

# Computer Architectures

## Lab 4

The owner of a shop is restocking shelves. He made a list with items to buy; the items are identified by means of a code. He wants to know the total amount of the expense according to the price list of his wholesaler. Write a program in ARM assembly language to compute the amount of the expense according to the following details.

The price list of the wholesaler is a table where each entry consists of two integer values: the identification code of the product (4 bytes) and the price (4 bytes). The table is ordered according to the identification code. For example:

```
Price_list DCD 0x004, 20, 0x006, 15, 0x007, 10, 0x00A, 5, 0x010, 8
           DCD 0x012, 7, 0x016, 22, 0x017, 17, 0x018, 38, 0x01A, 22
           DCD 0x01B, 34, 0x01E, 11, 0x022, 3, 0x023, 9, 0x025, 40
           DCD 0x027, 12, 0x028, 11, 0x02C, 45, 0x02D, 10, 0x031, 40
           DCD 0x033, 45, 0x035, 9, 0x036, 11, 0x039, 12, 0x03C, 19
           DCD 0x03E, 1, 0x041, 20, 0x042, 30, 0x045, 12, 0x047, 7
```

The list of items to buy is another table with two integer values for each entry: the identification of the product (4 bytes) and the desired quantity (4 bytes). The table is not ordered. For example:

```
Item_list DCD 0x022, 4, 0x006, 1, 0x03E, 10, 0x017, 2
```

A binary search should be implemented for searching the price of the items in the `Price_list` table. The C code of the binary search is:

```
first = 0
last = num_entries - 1;
index = 0;
while (first <= last)
{
    middle = (first + last) / 2;
    if (key == table[middle])
    {
        /* element found */
        index = middle;
        break;
    }
    else
        if (key < table[middle])
            last = middle - 1;
        else
            first = middle + 1;
}
```

At the end of the program, register `r10` should contain the amount of the expense if all products are present in the price list, or 0 if at least one product is missing in the price list.

**Extension.** Suppose that the entries in the price list are not sorted. Define a suitable block of memory in a data area with the `space` directive. Use it to sort `Price_list` by means of bubble sort. The algorithm can be described in C code as follows:

```
last = num;
while (last > 0)
{
    pairs = last - 1; last = 0
    for (j = 0; j <= pairs; j++)
    {
        if (entry[j] > entry[j + 1])
        {
            temp = entry[j];
            entry[j] = entry[j + 1];
            entry[j + 1] = temp;
            last = j;
        }
    }
}
```

Remember that the sort routine must sort both the key and the data associated with each entry.