



# INFORMATICA I

Matriz dinamica con **malloc**

Ing. Juan Carlos Cuttitta

*Universidad Tecnológica Nacional  
Facultad Regional Buenos Aires  
Departamento de Ingeniería Electrónica*

23 de junio de 2020

Ejemplo para armar una matriz (N x N) en forma dinamica utilizando malloc.

En éste ejemplo la matriz será de (2 x 3) quedando de la siguiente forma:

$$\begin{bmatrix} 11 & 12 & 13 \\ 21 & 22 & 23 \end{bmatrix}$$





# Matriz NxN con malloc en Arquitectura X86-32 bits

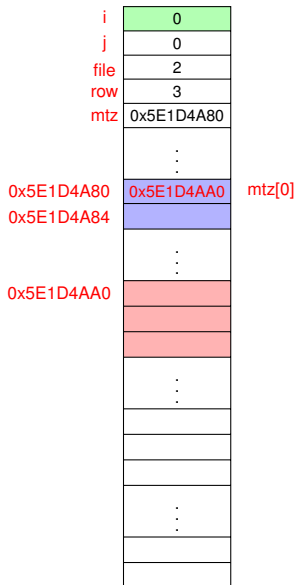
*ejemplo para matriz (2 x 3)*

i	0
j	0
file	2
row	3
mtz	0x5E1D4A80
	⋮
0x5E1D4A80	
0x5E1D4A84	
	⋮
	⋮
	⋮
	⋮
	⋮
	⋮

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main (void)
5 {
6     int file =0,row=0,i=0,j=0;
7     int **mtz;
8
9     printf("ingrese filas y columnas \n");
10    scanf("%d %d",&file ,&row);
11    mtz=(int **)malloc(file*sizeof(int *));
12    for (i=0 ; i<file ; i++){
13        mtz[i]=(int *)malloc(row*sizeof(int));
14    }
15    for (i=0 ; i<file ; i++){
16        for (j=0 ; j<row ; j++){
17            printf("file %d rows %d\n",i,j);
18            scanf("%d",&mtz[i][j]);
19        }
20    }
21    printf("\n");
22    for (i=0 ; i<file ; i++){
23        for (j=0 ; j<row ; j++){
24            printf("%d \t",mtz[i][j]);
25        }
26    }
27    for (i=0 ; i<file ; i++){
28        free(mtz[i]);
29    }
30    free(mtz);
31    return (0);
32 }
```

# Matriz NxN con malloc en Arquitectura X86-32 bits

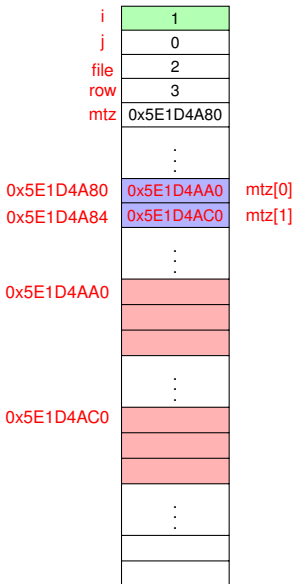
*ejemplo para matriz (2 x 3)*



```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main (void)
5 {
6     int file =0,row=0,i=0,j=0;
7     int **mtz;
8
9     printf("ingrese filas y columnas \n");
10    scanf("%d %d",&file ,&row);
11    mtz=(int **)malloc(file*sizeof(int *));
12    for (i=0 ; i<file ; i++){
13        mtz[i]=(int *)malloc(row*sizeof(int));
14    }
15    for (i=0 ; i<file ; i++){
16        for (j=0 ; j<row ; j++){
17            printf("file %d rows %d\n",i,j);
18            scanf("%d",&mtz[i][j]);
19        }
20    }
21    printf("\n");
22    for (i=0 ; i<file ; i++){
23        for (j=0 ; j<row ; j++){
24            printf("%d \t",mtz[i][j]);
25        }
26    }
27    for (i=0 ; i<file ; i++){
28        free(mtz[i]);
29    }
30    free(mtz);
31    return (0);
32 }
```

# Matriz NxN con malloc en Arquitectura X86-32 bits

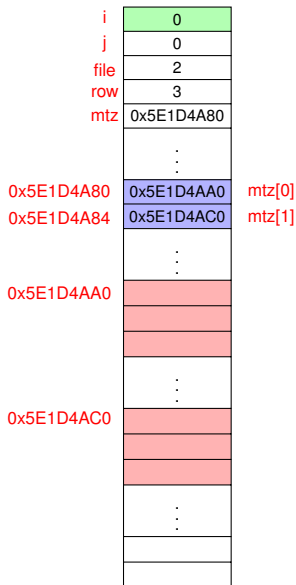
*ejemplo para matriz (2 x 3)*



```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main (void)
5 {
6     int file =0,row=0,i=0,j=0;
7     int **mtz;
8
9     printf("ingrese filas y columnas \n");
10    scanf("%d %d",&file ,&row);
11    mtz=(int **)malloc(file*sizeof(int *));
12    for (i=0 ; i<file ; i++){
13        mtz[i]=(int *)malloc(row*sizeof(int));
14    }
15    for (i=0 ; i<file ; i++){
16        for (j=0 ; j<row ; j++){
17            printf("file %d rows %d\n",i ,j);
18            scanf("%d",&mtz[i][j]);
19        }
20    }
21    printf("\n");
22    for (i=0 ; i<file ; i++){
23        for (j=0 ; j<row ; j++){
24            printf("%d \t",mtz[i][j]);
25        }
26    }
27    for (i=0 ; i<file ; i++){
28        free(mtz[i]);
29    }
30    free(mtz);
31    return (0);
32 }
```

# Matriz NxN con malloc en Arquitectura X86-32 bits

*ejemplo para matriz (2 x 3)*

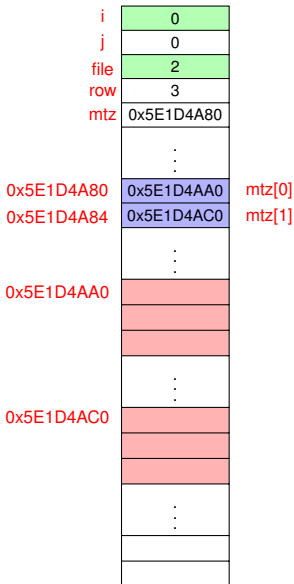


