

1. Considere que a arquitetura Y86 suporta uma instrução *iaddl C, rB*, a qual adiciona uma constante *C* ao valor armazenado no registo especificado pelo campo *rB*. Indique quais os sinais de controlo relevantes da organização sequencial do Y86 ativos em cada fase de execução da instrução.

2. Considere o seguinte programa em Y86. Identifique para cada ciclo do relógio a ocupação de cada estágio do processador, para a versão PIPE do Y86. Justifique a sua resposta indicando quais os valores encaminhados.

```
i1:    irmovl $10,%ecx
i2:    xorl %eax, %eax
i3:    jne i6
i4:    pushl %eax
i5:    halt
i6:    pushl %ecx
i7:    halt
```

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Justificação

Nome: \_\_\_\_\_ Número: \_\_\_\_\_

3. Considere o seguinte extrato de programa escrito em Y86:

```
I1:    mrmovl 0(%ecx), %edx
I2:    addl %edx, %eax
I3:    rmmovl %eax, 1000(%ecx)
I4:    addl %esi, %ecx
I5:    mrmovl 0(%ecx), %edx
I6:    addl %edx, %eax
I7:    rmmovl %eax, 1000(%ecx)
```

3.1 Identifique as dependências de dados existentes neste programa. Indique quais dessas dependências originam bolhas na arquitetura Y86 PIPE e explique como é que o escalonamento dinâmico e a renomeação de registos permitem remover essas bolhas.

3.2 Apresente o escalonamento dinâmico deste programa (depois de remoção de dependências através da renomeação de registos), numa arquitetura superescalar de duas vias, com a capacidade de executar uma operação aritmética/salto simultaneamente com um acesso à memória. Apresente apenas a fase de execução de cada instrução e considere que a latência de ambas as unidades funcionais corresponde a um só ciclo de execução.

Ciclo	Aritmética/salto	Acesso à memória
1		
2		
3		
4		
5		
6		
7		

Nome: \_\_\_\_\_ Número: \_\_\_\_\_

4. Considere o seguinte programa em *pseudo* X86.

```
ciclo:  mov 0(%esi), %eax
        add %eax, $1000(%esi)
        add 4, %esi
        sub 1, %ecx
        jnz ciclo
```

4.1 Explique em que consiste a optimização designada por *desdobramento de ciclos* e calcule, para este programa, os ganhos obtidos com essa optimização, considerando um desdobramento de 4 vezes e que o valor inicial de %ecx é múltiplo de 4. Para este exercício poderá assumir uma arquitetura com CPI igual a 1 para todas as instruções.

4.2 Indique quais as principais vantagens da utilização do processamento vectorial e indique o ganho potencial da utilização de instruções AVX neste programa, recentemente introduzidas nas arquiteturas Intel. A extensão AVX introduziu suporte a instruções sobre registos vectoriais de 256 bits.