UNIVERSIDADE FEDERAL DO CEARÁ - CAMPUS RUSSAS
CIÊNCIA DA COMPUTAÇÃO/ENGENHARIA DE SOFTWARE

FERNANDA COSTA DE SOUSA

Instruções para a prova:
1 - Use caneta esferográfica de tinta azul ou preta tanto para marcar ou escrever as respostas, todas as respostas devem estar nas folhas da prova.
2 – O tempo disponível para realização desta prova será de 120 minutos.
3 – Não é permitido que o aluno se comunique com os demais estudantes nem troque material com eles ou consulte material bibliográfico, cadernos ou anotações de qualquer espécie. A desobediência dessa regra resultará em nota 0.
4 - O enunciado das questões contém todas as informações necessárias para respondê-las. A interpretação do enunciado faz parte da prova.
5 - As estrelas antes de cada questão representam a sua dificuldade de resolução. ⋆ são questões fáceis, ⋆ ⋆ são questões médias e ⋆ ⋆ ⋆ são questões difíceis.

1. ⋆ (1 Pontos - **Definições Básicas**) Marque com (X) as assertivas que são verdadeiras

   () 2/4 = 0.5 Falsa. A divisão é inteira!
   (X) 2 % 3 = 2
   (X) 3/10 = 0
   () x_y não é um identificador válido
   () 12x é um identificador válido Falsa. Começa por número
   () tensão é um identificador válido Falsa. Caractere especial.
   () V & F = V
   (X) V | F = V
   (X) (3 + 3)/2 = 3

2. ⋆ ⋆ ⋆ (2 Pontos - **Laços e Condicionais**)(*Canguru's are back, baby!*) O professor Alexandre, novamente em um café em Amsterdam comendo alguns bolinhos, avista agora não dois, mais três cangurus. O primeiro canguru pula 5cm para frente a cada salto, o segundo canguru pula 7cm para frente a cada salto e o terceiro canguru pula 9cm para frente. Os três cangurus, segundo o professor Alexandre, começaram juntos saltando em linha reta em uma estrada com flores a cada 1cm. No primeiro salto o primeiro canguru pisa na flor que está a 5cm do ponto de largada, o segundo na flor que está a 7cm e o terceiro a flor que esta a 9cm. Escreva um programa que determinar a distancia da primeira flor a ser pisada por todos os três cangurus.

```
Entrada:
Saída: Indice da primeira flor pisada pelos 3 cangurus
VAR[int] canguru1 := 5;
VAR[int] canguru2 := 7;
VAR[int] canguru3 := 9;
enquanto ((canguru1 != canguru2) | (canguru1 != canguru3) | (canguru2 != canguru3)) faça
    se canguru1 < canguru2 então
        se canguru1 < canguru3 então
            canguru1 := canguru1 + 5;
        senão
            canguru3 := canguru3 + 9;
    senão
        se canguru2 < canguru3 então
            canguru2 := canguru2 + 7;
        senão
            canguru3 := canguru3 + 9;
retorna canguru1;
```

3. ⋆⋆ (2 pontos - **Vetores e Matrizes**) Dado um vetor a de tamanho n de números distintos, dizemos que a representa uma ordem parcial se $a[i] \leq a[2i+1]$ e $a[i] \leq a[2i+2]$ para todo i tal que $2i+2 < n$. Escreva um programa que verifica se um vetor a de tamanho n dado de entrada representa uma ordem parcial

Entrada: int[n] a, int n
Saída: V se a é uma ordem parcial e F em caso contrário
`VAR[int]` i := 0 ;
enquanto $2i+2 < n$ **faça**
  se $((a[i] > a[2i+1]) \mid (a[i] > a[2i+2]))$ **então**
    retorna F;
retorna V;

4. ⋆⋆ (2 pontos - **Funções**) Escreva um programa que conta o número de divisores primos (números primos que dividem o número em questão) de um número inteiro positivo n dado de entrada.

Entrada: int n
Saída: Número de divisores primos do número em questão
`VAR[int]` i ;
`VAR[int]` cnt:=0 ;
para *i:= 1 ATE n* **faça**
  se $((primo(i))\&(n\%i == 0))$ **então**
    cnt := cnt + 1;
retorna *cnt*;

5. ⋆ (1 pontos - **Recursividade**) O professor Pablo, com inveja de Fibonacci pela sua famosa serie recursiva, resolveu definir ele mesmo uma serie recursiva, abaixo temos a serie de Pablonacci.

$$P(n) = \begin{cases} P(n-1) + 2.P(n-2) & \text{se } n \text{ é par} \\ 2.P(n-1) + P(n-2) & \text{se } n \text{ é ímpar} \\ P(2) = 3 & \\ P(1) = 2 & \end{cases}$$

Escreva uma função que implemente o cálculo da função recursiva de Pablonacci

```
int P(int n) {
    se n%2 == 0 então
        se n == 2 então
            retorna 3;
        senão
            retorna P(n-1) + 2*P(n-2);
    senão
        se n == 1 então
            retorna 2;
        senão
            retorna 2*P(n-1) + P(n-2);
}
```

6. ⋆ ⋆ ⋆ (2 pontos - **Recursividade**) O problema das torres de Hanoi é um dos problemas mais famosos em computação e um dos problemas mais usados para exemplificar a aplicação da ideia de recorrência. No problema original temos n discos de diâmetros distintos e 3 pinos. Os discos são colocados (ordenados por tamanho, do maior para o menor) no primeiro pino, chamado de origem e então devemos transporta-los para o terceiro pino, chamado de destino, utilizando o segundo pino como auxiliar, nunca colocando um disco de diâmetro maior sobre um disco de diâmetro menor. Uma variante deste problema, consiste em termos 2 pinos de origem. Como no problema clássico temos n discos de diâmetros distintos, entretanto, estes discos estão divididos em dois pinos de origem. Assim como no problema clássico, nosso objetivo é mover todos os discos dos dois pinos de origem para o pino de destino sem nunca colocar um disco de diâmetro maior sobre um disco de diâmetro menor. Observe que existe apenas UM pino de destino, assim as duas torres de discos devem ser combinadas. Descreva, com suas palavras, uma ideia de como podemos resolver esse problema.

Uma ideia para a solução é a seguinte: Determinamos em qual dos pinos está o disco de maior diâmetro, então movemos os discos que estão acima dele para o pino auxiliar utilizando o pino de destino como suporte usando o algoritmo utilizado para o problema das torres de Hanoi com apenas 3 pinos. Após removermos os discos que estão sobre o maior disco, podemos movê-lo para o pino de destino, deixando um dos pinos sem nenhum disco.

Veja que agora temos o mesmo problema, possuímos 2 pinos com os discos ordenados em cada um deles, um pino de destino que contém o maior disco e um pino sem nenhum disco que pode ser utilizado como auxiliar. Com isso, podemos aplicar a mesma ideia utilizada para mover o disco de maior diâmetro para mover o disco de maior diâmetro entre os discos que sobraram. Assim, basta mover os discos em ordem, sempre movendo o de maior diâmetro utilizando a sequencia de movimentos de Hanoi para 3 torres em cada execução