```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main (void)
5 {
6     int file =0,row=0,i=0,j=0;
7     int **mtz;
8
9     printf("ingrese filas y columnas \n");
10    scanf("%d %d",&file ,&row);
11    mtz=(int **)malloc(file*sizeof(int *));
12    for (i=0 ; i<file ; i++){
13        mtz[i]=(int *)malloc(row*sizeof(int));
14    }
15    for (i=0 ; i<file ; i++){
16        for (j=0 ; j<row ; j++){
17            printf("file %d rows %d\n",i ,j);
18            scanf("%d",&mtz[i][j]);
19        }
20    }
21    printf("\n");
22    for (i=0 ; i<file ; i++){
23        for (j=0 ; j<row ; j++){
24            printf("%d \t",mtz[i][j]);
25        }
26    }
27    for (i=0 ; i<file ; i++){
28        free(mtz[i]);
29    }
30    free(mtz);
31    return (0);
32 }
```



# Matriz NxN con malloc en Arquitectura X86-32 bits

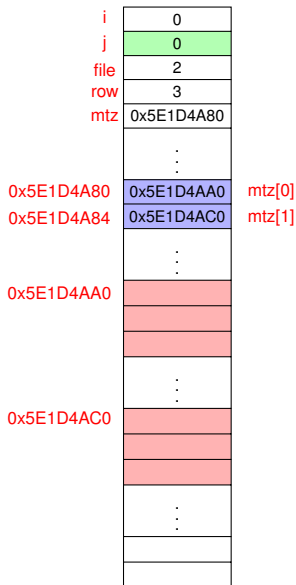
*ejemplo para matriz (2 x 3)*



```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main (void)
5 {
6     int file=0,row=0,i=0,j=0;
7     int **mtz;
8
9     printf("ingrese filas y columnas \n");
10    scanf("%d %d",&file,&row);
11    mtz=(int **)malloc(file*sizeof(int *));
12    for (i=0 ; i<file ; i++){
13        mtz[i]=(int *)malloc(row*sizeof(int));
14    }
15    for (i=0 ; i<file ; i++){
16        for (j=0 ; j<row ; j++){
17            printf("file %d rows %d\n",i,j);
18            scanf("%d",&mtz[i][j]);
19        }
20    }
21    printf("\n");
22    for (i=0 ; i<file ; i++){
23        for (j=0 ; j<row ; j++){
24            printf("%d \t",mtz[i][j]);
25        }
26    }
27    for (i=0 ; i<file ; i++){
28        free(mtz[i]);
29    }
30    free(mtz);
31    return (0);
32 }
```

# Matriz NxN con malloc en Arquitectura X86-32 bits

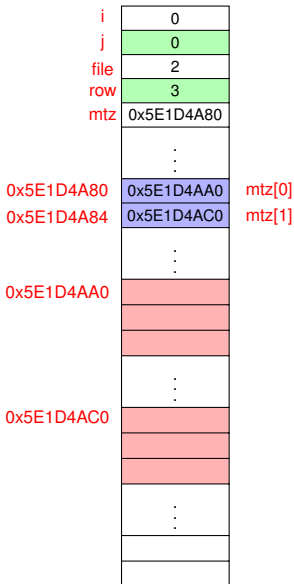
*ejemplo para matriz (2 x 3)*



```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main (void)
5 {
6     int file =0,row=0,i=0,j=0;
7     int **mtz;
8
9     printf("ingrese filas y columnas \n");
10    scanf("%d %d",&file ,&row);
11    mtz=(int **)malloc(file*sizeof(int *));
12    for (i=0 ; i<file ; i++){
13        mtz[i]=(int *)malloc(row*sizeof(int));
14    }
15    for (i=0 ; i<file ; i++){
16        for (j=0 ; j<row ; j++){
17            printf("file %d rows %d\n",i ,j);
18            scanf("%d",&mtz[i][j]);
19        }
20    }
21    printf("\n");
22    for (i=0 ; i<file ; i++){
23        for (j=0 ; j<row ; j++){
24            printf("%d \t",mtz[i][j]);
25        }
26    }
27    for (i=0 ; i<file ; i++){
28        free(mtz[i]);
29    }
30    free(mtz);
31    return (0);
32 }
```

# Matriz NxN con malloc en Arquitectura X86-32 bits

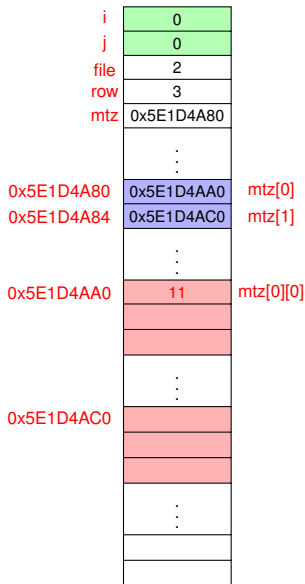
*ejemplo para matriz (2 x 3)*



```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main (void)
5 {
6     int file =0,row=0,i=0,j=0;
7     int **mtz;
8
9     printf("ingrese filas y columnas \n");
10    scanf("%d %d",&file ,&row);
11    mtz=(int **)malloc(file*sizeof(int *));
12    for (i=0 ; i<file ; i++){
13        mtz[i]=(int *)malloc(row*sizeof(int));
14    }
15    for (i=0 ; i<file ; i++){
16        for (j=0 ; j<row ; j++){
17            printf("file %d rows %d\n",i ,j);
18            scanf("%d",&mtz[i][j]);
19        }
20    }
21    printf("\n");
22    for (i=0 ; i<file ; i++){
23        for (j=0 ; j<row ; j++){
24            printf("%d \t",mtz[i][j]);
25        }
26    }
27    for (i=0 ; i<file ; i++){
28        free(mtz[i]);
29    }
30    free(mtz);
31    return (0);
32 }
```

# Matriz NxN con malloc en Arquitectura X86-32 bits

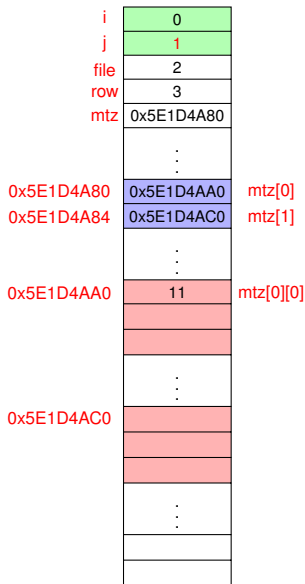
*ejemplo para matriz (2 x 3)*



