



## 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 otimização. Para atingir este objectivo o aluno deve desenvolver um pequeno programa em C, compilá-lo e interpretar o código gerado.

---

**NOTA:** A QUESTÃO ASSINALADA COM **TPC** DEVE SER RESOLVIDA **ANTES DA SESSÃO TP** E ENTREGUE AO DOCENTE NO INÍCIO DA MESMA. A FOLHA ANEXA A ESTE MÓDULO, ASSINALADA COMO TPC, DESTINA-SE A ESSE FIM.

---

## 2. Linguagem de alto nível

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

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

## 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 (...)`.

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

## 4. Compilação com otimização

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

## 6. Compilação sem otimização

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.

<b>TPC</b>	
Número:	Nome:

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

Questão resolvida na aula	
---------------------------	--

Número:	Nome:
---------	-------