

C

Jan de Leeuw

2023-11-12

```
#include <stdio.h>
#include <stdlib.h>
#include <R.h>

void riddle(int *pn, int *pp, double *m1, double *m2, double *m3) {
    int n = *pn, p = *pp;
    double fill[8] = {1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0};
    double(*mot)[n] = malloc((sizeof *mot) * p);
    double(*mat)[n][p] = malloc(sizeof *mat);
    Rprintf("sizeof *mot %3d sizeof mot %3d\n", (int)sizeof *mot, (int)sizeof mot);
    Rprintf("sizeof *mat %3d sizeof mat %3d\n", (int)sizeof *mat, (int)sizeof mat);
    int k = 0;
    for (int j = 0; j < p; j++) {
        Rprintf("sizeof (mat + j) %3d sizeof *(mot + j) %3d sizeof *(mat + j) %3d\n",
            (int)sizeof (mat + j), (int)sizeof *(mot + j), (int)sizeof *(mat + j));
    }
    return;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < p; j++) {
            int ij = (j - 1) * n + i;
            mot[i][j] = fill[k];
            *(m1 + ij) = mot[i][j];
            k++;
        }
    }
    k = 0;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < p; j++) {
            int ij = (j - 1) * n + i;
            (*mat)[i][j] = fill[k];
            *(m2 + ij) = (*mat)[i][j];
            k++;
        }
    }
    k = 0;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < p; j++) {
            int ij = (j - 1) * n + i;
            *(mot + j)[i] = fill[k];
            *(m3 + ij) = *(mot + j)[i];
            k++;
        }
    }
}
```

```
}
}
```

Test the `riddle()` function:

```
h <-
```

```
.C(
  'riddle',
  n = as.integer(4),
  p = as.integer(2),
  m1 = as.double(1:8),
  m2 = as.double(1:8),
  m3 = as.double(1:8)
)
```

```
## sizeof *mot 32 sizeof mot 8
## sizeof *mat 64 sizeof mat 8
## sizeof (mat + j) 8 sizeof *(mot + j) 32 sizeof *(mat + j) 64
## sizeof (mat + j) 8 sizeof *(mot + j) 32 sizeof *(mat + j) 64
```

```
print(matrix(h$m1, 4, 2))
```

```
##      [,1] [,2]
## [1,] 1    5
## [2,] 2    6
## [3,] 3    7
## [4,] 4    8
```

```
print(matrix(h$m2, 4, 2))
```

```
##      [,1] [,2]
## [1,] 1    5
## [2,] 2    6
## [3,] 3    7
## [4,] 4    8
```

```
print(matrix(h$m3, 4, 2))
```

```
##      [,1] [,2]
## [1,] 1    5
## [2,] 2    6
## [3,] 3    7
## [4,] 4    8
```