```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main (void)
5 {
6     int file =0,row=0,i=0,j=0;
7     int **mtz;
8
9     printf("ingrese filas y columnas \n");
10    scanf("%d %d",&file ,&row);
11    mtz=(int **)malloc(file*sizeof(int *));
12    for (i=0 ; i<file ; i++){
13        mtz[i]=(int *)malloc(row*sizeof(int));
14    }
15    for (i=0 ; i<file ; i++){
16        for (j=0 ; j<row ; j++){
17            printf("file %d rows %d\n",i ,j);
18            scanf("%d",&mtz[i][j]);
19        }
20    }
21    printf("\n");
22    for (i=0 ; i<file ; i++){
23        for (j=0 ; j<row ; j++){
24            printf("%d \t",mtz[i][j]);
25        }
26    }
27    for (i=0 ; i<file ; i++){
28        free(mtz[i]);
29    }
30    free(mtz);
31    return (0);
32 }
```

# Matriz NxN con malloc en Arquitectura X86-32 bits

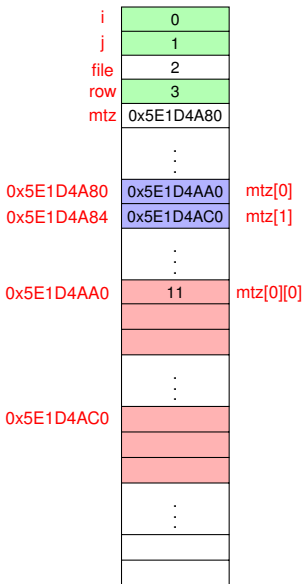
*ejemplo para matriz (2 x 3)*



```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main (void)
5 {
6     int file =0,row=0,i=0,j=0;
7     int **mtz;
8
9     printf("ingrese filas y columnas \n");
10    scanf("%d %d",&file ,&row);
11    mtz=(int **)malloc(file*sizeof(int *));
12    for (i=0 ; i<file ; i++){
13        mtz[i]=(int *)malloc(row*sizeof(int));
14    }
15    for (i=0 ; i<file ; i++){
16        for (j=0 ; j<row ; j++){
17            printf("file %d rows %d\n",i ,j);
18            scanf("%d",&mtz[i][j]);
19        }
20    }
21    printf("\n");
22    for (i=0 ; i<file ; i++){
23        for (j=0 ; j<row ; j++){
24            printf("%d \t",mtz[i][j]);
25        }
26    }
27    for (i=0 ; i<file ; i++){
28        free(mtz[i]);
29    }
30    free(mtz);
31    return (0);
32 }
```

# Matriz NxN con malloc en Arquitectura X86-32 bits

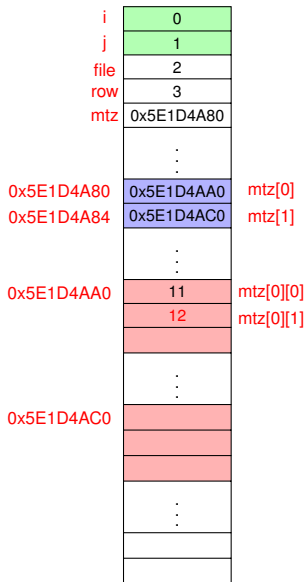
*ejemplo para matriz (2 x 3)*



```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main (void)
5 {
6     int file =0,row=0,i=0,j=0;
7     int **mtz;
8
9     printf("ingrese filas y columnas \n");
10    scanf("%d %d",&file ,&row);
11    mtz=(int **)malloc(file*sizeof(int *));
12    for (i=0 ; i<file ; i++){
13        mtz[i]=(int *)malloc(row*sizeof(int));
14    }
15    for (i=0 ; i<file ; i++){
16        for (j=0 ; j<row ; j++){
17            printf("file %d rows %d\n",i ,j);
18            scanf("%d",&mtz[i][j]);
19        }
20    }
21    printf("\n");
22    for (i=0 ; i<file ; i++){
23        for (j=0 ; j<row ; j++){
24            printf("%d \t",mtz[i][j]);
25        }
26    }
27    for (i=0 ; i<file ; i++){
28        free(mtz[i]);
29    }
30    free(mtz);
31    return (0);
32 }
```

# Matriz NxN con malloc en Arquitectura X86-32 bits

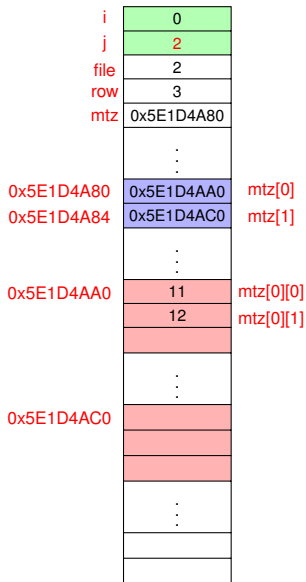
*ejemplo para matriz (2 x 3)*



```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main (void)
5 {
6     int file =0,row=0,i=0,j=0;
7     int **mtz;
8
9     printf("ingrese filas y columnas \n");
10    scanf("%d %d",&file ,&row);
11    mtz=(int **)malloc(file*sizeof(int *));
12    for (i=0 ; i<file ; i++){
13        mtz[i]=(int *)malloc(row*sizeof(int));
14    }
15    for (i=0 ; i<file ; i++){
16        for (j=0 ; j<row ; j++){
17            printf("file %d rows %d\n",i ,j);
18            scanf("%d",&mtz[i][j]);
19        }
20    }
21    printf("\n");
22    for (i=0 ; i<file ; i++){
23        for (j=0 ; j<row ; j++){
24            printf("%d \t",mtz[i][j]);
25        }
26    }
27    for (i=0 ; i<file ; i++){
28        free(mtz[i]);
29    }
30    free(mtz);
31    return (0);
32 }
```

# Matriz NxN con malloc en Arquitectura X86-32 bits

*ejemplo para matriz (2 x 3)*

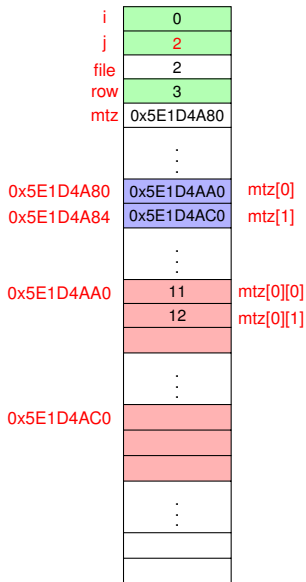


