Universidade Católica Dom Bosco
Curso de Engenharia de Computação
Disciplina: Compiladores II
Professor: Ricardo Santos
RA: **Nome:** **Data:**

2ª. Avaliação

1) (valor 1,5)Assinale a alternativa correta (V ou F)

( ) O algoritmo de geração de código baseado em programação dinâmica deve sempre calcular o vetor de custos de uma árvore das folhas para a raíz pois, um vértice raíz de uma sub-árvore S necessita dos resultados de seus descendentes para calcular seu vetor de custos.

( ) A otimização "Redução em força" possibilita transformar uma expressão do tipo: $x = y + 0$ em $x = y$.

( ) No algoritmo de Alcance de Definições ("Reaching Definitions"), o conjunto out para um bloco básico B pode ser definido como: $out_B = gen_B \bigcup (in_B - kill_B)$.

( ) Um bloco básico pode ser definido como um agrupamento de instruções de um programa onde o controle "entra" somente através da 1a. instrução do bloco e "sai" apenas pela última instrução.

( ) A execução das otimizações Eliminação de Subexpressões Comuns (CSE), Eliminação de Código Morto (DCE) e Propagação de Cópias terá o mesmo resultado final independente da ordem de execução.

( ) A aplicação do algoritmo global de alocação de registradores sobre uma árvore de expressão sempre resultará na utilização de, no máximo, dois registradores (considerando um conjunto de instruções (ISA) MIPS-like).

( ) Blocos Básicos (BBs) podem ser representados como uma lista encadeada de quadruplas. A primeira quadrupla tem uma flag especial tal que aponta para a quadrupla da instrução líder, um campo que aponta para o BB predecessor e outro campo que aponta para o BB sucessor. As quadruplas que representam as instruções do BB não necessitam de nenhuma alteração em sua estrutura.

2) (valor 1,5)Mostre o GFC (CFG) para o código a seguir. Mostre as instruções que estão dentro de cada bloco básico.

```
L0:
  i=i+1
  t1=4*i
  if i<n goto L1
  goto L2
L1:
  j=j-1
  t2=4*j
  if j<m goto L3
  goto L4
L3:
  a[t2]=8*a[t2]
  goto L0
L2:
  a[t2]=a[t2]-8
  goto L0
L4:
  a[t1]=a[t1]-8
  goto L0
```

3) (valor 2,5)Utilize os conceitos de redução em força e eliminação de variáveis de indução sobre a código a seguir. Mostre o código final e justifique por que a otimização pode (ou não) ser realizada sobre o código. Procure demonstrar cada transformação realizada.

```
s=0
i=0
if i<n goto L2
L1:
  j=4*i
  k=j+a
  x=M[k]
  s=s+x
  i=i+1
  goto L1
L2:
```

4) (valor 2,5)Considere o programa P a seguir e utilize o algoritmo de list scheduling para definir um escalonamento com menor quantidade de ciclos possível. Considere que as operações de multiplicação consomem 3 ciclos, deslocamentos possuem 2 ciclos e as demais operações 1 ciclo. Informe o conteúdo o Delay de cada operação, e o conteúdo dos conjuntos Cand, MCands, ECand e Sched após cada passo do algoritmo.

```
f=2*a
p=e+2
g=b*c
j=f*g
m=j-f
i=p<<2
t=m<<i
```

5) (valor 2,0)Forneça uma breve descrição sobre a otimização/análise de fluxo de dados que você está implementando no Trabalho 2 da disciplina de Compiladores II. Além disso, apresente o algoritmo (utilizando pseudo-código) e forneça um exemplo (em código de 3 end) onde seu algoritmo consegue realizar a otimização/análise de fluxo de dados.