

```

1: #include <stdio.h>
2: #include <stdlib.h>
3: #include <math.h>
4:
5: void input(int **m, int a, int b);
6: void input0(int **m, int a, int b);
7: void output(int **m, int a, int b);
8:
9: int main(void)
10: {
11:     printf("input a, b, c : ");
12:     int a, b, c;
13:     scanf("%d %d %d", &a, &b, &c);
14:
15:     int i0;
16:     int **M = malloc(sizeof(int*) * a);
17:     for(i0 = 0 ; i0 < a; i0++)
18:     {
19:         *(M + i0) = malloc(sizeof(int) * b);
20:     }
21:     printf("<1-1> M[a][b] print : \n");
22:
23:     input(M, a, b);
24:     output(M, a, b);
25:     int **N = malloc(sizeof(int *) * b); // N[b][c] malloc
26:
27:     int i1;
28:     for(i1 = 0; i1 < b; i1++)
29:     {
30:         *(N + i1) = malloc(sizeof(int) * c);
31:     }
32:     printf("<1-2> N[b][c] print : \n");
33:
34:     input(N, b, c);
35:     output(N, b, c);
36:
37:     // M * N = L
38:
39:     int **L = malloc(sizeof(int *) * a);
40:
41:     int i2;
42:     for(i2 = 0; i2 < a; i2++)
43:     {
44:         *(L + i2) = malloc(sizeof(int) * c);
45:     }
46:     input0(L, a, c);
47:
48:     int i3;
49:     int j0;
50:     int k0;
51:     for(i3=0 ; i3 < a; i3++) // M * N = L
52:     {
53:         for(j0=0; j0 < c; j0++)
54:         {
55:             for(k0=0; k0 < b; k0++)
56:             {
57:                 (*(L + i3) + j0) += M[i3][k0] * N[k0][j0];
58:             }
59:         }
60:     }
61:
62:     printf("<1-3> L[a][c] print : \n");
63:     output(L, a, c);
64:
65:     // memory free work
66:
67:     int i4;

```

```

68:     for(i4=0 ; i4 < a; i4++)
69:     {
70:         free(M[i4]);
71:     }
72:     free(M);
73:
74:     int i5;
75:     for(i5=0; i5 < b; i5++)
76:     {
77:         free(N[i5]);
78:     }
79:     free(N);
80:
81:     int i6;
82:     for(i6=0 ; i6 < a; i6++)
83:     {
84:         free(L[i6]);
85:     }
86:     free(L);
87:
88:     return 0;
89: }
90: // function line
91:
92: void input(int **m, int a, int b)
93: {
94:     int i7, j1;
95:     for(i7=0 ; i7 < a; i7++)
96:     {
97:         for(j1=0; j1 < b; j1++)
98:         {
99:             (*(m + i7) + j1) = (rand() % 10000) + 1;
100:         }
101:     }
102: }
103:
104: void input0(int **m, int a, int b)
105: {
106:     int i8, j2;
107:     for(i8=0; i8 < a; i8++)
108:     {
109:         for(j2=0; j2 < b; j2++)
110:         {
111:             (*(m + i8) + j2) = 0;
112:         }
113:     }
114: }
115:
116: void output(int **m, int a, int b)
117: {
118:     int i9, j3;
119:     for(i9=0 ; i9 < a; i9++)
120:     {
121:         for(j3=0 ; j3 < b; j3++)
122:         {
123:             printf("%d \t", *(m + i9) + j3));
124:         }
125:         printf("\n");
126:     }
127: }
128:

```