```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main (void)
5 {
6     int file =0,row=0,i=0,j=0;
7     int **mtz;
8
9     printf("ingrese filas y columnas \n");
10    scanf("%d %d",&file ,&row);
11    mtz=(int **)malloc(file*sizeof(int *));
12    for (i=0 ; i<file ; i++){
13        mtz[i]=(int *)malloc(row*sizeof(int));
14    }
15    for (i=0 ; i<file ; i++){
16        for (j=0 ; j<row ; j++){
17            printf("file %d rows %d\n",i ,j);
18            scanf("%d",&mtz[i][j]);
19        }
20    }
21    printf("\n");
22    for (i=0 ; i<file ; i++){
23        for (j=0 ; j<row ; j++){
24            printf("%d \t",mtz[i][j]);
25        }
26    }
27    for (i=0 ; i<file ; i++){
28        free(mtz[i]);
29    }
30    free(mtz);
31    return (0);
32 }
```



# Matriz NxN con malloc en Arquitectura X86-32 bits

*ejemplo para matriz (2 x 3)*



```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main (void)
5 {
6     int file =0,row=0,i=0,j=0;
7     int **mtz;
8
9     printf("ingrese filas y columnas \n");
10    scanf("%d %d",&file ,&row);
11    mtz=(int **)malloc(file*sizeof(int *));
12    for (i=0 ; i<file ; i++){
13        mtz[i]=(int *)malloc(row*sizeof(int));
14    }
15    for (i=0 ; i<file ; i++){
16        for (j=0 ; j<row ; j++){
17            printf("file %d rows %d\n",i ,j);
18            scanf("%d",&mtz[i][j]);
19        }
20    }
21    printf("\n");
22    for (i=0 ; i<file ; i++){
23        for (j=0 ; j<row ; j++){
24            printf("%d \t",mtz[i][j]);
25        }
26    }
27    for (i=0 ; i<file ; i++){
28        free(mtz[i]);
29    }
30    free(mtz);
31    return (0);
32 }
```

# Matriz NxN con malloc en Arquitectura X86-32 bits

*ejemplo para matriz (2 x 3)*

i	0	
j	2	
file	2	
row	3	
mtz	0x5E1D4A80	
	⋮	
0x5E1D4A80	0x5E1D4AA0	mtz[0]
0x5E1D4A84	0x5E1D4AC0	mtz[1]
	⋮	
0x5E1D4AA0	11	mtz[0][0]
	12	mtz[0][1]
	13	mtz[0][2]
	⋮	
0x5E1D4AC0		
	⋮	

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main (void)
5 {
6     int file =0,row=0,i=0,j=0;
7     int **mtz;
8
9     printf("ingrese filas y columnas \n");
10    scanf("%d %d",&file ,&row);
11    mtz=(int **)malloc(file*sizeof(int *));
12    for (i=0 ; i<file ; i++){
13        mtz[i]=(int *)malloc(row*sizeof(int));
14    }
15    for (i=0 ; i<file ; i++){
16        for (j=0 ; j<row ; j++){
17            printf("file %d rows %d\n",i ,j);
18            scanf("%d",&mtz[i][j]);
19        }
20    }
21    printf("\n");
22    for (i=0 ; i<file ; i++){
23        for (j=0 ; j<row ; j++){
24            printf("%d \t",mtz[i][j]);
25        }
26    }
27    for (i=0 ; i<file ; i++){
28        free(mtz[i]);
29    }
30    free(mtz);
31    return (0);
32 }
```

# Matriz NxN con malloc en Arquitectura X86-32 bits

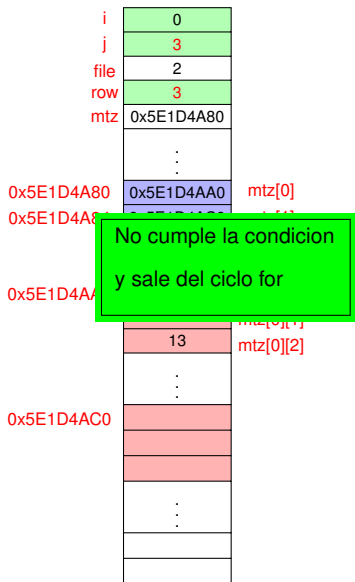
*ejemplo para matriz (2 x 3)*

i	0	
j	3	
file	2	
row	3	
mtz	0x5E1D4A80	
	⋮	
0x5E1D4A80	0x5E1D4AA0	mtz[0]
0x5E1D4A84	0x5E1D4AC0	mtz[1]
	⋮	
0x5E1D4AA0	11	mtz[0][0]
	12	mtz[0][1]
	13	mtz[0][2]
	⋮	
0x5E1D4AC0		
	⋮	

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main (void)
5 {
6     int file =0,row=0,i=0,j=0;
7     int **mtz;
8
9     printf("ingrese filas y columnas \n");
10    scanf("%d %d",&file ,&row);
11    mtz=(int **)malloc(file*sizeof(int *));
12    for (i=0 ; i<file ; i++){
13        mtz[i]=(int *)malloc(row*sizeof(int));
14    }
15    for (i=0 ; i<file ; i++){
16        for (j=0 ; j<row ; j++){
17            printf("file %d rows %d\n",i ,j);
18            scanf("%d",&mtz[i][j]);
19        }
20    }
21    printf("\n");
22    for (i=0 ; i<file ; i++){
23        for (j=0 ; j<row ; j++){
24            printf("%d \t",mtz[i][j]);
25        }
26    }
27    for (i=0 ; i<file ; i++){
28        free(mtz[i]);
29    }
30    free(mtz);
31    return (0);
32 }
```

# Matriz NxN con malloc en Arquitectura X86-32 bits

ejemplo para matriz (2 x 3)



