite vrijednost za broj stupaca < %d:", dim);
        scanf("%d", &n);
    } while (m < 1 || m > dim || n < 1 || n > dim);

    // učitavanje elemenata matrice
    printf("\nUnos elemenata matrice :\n");
    for (i = 0; i < m; i++) {
        for (j = 0; j < n; j++) {
            printf("Unesite element [%d][%d] : ", i, j);
            scanf("%d", &mat[i][j]);
        }
    }

    // ispis prije transponiranja
    printf("\n\nIspis matrice prije transponiranja:\n");
    for (i = 0; i < m; i++) {
        for (j = 0; j < n; j++) {
            printf("%3d", mat[i][j]);
        }
        printf("\n");
    }
}
```

```

max=m>n ? m : n;
// transponiranje
for ( i=0; i<max; ++i ) {
    for ( j=i+1; j<max; ++j ) { // petlja mora ici od i+1 !!
        pom = mat[i][j];
        mat[i][j] = mat[j][i];
        mat[j][i] = pom;
    }
}

// ispis nakon transponiranja
// broj redaka je sada broj stupaca ...
printf("\nIspis matrice nakon transponiranja:\n");
for (i = 0; i < n; i++) {
    for (j = 0; j < m; j++) {
        printf("%3d", mat[i][j]);
    }
    printf("\n");
}
return 0;
} // main

```

Primjer izvođenja:

```

Upisite vrijednost za broj redaka < 50: 3
Upisite vrijednost za broj stupaca < 50: 2
Unos elemenata matrice :
Unesite element [0][0] : 1
Unesite element [0][1] : 2
Unesite element [1][0] : 3
Unesite element [1][1] : 4
Unesite element [2][0] : 5
Unesite element [2][1] : 6

```

```

Ispis matrice prije transponiranja :
1   2
3   4
5   6

```

```

Ispis matrice nakon transponiranja :
1  3  5
2  4  6

```