UNIVERSIDADE FEDERAL DO MARANHÃO		Departamento de Informática - DEINF		1ra PROVA	
				P	
Disciplina: Teoria da Computação		Curso: CIÊNCIA DA COMPUTAÇÃO		T	1,0
Código 5607.5	Carga Horária: 60 horas		Créditos: 4.0.0	NOTA	
Professor: Luciano Reis Coutinho		Email: luciano.rc@ufma.br			

1a Avaliaç[redacted] **Data: 15 abril de 2024**

Aluno : ______ **Código:** ______________

INSTRUÇÕES

- Cada questão consiste em um enunciado e um conjunto de requisitos. Respostas dadas que não atendam aos requisitos podem em última instância ser completamente desconsideradas durante a correção da prova.
- A interpretação das questões faz parte da avaliação. Caso ache um enunciado ambíguo ou impreciso escreva na folha de resposta sua interpretação e a correspondente resposta. Todas as questões devem ser interpretadas tendo em vista que foi discutido nas aulas de Teoria da Computação.
- **O tempo total de prova é de 100 min. Início**: 14:00, **término**: 15:40.

QUESTÕES

1. Considere o programa em C abaixo (programa iterativo).

```
#include <stdio.h>
int main ()
{
    int n, i=1, f=1;

    printf ("enter no=");
    scanf ("%d", &n);
    while (i<=n)
    {
        f=f*i;
        i++;
    }
    printf ("factorial=%d",f);
    return 0;
}
```

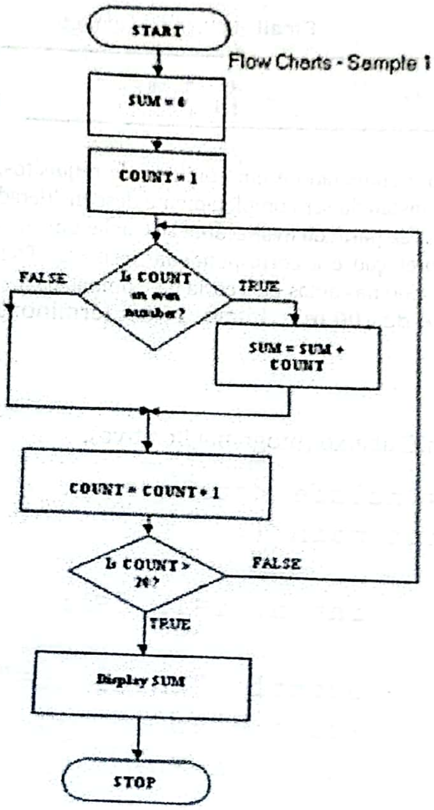
(a) **(1,0 ponto)** Defina uma máquina de registradores $M = \langle V, X, Y, pi\_x, pi\_y, PI\_f, PI\_t \rangle$ na qual o programa pode ser interpretado. Ou seja, defina estrutura de memória (V = registradores), conjunto de entrada (X), conjunto de saída (Y), função de entrada (pi_x), função de saída (pi_y), conjunto de operações (PI_f), e conjunto de testes (PI_t), necessários para a execução do programa.

(b) **(1,0 ponto)** Represente o programa como um programa monolítico (fluxograma ou instruções rotuladas conforme discutidas em sala de aula) para a máquina definida na letra (a).

2. **(1,0 ponto)** Sejam P e Q programas e M e N máquinas. Marque a resposta CORRETA:
(a) M é equivalente a N se, somente se, $\exists P \; \exists Q \; (\langle P,M\rangle = \langle Q, N\rangle)$.
(b) P e Q são equivalentes fortemente se, somente se, $\forall M \; \exists N \; (\langle P,M\rangle = \langle Q, N\rangle)$.
(c) P e Q são equivalentes fortemente se, somente se, $\exists M \; \exists N \; (\langle P,M\rangle = \langle Q, N\rangle)$.
(d) N simula fortemente M se, somente se, $\exists P \; \exists Q (\langle P,M\rangle = \langle Q, N\rangle)$.
(e) N simula fortemente M se, somente se, $\forall P \; \exists Q \; (\langle P,M\rangle = \langle Q, N\rangle)$.

3. Tendo em vista as definições de **monolíticos** e **recursivos** e a definição de **equivalência forte** entre programas que foram apresentadas durante as aulas:

(a) **(1,0 ponto)** represente o fluxograma abaixo na forma de instruções rotuladas.



(b) **(1,0 ponto)** traduza o programa da letra (a) para um programa recursivo equivalente fortemente.

4. **(1,0 ponto)** Dados o programa abaixo, e a **máquina de dois registradores** discutidas em sala de aula, pergunta-se: Qual a função computada pelo programa Q abaixo quando o teste T é interpretado como sendo `a_zero`, a operação F como sendo `sub_a` e G como sendo `ad_b` ? Escreva uma FÓRMULA que define a função e JUSTIFIQUE a sua resposta apresentando em no mínimo 5 linhas de texto baseado no assunto que foi estudado em sala de aula. Resposta sem justificativa válida será desconsiderada na correção.

Q: `até T faça (F; ( se T então √ senão G;F ) )`

5. **(2,0 pontos)** Utilizando o método discutido em sala de aula, verifique se os programas P1 e P2 a seguir são ou não são equivalentes fortemente. Lembrete do método: (1) transforme os programas para instruções rotuladas compostas; (2) identifique e simplificando ciclos infinitos; (3) construa a cadeia de conjuntos B0, B1, ..., Bk de rótulos equivalentes fortemente; caso Bk = {} os programas são equivalentes fortemente, caso contrário, não o são.

P1:

```
até T
        faça (√);
        enquato T
        faça (F; G; (se T
                então F; até T faça (√)
          senão √ )
  )
```

P2:

1: se T então va_para 2 senão va_para 1
2: faça F va_para 3
3: faça G va_para 4
4: se T então va_para 5 senão va_para 6
5: faça F va_para 1

6. **(2,0 pontos)** Considere a máquina `um_reg` definida abaixo:

```
um_reg = < N, N, N, id , id , {ad, sub}, {zero} >
```

sendo

```
id: N → N   , tal que  id(n)=n

ad: N → N   , tal que  ad(n)=n+1
sub: N → N, tal que sub(n)=n-1, se n ≠ 0;  sub(n)=0, se n=0

zero: N → {verdadeiro, falso}, tal que
          zero(0)=verdadeiro  e   zero(n)=falso, se  n ≠ 0.
```

Escreva um programa (de qualquer tipo) que compute a função **f: N → N, f(x) = x/2** na máquina `um_reg`.

Boa Sorte!