```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main (void)
5 {
6     int file =0,row=0,i=0,j=0;
7     int **mtz;
8
9     printf("ingrese filas y columnas \n");
10    scanf("%d %d",&file ,&row);
11    mtz=(int **)malloc(file*sizeof(int *));
12    for (i=0 ; i<file ; i++){
13        mtz[i]=(int *)malloc(row*sizeof(int));
14    }
15    for (i=0 ; i<file ; i++){
16        for (j=0 ; j<row ; j++){
17            printf("file %d rows %d\n",i ,j);
18            scanf("%d",&mtz[i][j]);
19        }
20    }
21    printf("\n");
22    for (i=0 ; i<file ; i++){
23        for (j=0 ; j<row ; j++){
24            printf("%d \t",mtz[i][j]);
25        }
26    }
27    for (i=0 ; i<file ; i++){
28        free(mtz[i]);
29    }
30    free(mtz);
31    return (0);
32 }
```

# Matriz NxN con malloc en Arquitectura X86-32 bits

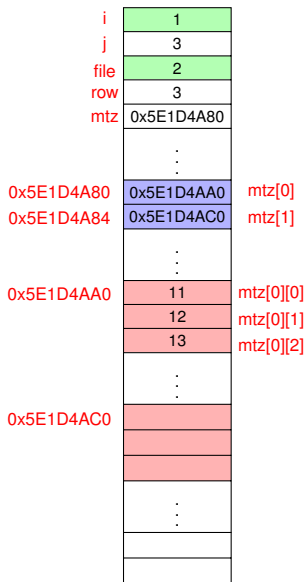
*ejemplo para matriz (2 x 3)*

i	1	
j	3	
file	2	
row	3	
mtz	0x5E1D4A80	
	⋮	
0x5E1D4A80	0x5E1D4AA0	mtz[0]
0x5E1D4A84	0x5E1D4AC0	mtz[1]
	⋮	
0x5E1D4AA0	11	mtz[0][0]
	12	mtz[0][1]
	13	mtz[0][2]
	⋮	
0x5E1D4AC0		
	⋮	

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main (void)
5 {
6     int file =0,row=0,i=0,j=0;
7     int **mtz;
8
9     printf("ingrese filas y columnas \n");
10    scanf("%d %d",&file ,&row);
11    mtz=(int **)malloc(file*sizeof(int *));
12    for (i=0 ; i<file ; i++){
13        mtz[i]=(int *)malloc(row*sizeof(int));
14    }
15    for (i=0 ; i<file ; i++){
16        for (j=0 ; j<row ; j++){
17            printf("file %d rows %d\n",i ,j);
18            scanf("%d",&mtz[i][j]);
19        }
20    }
21    printf("\n");
22    for (i=0 ; i<file ; i++){
23        for (j=0 ; j<row ; j++){
24            printf("%d \t",mtz[i][j]);
25        }
26    }
27    for (i=0 ; i<file ; i++){
28        free(mtz[i]);
29    }
30    free(mtz);
31    return (0);
32 }
```

# Matriz NxN con malloc en Arquitectura X86-32 bits

*ejemplo para matriz (2 x 3)*



```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main (void)
5 {
6     int file =0,row=0,i=0,j=0;
7     int **mtz;
8
9     printf("ingrese filas y columnas \n");
10    scanf("%d %d",&file ,&row);
11    mtz=(int **)malloc(file*sizeof(int *));
12    for (i=0 ; i<file ; i++){
13        mtz[i]=(int *)malloc(row*sizeof(int));
14    }
15    for (i=0 ; i<file ; i++){
16        for (j=0 ; j<row ; j++){
17            printf("file %d rows %d\n",i ,j);
18            scanf("%d",&mtz[i][j]);
19        }
20    }
21    printf("\n");
22    for (i=0 ; i<file ; i++){
23        for (j=0 ; j<row ; j++){
24            printf("%d \t",mtz[i][j]);
25        }
26    }
27    for (i=0 ; i<file ; i++){
28        free(mtz[i]);
29    }
30    free(mtz);
31    return (0);
32 }
```

# Matriz NxN con malloc en Arquitectura X86-32 bits

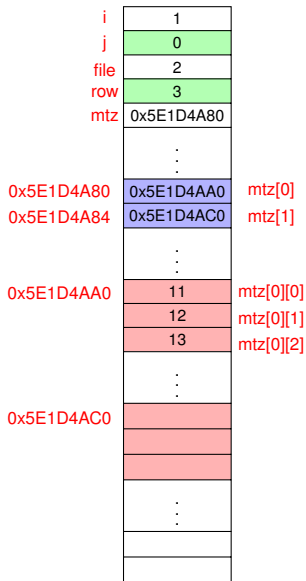
*ejemplo para matriz (2 x 3)*

i	1	
j	0	
file	2	
row	3	
mtz	0x5E1D4A80	
	⋮	
0x5E1D4A80	0x5E1D4AA0	mtz[0]
0x5E1D4A84	0x5E1D4AC0	mtz[1]
	⋮	
0x5E1D4AA0	11	mtz[0][0]
	12	mtz[0][1]
	13	mtz[0][2]
	⋮	
0x5E1D4AC0		
	⋮	

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main (void)
5 {
6     int file =0,row=0,i=0,j=0;
7     int **mtz;
8
9     printf("ingrese filas y columnas \n");
10    scanf("%d %d",&file ,&row);
11    mtz=(int **)malloc(file*sizeof(int *));
12    for (i=0 ; i<file ; i++){
13        mtz[i]=(int *)malloc(row*sizeof(int));
14    }
15    for (i=0 ; i<file ; i++){
16        for (j=0 ; j<row ; j++){
17            printf("file %d rows %d\n",i ,j);
18            scanf("%d",&mtz[i][j]);
19        }
20    }
21    printf("\n");
22    for (i=0 ; i<file ; i++){
23        for (j=0 ; j<row ; j++){
24            printf("%d \t",mtz[i][j]);
25        }
26    }
27    for (i=0 ; i<file ; i++){
28        free(mtz[i]);
29    }
30    free(mtz);
31    return (0);
32 }
```

# Matriz NxN con malloc en Arquitectura X86-32 bits

*ejemplo para matriz (2 x 3)*

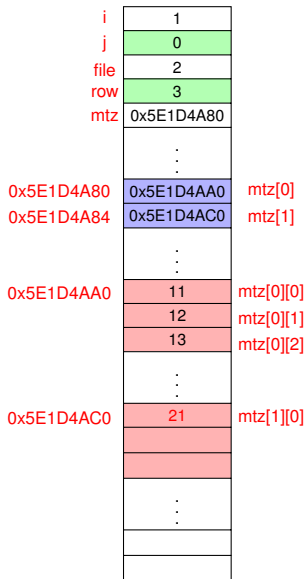


