# Microprocessor-Based Systems Laboratory

## Lab Session 3: Development of programs using C and 80x86 assembler

In this session the students have to implement a program (pract3.c) in C to call to some functions that will be developed in assembler and located in a different file (pract3a.asm). The requirements for each function have to be analyzed in order to be implemented correctly.

### Module 1: pract3a.asm

This module requires the codification of three math and text functions. These functions shall be called from the program written in C according to the following prototypes:

1.- int calculaMediana(int a, int b, int c, int d) **(3 pts)**

Returns the median of a, b, c and d (integer 16-bit numbers, signed).

2.- void enteroACadenaHexa (int num, char* outStr) **(4 pts)**

This function receives a **signed** integer number "num" and returns a pointer to a character string "outStr". The string outStr (null-terminated) will contain the ASCII characters corresponding to the value of num in hexadecimal format. For example, if num = 1020 (3FCh), the string outStr will contain the following 4 bytes: 33h (ASCII code for 3), 46h (ASCII code for F), 43h (ASCII code for C), 00h (termination).

3.- void calculaLetraDNI(char* inStr, char* letra) **(3 ptos)**

This function receives a pointer to the string inStr, and returns a pointer to a byte letter. The byte letter will contain the ASCII code for the letter corresponding to the DNI represented as a character string inStr (8 ASCII characters, for example: "12345678"). For example, if inStr equals "12345678", the byte letter will be equal to 5A (corresponding to the character 'Z').

To verify the NIF letter, the control digit is divided by 23 and the rest is replaced by a letter according the following table:

| RESTO | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LETRA | T | R | W | A | G | M | Y | F | P | D | X | B |

| RESTO | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| LETRA | N | J | Z | S | Q | V | H | L | C | K | E |

For instance, if the DNI number is 12345678, this number divided by 23 results with a rest of 14. Therefore, the assigned letter is "Z"

Notes:

- As the compiler precedes all extern references with the character '_', It is necessary that all the assembler functions start with this character (example: _calculaMediana)
- All the assembler functions have to be declared as PUBLIC in order to be used from the C program
- Each assembler module will exclusively contain a segment code, starting with the following 2 lines:

```
PUBLIC <module name> SEGMENT BYTE PUBLIC 'CODE'
       ASSUME CS: <module name>
```

- The C program (pract3.c) has to be compiled in LARGE model. The assembler functions will be declared as FAR.
- The C program (pract3.c) will ask the user for different input values in order to test each assembler function. The values returned by the different functions will be printed to screen in the adequate format for its correct understanding.
- Strings in C language should be always null-terminated.

**DELIVERY: Date and contents**

Upload to Moodle a ZIP file containing the source files (only the .asm) of the exercises and the makefile. Only one member of the team may upload the file. The files shall contain the authors' name and the team number in the header.

The source files shall be correctly tabulated and commented. The lack of comments or poor quality comments will be qualified negatively.

**The date limit to upload the files for Friday groups is April 16th at 23:55h**

**Annex**: Compilation of a project containing C and assembler files

In this practice we will have three different source files, one main program written in C (pract3.c) and one file containing the functions written in assembler (pract3a.asm).

The TurboC (tcc) compiler will be used to compile C files. The different options provided by TurboC can be shown by typing "tcc" at the DosBox prompt. The assembler modules will be assembled as in the previous practices.

The C program shall be compiled with the option –ml (memory model large), whereas the assembler modules shall be assembled with the option /ml (case sensitivity on all symbols).

In order to generate the executable file "pract3.exe" the following makefile can be used (including tabulations):

```
all: pract3.exe

pract3.exe: pract3.obj pract3a.obj

	tcc -v -ml -Lc:\compila\tc\lib pract3.obj pract3a.obj

pract3.obj: pract3.c

	tcc -c -v -ml -Ic:\compila\tc\include pract3.c

pract3a.obj: pract3a.asm

	tasm /zi /ml pract3a,,pract3a
```