1. Considere o seguinte programa escrito em *assembly* do Y86:

```
        movl $100,%ecx
ciclo:  movl 0(%eax), %edx
        addl %ecx, %edx
        movl %edx, 1000(%eax)
        subl 4, %ecx
        jnz ciclo
```

Classe	CPI
Aritméticas/movimento de dados	1
Acessos à memória	3
Controlo de fluxo	2

Calcule o tempo de execução do programa num processador com uma frequência de relógio de 2GHz, considerando o CPI de cada uma das classes de instruções indicado na tabela.

2. Pretende-se escolher o melhor processador para executar um dado programa. A opção é entre um processador base e um processador semelhante (i.é., com o mesmo CPIcpu, igual a 1) mas que executa a metade da frequência e possui o dobro da *cache*. Indique, justificando, qual será a melhor opção. Considere que o código desse programa cabe na *cache* de ambos os processadores, que o programa que se pretende executar apresenta, no processador base, o dobro do *miss rate* de dados, que o *miss rate* de dados no processador base é de 10%, que a *miss penalty* é de 100 ciclos no processador base e que o programa apresenta 50% de acessos à memória.

Nome: _____ Número: _____

3. Considere a seguinte função em C.

```
int func(int h[][256], int H) {
    int soma;
    for (int y=0 ; y<H ; y++) {
            for (int x=0 ; x<256 ; x++) {
                    soma += h[y][x];
            }
    }
    return(soma);
}
```

Indique, justificando, se esta função apresenta localidade espacial e/ou localidade temporal.

4. Considere que a arquitetura Y86 suporta a instrução *ret [rA]*, semelhante à instrução de *ret* mas o endereço de retorno é indicado pela posição de memória [rA], em vez do tradicional endereço armazenado na pilha (i.e., [esp]). Indique quais os sinais de controlo relevantes da organização sequencial do Y86 ativos em cada fase de execução desta instrução.

Nome: _____ Número: _____

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

```
I1:     mrmovl 0(%ecx), %edx
I2:     addl %edx, %eax
I3:     mrmovl 4(%ecx), %edx
I4:     addl %edx, %eax
I5:     addl 8, %ecx
```

5.1 Identifique as dependências de dados existentes neste programa (sugestão: apresente essas dependências na forma de um grafo).

5.2 Indique quais as dependências que podem ser removidas através da renomeação de registos. Explique, também, de que forma o escalonamento dinâmico pode remover bolhas que surgem ao executar este programa na arquitetura Y86 PIPE.

Nome: _____ Número: _____

6. 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 também quais os valores encaminhados.

```
i1:     irmovl $10,%ecx
i2:     irmovl $20,%eax
i3:     call i5
i4:     halt
i5:     irmovl $20,%ecx
i6:     addl %eax,%ecx
i7:     ret
```

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

Nome: _____ Número: _____