```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main (void)
5 {
6     int file =0,row=0,i=0,j=0;
7     int **mtz;
8
9     printf("ingrese filas y columnas \n");
10    scanf("%d %d",&file ,&row);
11    mtz=(int **)malloc(file*sizeof(int *));
12    for (i=0 ; i<file ; i++){
13        mtz[i]=(int *)malloc(row*sizeof(int));
14    }
15    for (i=0 ; i<file ; i++){
16        for (j=0 ; j<row ; j++){
17            printf("file %d rows %d\n",i ,j);
18            scanf("%d",&mtz[i][j]);
19        }
20    }
21    printf("\n");
22    for (i=0 ; i<file ; i++){
23        for (j=0 ; j<row ; j++){
24            printf("%d \t",mtz[i][j]);
25        }
26    }
27    for (i=0 ; i<file ; i++){
28        free(mtz[i]);
29    }
30    free(mtz);
31    return (0);
32 }
```



# Matriz NxN con malloc en Arquitectura X86-32 bits

*ejemplo para matriz (2 x 3)*



```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main (void)
5 {
6     int file =0,row=0,i=0,j=0;
7     int **mtz;
8
9     printf("ingrese filas y columnas \n");
10    scanf("%d %d",&file ,&row);
11    mtz=(int **)malloc(file*sizeof(int *));
12    for (i=0 ; i<file ; i++){
13        mtz[i]=(int *)malloc(row*sizeof(int));
14    }
15    for (i=0 ; i<file ; i++){
16        for (j=0 ; j<row ; j++){
17            printf("file %d rows %d\n",i ,j);
18            scanf("%d",&mtz[i][j]);
19        }
20    }
21    printf("\n");
22    for (i=0 ; i<file ; i++){
23        for (j=0 ; j<row ; j++){
24            printf("%d \t",mtz[i][j]);
25        }
26    }
27    for (i=0 ; i<file ; i++){
28        free(mtz[i]);
29    }
30    free(mtz);
31    return (0);
32 }
```

# Matriz NxN con malloc en Arquitectura X86-32 bits

*ejemplo para matriz (2 x 3)*

i	1	
j	1	
file	2	
row	3	
mtz	0x5E1D4A80	
	⋮	
0x5E1D4A80	0x5E1D4AA0	mtz[0]
0x5E1D4A84	0x5E1D4AC0	mtz[1]
	⋮	
0x5E1D4AA0	11	mtz[0][0]
	12	mtz[0][1]
	13	mtz[0][2]
	⋮	
0x5E1D4AC0	21	mtz[1][0]
	⋮	

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main (void)
5 {
6     int file =0,row=0,i=0,j=0;
7     int **mtz;
8
9     printf("ingrese filas y columnas \n");
10    scanf("%d %d",&file ,&row);
11    mtz=(int **)malloc(file*sizeof(int *));
12    for (i=0 ; i<file ; i++){
13        mtz[i]=(int *)malloc(row*sizeof(int));
14    }
15    for (i=0 ; i<file ; i++){
16        for (j=0 ; j<row ; j++){
17            printf("file %d rows %d\n",i,j);
18            scanf("%d",&mtz[i][j]);
19        }
20    }
21    printf("\n");
22    for (i=0 ; i<file ; i++){
23        for (j=0 ; j<row ; j++){
24            printf("%d \t",mtz[i][j]);
25        }
26    }
27    for (i=0 ; i<file ; i++){
28        free(mtz[i]);
29    }
30    free(mtz);
31    return (0);
32 }
```

# Matriz NxN con malloc en Arquitectura X86-32 bits

*ejemplo para matriz (2 x 3)*

i	1	
j	1	
file	2	
row	3	
mtz	0x5E1D4A80	
	⋮	
0x5E1D4A80	0x5E1D4AA0	mtz[0]
0x5E1D4A84	0x5E1D4AC0	mtz[1]
	⋮	
0x5E1D4AA0	11	mtz[0][0]
	12	mtz[0][1]
	13	mtz[0][2]
	⋮	
0x5E1D4AC0	21	mtz[1][0]
	⋮	

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main (void)
5 {
6     int file =0,row=0,i=0,j=0;
7     int **mtz;
8
9     printf("ingrese filas y columnas \n");
10    scanf("%d %d",&file ,&row);
11    mtz=(int **)malloc(file*sizeof(int *));
12    for (i=0 ; i<file ; i++){
13        mtz[i]=(int *)malloc(row*sizeof(int));
14    }
15    for (i=0 ; i<file ; i++){
16        for (j=0 ; j<row ; j++){
17            printf("file %d rows %d\n",i ,j);
18            scanf("%d",&mtz[i][j]);
19        }
20    }
21    printf("\n");
22    for (i=0 ; i<file ; i++){
23        for (j=0 ; j<row ; j++){
24            printf("%d \t",mtz[i][j]);
25        }
26    }
27    for (i=0 ; i<file ; i++){
28        free(mtz[i]);
29    }
30    free(mtz);
31    return (0);
32 }
```

# Matriz NxN con malloc en Arquitectura X86-32 bits

*ejemplo para matriz (2 x 3)*

i	1	
j	1	
file	2	
row	3	
mtz	0x5E1D4A80	
	⋮	
0x5E1D4A80	0x5E1D4AA0	mtz[0]
0x5E1D4A84	0x5E1D4AC0	mtz[1]
	⋮	
0x5E1D4AA0	11	mtz[0][0]
	12	mtz[0][1]
	13	mtz[0][2]
	⋮	
0x5E1D4AC0	21	mtz[1][0]
	22	mtz[1][1]
	⋮	

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main (void)
5 {
6     int file=0,row=0,i=0,j=0;
7     int **mtz;
8
9     printf("ingrese filas y columnas \n");
10    scanf("%d %d",&file,&row);
11    mtz=(int **)malloc(file*sizeof(int *));
12    for (i=0 ; i<file ; i++){
13        mtz[i]=(int *)malloc(row*sizeof(int));
14    }
15    for (i=0 ; i<file ; i++){
16        for (j=0 ; j<row ; j++){
17            printf("file %d rows %d\n",i,j);
18            scanf("%d",&mtz[i][j]);
19        }
20    }
21    printf("\n");
22    for (i=0 ; i<file ; i++){
23        for (j=0 ; j<row ; j++){
24            printf("%d \t",mtz[i][j]);
25        }
26    }
27    for (i=0 ; i<file ; i++){
28        free(mtz[i]);
29    }
30    free(mtz);
31    return (0);
32 }
```

