



## 1. Introdução

Pretende-se com esta aula prática que os alunos entendam o código gerado por um compilador de C para um programa com variáveis estruturadas e ciclos, utilizando diferentes níveis de optimização. Para atingir este objectivo o aluno deve desenvolver um pequeno programa em C, compilá-lo e interpretar o código gerado.

## 2. Linguagem de alto nível

Escreva em C, usando o editor de texto que considerar mais adequado, o seguinte programa:

```
prog.c
#include <stdio.h>

typedef int mytype;

mytype array[100];
int sum, i;

void ini (void)
{
    for (i=0 ; i<100 ; i++)
        array[i] = i;
}

main ()
{
    ini ();

    sum = 0;
    for (i=0 ; i<100 ; i++)
        sum += array[i];

    printf ("Sum=%d\n", sum);
}
```

## 3. Compilação sem optimização

Compile o programa `prog.c` usando o comando

```
gcc -S prog.c
```

Analisando o código *assembly* da função `main()`, responda às seguintes questões:

**Questão 1** – Identifique e explique as instruções responsáveis pelo teste do `for (...)`.

**Questão 2** – Identifique e explique as instruções responsáveis pelo cálculo do endereço de `array[i]`.

## 4. Compilação com optimização

Compile o programa `prog.c` usando o comando

```
gcc -O1 -S prog.c
```

Analisando o código *assembly* da função `main()`, responda às seguintes questões:

**Questão 3** – Identifique e explique as instruções responsáveis pelo teste do `for (...)`. Compare com a resposta à questão 1.

**Questão 4** – Identifique e explique as instruções responsáveis pelo cálculo do endereço de `array[i]`. Compare com a resposta à questão 2.

## 5. Linguagem de alto nível

Modifique o programa anterior, conforme apresentado na seguinte tabela:

<code>prog.c</code>	
<pre>#include &lt;stdio.h&gt;  typedef struct {     char s[4];     int a; } mytype;  mytype array[100]; int sum, i;  void ini (void) {     for (i=1 ; i&lt;100 ; i++)     {         array[i].s[0] = '\0';         array[i].a = i; } }</pre>	<pre>main () {     ini ();      sum = 0;     for (i=0 ; i&lt;100 ; i++)         sum += array[i].a;      printf ("Sum=%d\n", sum); }</pre>

## 6. Compilação sem optimização

Compile o programa `prog.c` usando o comando

```
gcc -S prog.c
```

Analisando o código *assembly* da função `main()`, responda às seguintes questões:

**Questão 5** – Identifique e explique as instruções responsáveis pelo cálculo do endereço de `array[i].a`. Compare com a resposta à questão 2.

**Questão 6** – Modifique no código em C o tamanho do campo `s` da estrutura para 8 caracteres. Identifique e explique as instruções responsáveis pelo cálculo do endereço de `array[i].a`. Compare com a resposta à questão 5.