Universidade Federal de Santa Catarina - Campus Blumenau
Departamento de Engenharia de Controle, Automação e Computação **Avaliação 2 – 2ªchamada**
BLU3202 — **Algoritmos e Estruturas de Dados** 2022/1

Aluno(a): .....

1. Seja X uma pilha de caracteres. Seja $get(X)$ uma função cujo pseudocódigo é:

```
get(){
   // realiza a operação pop na pilha "X" e
   // armazena o elemento retirado na variável "c"
   pop(X,c);
   // imprime o elemento armazenado em "c"
   print(c);
}
```

   A combinação de operações *push* e *get* permite formar anagramas a partir de uma entrada. Por exemplo, a partir da sequência de caracteres A, R, T, S, pode-se obter o anagrama ARTS com a seguinte sequência de operações:

```
push(S);push(T);push(A);get();push(R);get();get();get();
```

   Considere a entrada, em sequência, dos caracteres A, B, C, D, E, F. Assinale os anagramas que podem ser produzidos a partir dessa entrada combinando operações *push* e *get*. Observação: cada escolha errada anula uma escolha certa.

( ) ACBFDE ( ) CDBEFA
( ) FEDCBA ( ) CFEBDA
( ) ABCDEF
( ) BDCFEA ( ) DBACEF

2. Considere a seguinte sequência de operações aplicadas simultaneamente tanto a uma fila quanto a uma pilha: `push(9)`, `push(5)`, `pop()`, `push(2)`, `push(7)`, `pop()`, `push(10)`, `push(8)`, `pop()`

   Nas alternativas abaixo, assinale V para as verdadeiras e F para as falsas

( ) Após a realização das operações, a soma dos elementos remanescentes na pilha será menor do que a soma dos elementos remanescentes na fila.

( ) Ao longo da execução das operações (incluindo a última), a soma dos elementos armazenados na fila nunca será menor do que a soma dos elementos armazenados na pilha.

( ) Ao longo da execução das operações (incluindo a última), a soma dos elementos armazenados na pilha nunca será menor do que a soma dos elementos armazenados na fila.

( ) A diferença entre a soma dos elementos presentes na pilha e na fila nunca ultrapassa 5.

3. Seja *push(n)* a operação que insere um número n em uma *pilha*. Considere a execução desta operação, com n assumindo os valores 1, 2, 3, 4 e 5, nesta ordem. Em meio às inserções, pode haver remoções (*pop*). Indique a ordem das operações *push(n)* e *pop()* para que a sequência de números retirados da pilha com a operação *pop* seja:

   (a) 2, 4, 3, 1, 5:
   _____

   _____

   _____

   (b) 1, 3, 2, 5, 4:
   _____

   _____

   _____

4. Considere as seguintes estruturas utilizadas para implementar uma fila:

```
typedef struct{
   int dado;
   struct item *next;
}item;
```

```
typedef struct{
   item *inicio;
   item *fim;
}fila;
```

   Complete o código abaixo para implementar a função *pop* em uma fila, assumindo que o parâmetro `*f` é um ponteiro para o primeiro elemento da estrutura.

```
1 int pop(fila *f){
2    int dado = f->inicio->dado;
3    ____________________;
4    if(f->inicio==NULL)
5       ____________________;
6    return dado;
7 }
```

   Linha 3: _____
   Linha 5: _____

5. Nas alternativas abaixo, assinale V para as verdadeiras e F para as falsas:

   ( ) A sondagem linear aplica-se somente a tabelas que em que a quantidade de itens a serem armazenados é menor ou igual ao tamanho da tabela.

   ( ) Para encontrar uma chave em uma tabela hash, é necessário utilizar técnicas semelhantes às utilizadas na busca por valores em vetores, tais como examinar cada um dos elementos até encontrar o valor procurado.

   ( ) A posição de uma chave em uma tabela hash depende apenas do seu valor. Ou seja, uma mesma chave terá o mesmo hash em qualquer tabela em que for inserida.

   ( ) No pior caso, a busca por uma chave pode ter custo computacional maior em tabelas em que as colisões são resolvidas por encadeamento em comparação à solução de colisões por sondagem linear.

   ( ) No pior caso, a inserção de uma chave tem custo computacional igual em tabelas em que as colisões são resolvidas por encadeamento em comparação à solução de colisões por sondagem linear.

6. Considere a inserção das seguintes chaves na dada ordem em uma tabela hash de tamanho 7: {**293, 39, 250, 25, 129, 266, 272**} usando o método da divisão com a função hash $\mathbf{h(x) = x \bmod 7}$. Mostre as tabelas resultantes resolvendo as colisões por encadeamento e por sondagem linear. Responda no seguinte formato: `0:[...], 1:[...], ...., n:[...]`

   _____

   _____

   _____

   _____

   _____

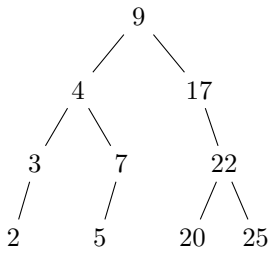
   _____

   _____

   _____

   _____

Para as questões 7 a 8, considere a árvore da figura abaixo:



7. Escreva abaixo a sequência de números que será impressa caso a árvore seja percorrida

   a) Em pré-ordem: _____

   b) Em ordem: _____

   c) Em pós-ordem: _____

8. Informe qual será a *soma* do valor das folhas da árvore após a exclusão dos nós indicados abaixo:

   - 5: _____
   - 4: _____
   - 3: _____

9. Desenhe a árvore binária de pesquisa (não balanceada) gerada pela inserção da seguinte sequência de números: 50, 30, 80, 90, 60, 95, 55, 85, 65, 87, 70, 82, 61, 89.

10. Seja uma árvore *completa*. São necessárias c comparações para constatar que a informação procurada não está armazenada nesta árvore. Informe a quantidade de nós desta árvore para os valores de c abaixo:

    (a) $c = 5$: _____

    (b) $c = 9$: _____

    (c) $c = 11$: _____