# Matriz NxN con malloc en Arquitectura X86-32 bits

*ejemplo para matriz (2 x 3)*

i	1	
j	2	
file	2	
row	3	
mtz	0x5E1D4A80	
	⋮	
0x5E1D4A80	0x5E1D4AA0	mtz[0]
0x5E1D4A84	0x5E1D4AC0	mtz[1]
	⋮	
0x5E1D4AA0	11	mtz[0][0]
	12	mtz[0][1]
	13	mtz[0][2]
	⋮	
0x5E1D4AC0	21	mtz[1][0]
	22	mtz[1][1]
	⋮	

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main (void)
5 {
6     int file =0,row=0,i=0,j=0;
7     int **mtz;
8
9     printf("ingrese filas y columnas \n");
10    scanf("%d %d",&file ,&row);
11    mtz=(int **)malloc(file*sizeof(int *));
12    for (i=0 ; i<file ; i++){
13        mtz[i]=(int *)malloc(row*sizeof(int));
14    }
15    for (i=0 ; i<file ; i++){
16        for (j=0 ; j<row ; j++){
17            printf("file %d rows %d\n",i ,j);
18            scanf("%d",&mtz[i][j]);
19        }
20    }
21    printf("\n");
22    for (i=0 ; i<file ; i++){
23        for (j=0 ; j<row ; j++){
24            printf("%d \t",mtz[i][j]);
25        }
26    }
27    for (i=0 ; i<file ; i++){
28        free(mtz[i]);
29    }
30    free(mtz);
31    return (0);
32 }
```

# Matriz NxN con malloc en Arquitectura X86-32 bits

*ejemplo para matriz (2 x 3)*

i	1	
j	2	
file	2	
row	3	
mtz	0x5E1D4A80	
	⋮	
0x5E1D4A80	0x5E1D4AA0	mtz[0]
0x5E1D4A84	0x5E1D4AC0	mtz[1]
	⋮	
0x5E1D4AA0	11	mtz[0][0]
	12	mtz[0][1]
	13	mtz[0][2]
	⋮	
0x5E1D4AC0	21	mtz[1][0]
	22	mtz[1][1]
	⋮	

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main (void)
5 {
6     int file =0,row=0,i=0,j=0;
7     int **mtz;
8
9     printf("ingrese filas y columnas \n");
10    scanf("%d %d",&file ,&row);
11    mtz=(int **)malloc(file*sizeof(int *));
12    for (i=0 ; i<file ; i++){
13        mtz[i]=(int *)malloc(row*sizeof(int));
14    }
15    for (i=0 ; i<file ; i++){
16        for (j=0 ; j<row ; j++){
17            printf("file %d rows %d\n",i ,j);
18            scanf("%d",&mtz[i][j]);
19        }
20    }
21    printf("\n");
22    for (i=0 ; i<file ; i++){
23        for (j=0 ; j<row ; j++){
24            printf("%d \t",mtz[i][j]);
25        }
26    }
27    for (i=0 ; i<file ; i++){
28        free(mtz[i]);
29    }
30    free(mtz);
31    return (0);
32 }
```

# Matriz NxN con malloc en Arquitectura X86-32 bits

*ejemplo para matriz (2 x 3)*

i	1	
j	2	
file	2	
row	3	
mtz	0x5E1D4A80	
	⋮	
0x5E1D4A80	0x5E1D4AA0	mtz[0]
0x5E1D4A84	0x5E1D4AC0	mtz[1]
	⋮	
0x5E1D4AA0	11	mtz[0][0]
	12	mtz[0][1]
	13	mtz[0][2]
	⋮	
0x5E1D4AC0	21	mtz[1][0]
	22	mtz[1][1]
	23	mtz[1][2]
	⋮	

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main (void)
5 {
6     int file =0,row=0,i=0,j=0;
7     int **mtz;
8
9     printf("ingrese filas y columnas \n");
10    scanf("%d %d",&file ,&row);
11    mtz=(int **)malloc(file*sizeof(int *));
12    for (i=0 ; i<file ; i++){
13        mtz[i]=(int *)malloc(row*sizeof(int));
14    }
15    for (i=0 ; i<file ; i++){
16        for (j=0 ; j<row ; j++){
17            printf("file %d rows %d\n",i ,j);
18            scanf("%d",&mtz[i][j]);
19        }
20    }
21    printf("\n");
22    for (i=0 ; i<file ; i++){
23        for (j=0 ; j<row ; j++){
24            printf("%d \t",mtz[i][j]);
25        }
26    }
27    for (i=0 ; i<file ; i++){
28        free(mtz[i]);
29    }
30    free(mtz);
31    return (0);
32 }
```

# Matriz NxN con malloc en Arquitectura X86-32 bits

*ejemplo para matriz (2 x 3)*

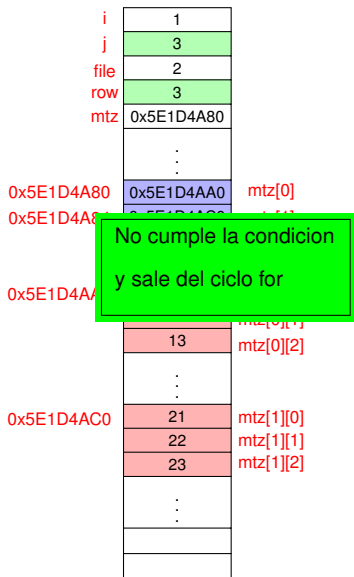
i	1	
j	3	
file	2	
row	3	
mtz	0x5E1D4A80	
	⋮	
0x5E1D4A80	0x5E1D4AA0	mtz[0]
0x5E1D4A84	0x5E1D4AC0	mtz[1]
	⋮	
0x5E1D4AA0	11	mtz[0][0]
	12	mtz[0][1]
	13	mtz[0][2]
	⋮	
0x5E1D4AC0	21	mtz[1][0]
	22	mtz[1][1]
	23	mtz[1][2]
	⋮	

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main (void)
5 {
6     int file =0,row=0,i=0,j=0;
7     int **mtz;
8
9     printf("ingrese filas y columnas \n");
10    scanf("%d %d",&file ,&row);
11    mtz=(int **)malloc(file*sizeof(int *));
12    for (i=0 ; i<file ; i++){
13        mtz[i]=(int *)malloc(row*sizeof(int));
14    }
15    for (i=0 ; i<file ; i++){
16        for (j=0 ; j<row ; j++){
17            printf("file %d rows %d\n",i ,j);
18            scanf("%d",&mtz[i][j]);
19        }
20    }
21    printf("\n");
22    for (i=0 ; i<file ; i++){
23        for (j=0 ; j<row ; j++){
24            printf("%d \t",mtz[i][j]);
25        }
26    }
27    for (i=0 ; i<file ; i++){
28        free(mtz[i]);
29    }
30    free(mtz);
31    return (0);
32 }
```



