

6.4 Exercícios

6.1 Para cada uma das expressões C a seguir, para as quais todas as variáveis são do tipo inteiro de quatro bytes:

```
x = ( a + b + c ) * ( b + c ) - d;  
r = a * b + pow(c, 2);  
b[i] = a[i] + a[i-1]
```

- (a) Escreva o bloco de instruções em código intermediário de três endereços.
 - (b) Para cada bloco, escreva um bloco equivalente otimizado. Indique claramente quais técnicas de otimização foram utilizadas.
 - (c) Apresente a representação do código otimizado em notação de quádruplas para cada expressão.
 - (d) Repita o item anterior usando a notação de triplas.
- 6.2 Para o código otimizado gerado para cada expressão na questão anterior, apresente o código em linguagem simbólica gerado para um processador com instruções aritméticas ADD, SUB e MUL e com arquitetura de
- (a) três endereços;
 - (b) dois endereços;
 - (c) um endereço;
 - (d) zero endereços.
- 6.3 Para cada uma das expressões aritméticas a seguir, apresente a representação equivalente em notação pós-fixa:
- (a) $(a + b + c) \times (b + c) - d$
 - (b) $a - (b - a \times (c + b/d))$
 - (c) $(a + b) - (a - (c - d) \times (e - f) + g)/h$
- 6.4 Apresente o código em formato intermediário de três endereços para o seguinte fragmento de código C:
- ```
i = 0; /* decl: int i */
while (i < 10) {
 a[i] = 0; /* decl: int a[10] */
 i = i + 1;
}
```
- 6.5 Para o seguinte fragmento de código C, com as variáveis do tipo int (quatro bytes):

```
for (i = 0; i < 100; ++i) {
 eps = 0.001;
 if (a[i] != 0)
 c[i] = b[i] / a[i];
 else
 c[i] = eps;
}
```

- (a) Apresente o código em formato intermediário de três endereços, sem otimização.
- (b) Otimize o código em formato intermediário, indicando claramente quais foram as estratégias utilizadas.