

PRINCIPIO DE COMPUTADORES.

PRÁCTICA 4

Descripción.

Te proponemos realizar en ensamblador un programa que dada una matriz definida en memoria te permita invertir el orden de los elementos de una fila o una columna que el usuario pueda meter por el teclado. Si tienes dificultad para hacer el código, o bien ara que puedas comprobar el funcionamiento, te dejamos un código en C++ (matrices.cpp) que hace lo que te pedimos hacer en MIPS.

```
1  #include <iostream>
2  int main(void) {
3      const int nrows = 4, ncols = 3;
4      int matrix[nrows][ncols] = {{11,12,13},{21,22,23},{31,32,33},{41,42,43}};
5
6      std::cout << "Practica 4. Trabajando con Matrices\n";
7      int selection;
8      do {
9          int i,j;
10         for (i = 0; i < nrows ; i++) {
11             for (j = 0; j < ncols ; j++)
12                 std::cout << matrix[i][j] << " ";
13             std::cout << std::endl;
14         }
15
16         do {
17             std::cout << "Elija opcion <0> Salir, <1> invertir fila, <2> invertir columna: ";
18             std::cin >> selection;
19         } while (selection < 0 || selection > 2);
20
21         if (selection != 0) {
22             if (selection == 1) {
23                 int f,aux;
24                 do {
25                     std::cout << "Seleccione fila [" << 1 << "," << nrows << "]: ";
26                     std::cin >> f;
27                 } while (f < 1 || f > nrows);
28                 f--;
29                 for (j = 0; j <= (ncols-1) / 2; j++) {
30                     aux = matrix[f][j];
31                     matrix[f][j] = matrix[f][ncols-1-j];
32                     matrix[f][ncols-1-j] = aux;
33                 }
34             } else { // selectiones 2
35                 int c,aux;
36                 do {
37                     std::cout << "Seleccione columna [" << 1 << "," << ncols << "]: ";
38                     std::cin >> c;
39                 } while ( c < 1 || c > ncols);
40                 c--;
41                 for (i = 0; i <= (nrows-1) / 2; i++) {
42                     aux = matrix[i][c];
43                     matrix[i][c] = matrix[nrows-1-i][c];
44                     matrix[nrows-1-i][c] = aux;
45                 }
46             }
47         }
48     } while (selection != 0);
49     return(0);
50 }
```

La ejecución debe ser algo similar a lo que se muestra a continuación:

```
Practica 4. Trabajando con Matrices.

11 12 13
21 22 23
31 32 33
41 42 43

Elija opcion <0> Salir, <1> invertir fila, <2> invertir columna: 1

Seleccione fila [1,4] = 3
11 12 13
21 22 23
33 32 31
41 42 43

Elija opcion <0> Salir, <1> invertir fila, <2> invertir columna: 2

Seleccione columna [1,3] = 1
41 12 13
33 22 23
21 32 31
11 42 43

Elija opcion <0> Salir, <1> invertir fila, <2> invertir columna: 0
```

Siendo la definición de la matriz en el segmento de datos del programa la siguiente:

```
size = 4 # bytes que ocupa cada elemento
.data
matrix: .word 11, 12, 13
        .word 21, 22, 23
        .word 31, 32, 33
        .word 41, 42, 43

nrows: .word 4
ncols: .word 3
```

La evaluación de esta práctica será realizada por los profesores de prácticas, y se resolverá según los siguientes criterios:

- 1) Para aprobar la práctica con un 5, el código debe ejecutarse sin errores, y como mínimo debe imprimir por la pantalla la matriz que esté cargada en memoria y ser capaz de invertir la fila que el usuario seleccione por la pantalla. Es imprescindible además que en la corrección el alumno o alumna demuestre que conoce el código y pueda implementar algún cambio que proponga el profesor.
- 2) Para sacar hasta un 8, el código debe ejecutarse sin errores, y como mínimo debe hacer lo anteriormente expuesto, pero además ser capaz de invertir la columna que el usuario seleccione por pantalla. El programa además debe ser robusto y asegurar que los valores que introduce el usuario son coherentes. Además, debe defenderse correctamente el código y poder implementar algún cambio que proponga el profesor.
- 3) Para sacar hasta un 10, el código debe incluir lo anteriormente expuesto y además permitir al usuario introducir la matriz por teclado, pudiendo el usuario seleccionar el número de filas y columnas que tiene la matriz, y almacenándola en el segmento de datos (que previamente debe haber reservado un tamaño máximo mediante la directiva `.space`). Además, debe defenderse correctamente el código y poder implementar algún cambio que proponga el profesor.