



1. Introdução

Pretende-se com este módulo que os alunos se familiarizem com o código *assembly* gerado pelo compilador a partir de um programa escrito numa linguagem de alto nível.

Para atingir este objectivo o aluno deve desenvolver um pequeno programa em C, compilá-lo utilizando o `gcc` e identificar os segmentos de *assembly* que correspondem a cada instrução em C.

2. Linguagem de alto nível

Escreva em C, usando o editor de texto que considerar mais adequado, o programa apresentado na tabela 1.

main.c	
<pre>#include <stdio.h> int a, b, c; void imprime (void) { printf ("a=%d\tb=%d\tc=%d\n", a,b,c); }</pre>	<pre>main () { b = 10; a = b * 2; c = a - b; imprime (); }</pre>

Compile o programa `main.c` usando o comando

```
gcc -S main.c
```

Visualize o *assembly* gerado pelo compilador (`main.s`) e complete as seguintes questões:

Questão 1 – Identifique os 4 conjuntos de instruções *assembly* que correspondem a cada uma das instruções em C.

Questão 2 – Qual a instrução *assembly* utilizada para invocar funções? E quais são as instruções utilizadas para terminar uma função?

Questão 3 – Como é que o *assembly* referencia cada uma das variáveis globais `a`, `b` e `c`?

Questão 4 – Para realizar as operações aritméticas o compilador gerou instruções que copiam o valor das variáveis de memória para registos. Identifique os registos associados a cada variável em cada uma das operações aritméticas.

Questão 5 – Modifique o *assembly* de forma a que: $a = b * 8$ e $c = b - a$.

Questão 6 – A partir do ficheiro `main.s` modificado crie o executável `main` e verifique a correcção das suas alterações. Para tal use o comando:

```
gcc -o main main.s
```

Questão 7 – O comando invocado na questão anterior envolve vários passos do processo de conversão entre níveis de abstracção. Identifique-os e caracterize-os.

Questão 8 – A função `imprime()` invoca a função `printf()`, passando-lhe 4 parâmetros. Qual é o mecanismo utilizado para passar estes parâmetros?