# Matriz NxN con malloc en Arquitectura X86-32 bits

ejemplo para matriz (2 x 3)



# Matriz NxN con malloc en Arquitectura X86-32 bits

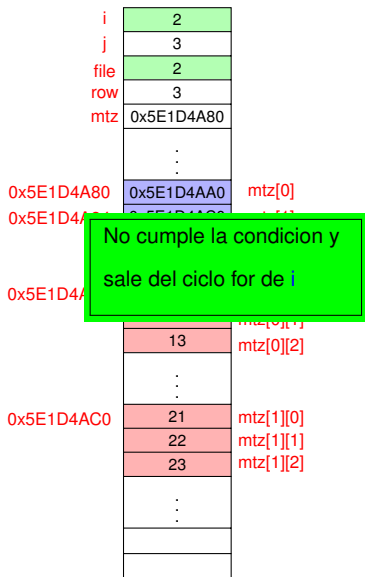
*ejemplo para matriz (2 x 3)*

i	2	
j	3	
file	2	
row	3	
mtz	0x5E1D4A80	
	:	
	:	
0x5E1D4A80	0x5E1D4AA0	mtz[0]
0x5E1D4A84	0x5E1D4AC0	mtz[1]
	:	
	:	
0x5E1D4AA0	11	mtz[0][0]
	12	mtz[0][1]
	13	mtz[0][2]
	:	
	:	
0x5E1D4AC0	21	mtz[1][0]
	22	mtz[1][1]
	23	mtz[1][2]
	:	
	:	

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main (void)
5 {
6     int file=0,row=0,i=0,j=0;
7     int **mtz;
8
9     printf("ingrese filas y columnas \n");
10    scanf("%d %d",&file,&row);
11    mtz=(int **)malloc(file*sizeof(int *));
12    for (i=0 ; i<file ; i++){
13        mtz[i]=(int *)malloc(row*sizeof(int));
14    }
15    for (i=0 ; i<file ; i++){
16        for (j=0 ; j<row ; j++){
17            printf("file %d rows %d\n",i,j);
18            scanf("%d",&mtz[i][j]);
19        }
20    }
21    printf("\n");
22    for (i=0 ; i<file ; i++){
23        for (j=0 ; j<row ; j++){
24            printf("%d \t",mtz[i][j]);
25        }
26    }
27    for (i=0 ; i<file ; i++){
28        free(mtz[i]);
29    }
30    free(mtz);
31    return (0);
32 }
```

# Matriz NxN con malloc en Arquitectura X86-32 bits

*ejemplo para matriz (2 x 3)*



```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main (void)
5 {
6     int file =0,row=0,i=0,j=0;
7     int **mtz;
8
9     printf("ingrese filas y columnas \n");
10    scanf("%d %d",&file ,&row);
11    mtz=(int **)malloc(file*sizeof(int *));
12    for (i=0 ; i<file ; i++){
13        mtz[i]=(int *)malloc(row*sizeof(int));
14    }
15    for (i=0 ; i<file ; i++){
16        for (j=0 ; j<row ; j++){
17            printf("file %d rows %d\n",i ,j);
18            scanf("%d",&mtz[i][j]);
19        }
20    }
21    printf("\n");
22    for (i=0 ; i<file ; i++){
23        for (j=0 ; j<row ; j++){
24            printf("%d \t",mtz[i][j]);
25        }
26    }
27    for (i=0 ; i<file ; i++){
28        free(mtz[i]);
29    }
30    free(mtz);
31    return (0);
32 }
```

# Matriz NxN con malloc en Arquitectura X86-32 bits

*ejemplo para matriz (2 x 3)*

i	2	
j	3	
file	2	
row	3	
mtz	0x5E1D4A80	
	⋮	
0x5E1D4A80	0x5E1D4AA0	mtz[0]
0x5E1D4A84	0x5E1D4AC0	mtz[1]
	⋮	
0x5E1D4AA0	11	mtz[0][0]
	12	mtz[0][1]
	13	mtz[0][2]
	⋮	
0x5E1D4AC0	21	mtz[1][0]
	22	mtz[1][1]
	23	mtz[1][2]

11	12	13
21	22	23

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main (void)
5 {
6     int file =0,row=0,i=0,j=0;
7     int **mtz;
8
9     printf("ingrese filas y columnas \n");
10    scanf("%d %d",&file ,&row);
11    mtz=(int **)malloc(file*sizeof(int *));
12    for (i=0 ; i<file ; i++){
13        mtz[i]=(int *)malloc(row*sizeof(int));
14    }
15    for (i=0 ; i<file ; i++){
16        for (j=0 ; j<row ; j++){
17            printf("file %d rows %d\n",i ,j);
18            scanf("%d",&mtz[i][j]);
19        }
20    }
21    printf("\n");
22    for (i=0 ; i<file ; i++){
23        for (j=0 ; j<row ; j++){
24            printf("%d \t",mtz[i][j]);
25        }
26    }
27    for (i=0 ; i<file ; i++){
28        free(mtz[i]);
29    }
30    free(mtz);
31    return (0);
32 }
```

# Matriz NxN con malloc en Arquitectura X86-32 bits

*ejemplo para matriz (2 x 3)*

i	2
j	3
file	2
row	3
mtz	0XXXXXXXX
	.
	.
	.
	.
	11
	12
	13
	.
	.
	21
	22
	23

Libero toda la memoria  
pedida con malloc

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main (void)
5 {
6     int file =0,row=0,i=0,j=0;
7     int **mtz;
8
9     printf("ingrese filas y columnas \n");
10    scanf("%d %d",&file ,&row);
11    mtz=(int **)malloc(file*sizeof(int *));
12    for (i=0 ; i<file ; i++){
13        mtz[i]=(int *)malloc(row*sizeof(int));
14    }
15    for (i=0 ; i<file ; i++){
16        for (j=0 ; j<row ; j++){
17            printf("file %d rows%d\n",i ,j);
18            scanf("%d",&mtz[i][j]);
19        }
20    }
21    printf("\n");
22    for (i=0 ; i<file ; i++){
23        for (j=0 ; j<row ; j++){
24            printf("%d \t",mtz[i][j]);
25        }
26    }
27    for (i=0 ; i<file ; i++){
28        free(mtz[i]);
29    }
30    free(mtz);
31    return (0);
32 }
```