#include <stdio.h>

#include <stdlib.h>

int main(){

int i,j;

int m,n;

double \*\*matrix;

printf("行列の行数と列数を入力: ");

scanf("%d %d",&m,&n);

matrix = (double \*\*)malloc(m \* sizeof(double \*));

for(i=0;i<m;i++){

matrix[i] = (double \*)malloc(n \* sizeof(double \*));

for(j=0;j<n;j++){

printf("(%d %d)の成分を入力: ",i+i,j+1);

scanf("%lf",&matrix[i][j]);

}

}

printf("行列: [\n");

for(i=0;i<m;i++){

for(j=0;j<n;j++){

printf("%lf ",matrix[i][j]);

}

printf("\n");

}

printf("]\n");

for(i=0;i<m;i++){

free(matrix[i]);

}

free(matrix);

return 0;

}

**実行結果**

行列の行数と列数を入力: 2 2

(0 1)の成分を入力: 1

(0 2)の成分を入力: 2

(2 1)の成分を入力: 3

(2 2)の成分を入力: 4

行列: [

1.000000 2.000000

3.000000 4.000000

]

#include <stdio.h>

#include <stdlib.h>

struct point

{

/\* data \*/

double x,y;

};

int main(){

int i,j;

int m;

int \*n;

struct point \*\*points;

printf("多角形の個数を入力: ");

scanf("%d",&m);

points = (struct point \*\*)malloc(m\* sizeof(struct point \*));

n = (int \*)malloc(m \* sizeof(int));

for (i=0;i<m;i++){

printf("多角形%dの頂点の個数を入力: ",i+1);

scanf("%d",&n[i]);

points[i]=(struct point \*)malloc(n[i] \* sizeof(struct point));

for(j=0;j<n[i];j++){

printf("多角形%dの頂点%dのx,y座標を入力: ",i+1,j+1);

scanf("%lf %lf",&points[i][j].x,&points[i][j].y);

}

}

for(i=0;i<m;i++){

printf("多角形%dの頂点:\n",i+1);

for(j=0;j<n[i];j++){

printf("(%lf %lf)\n",points[i][j].x,points[i][j].y);

}

}

}

**実行結果**

多角形の個数を入力: 4

多角形1の頂点の個数を入力: 3

多角形1の頂点1のx,y座標を入力: 0 0

多角形1の頂点2のx,y座標を入力: 2 2

多角形1の頂点3のx,y座標を入力: 3 3

多角形2の頂点の個数を入力: 3

多角形2の頂点1のx,y座標を入力: 0 0

多角形2の頂点2のx,y座標を入力: 1 1

多角形2の頂点3のx,y座標を入力: 2 2

多角形3の頂点の個数を入力: 3

多角形3の頂点1のx,y座標を入力: 0 0

多角形3の頂点2のx,y座標を入力: 1 1

多角形3の頂点3のx,y座標を入力: 2 2

多角形4の頂点の個数を入力: 3

多角形4の頂点1のx,y座標を入力: 0 0

多角形4の頂点2のx,y座標を入力: 1 1

多角形4の頂点3のx,y座標を入力: 2 2

多角形1の頂点:

(0.000000 0.000000)

(2.000000 2.000000)

(3.000000 3.000000)

多角形2の頂点:

(0.000000 0.000000)

(1.000000 1.000000)

(2.000000 2.000000)

多角形3の頂点:

(0.000000 0.000000)

(1.000000 1.000000)

(2.000000 2.000000)

多角形4の頂点:

(0.000000 0.000000)

(1.000000 1.000000)

(2.000000 2.000000)