

Considere as seguintes variáveis globais relativas à técnica para determinar o resto da divisão de um número por 9, tradicionalmente denominada por "noves fora".

```
.section .data
numero_atual: .short 0
```

Esta técnica aplica-se da seguinte forma:

1. Somam-se os dígitos do número (para extrair os dígitos, divide-se sucessivamente o número por 10 e os dígitos serão os sucessivos restos);
2. Verifica-se se a soma dos dígitos é superior a 9;
3. Se sim, somam-se os dígitos que compõem a soma anterior e verifica-se novamente se a soma atual é superior a 9;
4. O processo termina quando a soma dos dígitos da última soma for inferior ou igual a 9;
5. O resto da divisão será: (i) o valor da última soma, caso esta seja inferior a 9; (ii) zero, caso a última soma seja igual a 9 (daqui resulta o nome "noves fora").

Por exemplo, para determinar o resto da divisão de 987654329 por 9 somam-se os seus dígitos, obtendo-se 53. Como 53 ainda é maior do que 9, somam-se os seus dígitos, obtendo-se 8. Conclui-se assim que o número já é menor que 9 e que o resto desejado é 8. Assim, 987654329 -> 53 -> 8.

1. Implemente em C a função **void noves_fora(unsigned int *numeros, int n)**. A função recebe no parâmetro **números** um vetor com os **n** números a considerar. Considere que em cada elemento do vetor **numeros** está armazenado o número a dividir por 9 nos 16 bits mais significativos e que pretendemos armazenar o resto da divisão do número nos 8 bits menos significativos, de acordo com o seguinte esquema:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Número que se pretende dividir por 9																Não usado								Resto							

A função deverá usar aritmética de apontadores para percorrer o vetor, para obter cada um dos números a dividir por 9 e para preencher o byte menos significativo de cada elemento do vetor com o resto da divisão. Considere que esse resto é obtido através da invocação da função **calcula_resto()** desenvolvida em Assembly (e descrita no ponto seguinte). Antes de invocar a função deve atualizar a variável global **numero_atual**.

Esta função deve ser desenvolvida no ficheiro **noves_fora.c**

2. Implemente em Assembly a função **unsigned char calcula_resto()**. A função deve, dado o valor atual da variável global **numero_atual**, determinar e retornar o resto da divisão de um número por 9 utilizando a técnica descrita acima.

Esta função deve ser desenvolvida no ficheiro **calcula_resto.s**

3. Implemente o ficheiro principal da aplicação ('**main.c**') em linguagem C. Para efeitos do exercício, considere que existem 10 números com os seguintes valores (não necessita escrever o código para as leituras):

```
unsigned int numeros [10] = {0x0B0F0000, 0x0C0A0000, 0x0B080000, 0x07100000, 0x0F120000,
0xF0E10000, 0x0B0F0000, 0x0A0B0000, 0x0F100000, 0x090F0000}
```

O seu programa deve:

- Definir corretamente as variáveis necessárias para os dados apresentados;
- Invocar a função **noves_fora(...)**;
- Apresentar no ecrã os números para os quais se pretende obter o resto da divisão por 9 e o resto calculado através do "noves fora".

Escreva um Makefile para construir o executável na qual exista uma regra específica para cada fase de construção (compilação; *assemblagem* e *linkagem*) e para cada ficheiro!

Notas importantes

- ⚠ Não deve alterar o cabeçalho das funções
- ⚠ Serão penalizadas as seguintes situações: Código que não compila; Código com warnings; Código não indentado; Código sem comentários adequados
- ⚠ **No final deve criar um zip com todos os ficheiros da sua solução, com o nome "<turma>_<numero do aluno1>_<numero do aluno 2>.zip" e deve submetê-lo usando o link que foi criado no moodle para esse efeito.**
- ⚠ **Será considerada como fraude qualquer tipo de troca de dados/informação com terceiros durante a aula sem a autorização do docente.**