



Instituto de Informática – INF
Curso de Bacharelado em Ciência da Computação

Avaliação

Disciplina:	Estruturas de Dados – I	
Professor:	Wanderley de Souza Alencar	Nota:
Aluno(a):		Matrícula:
2ª Avaliação Formal – 2021/1		Data: 11/06/2021

INSTRUÇÕES PARA RESOLUÇÃO DESTA AVALIAÇÃO

1. Esta avaliação possui 05 (cinco) questões, todas com valor igual a 2,0 (dois) pontos;
2. As resoluções das questões devem ser submetidas à área da disciplina no *Sharif Judge System*, cujo endereço é: <<https://sharif.inf.ufg.br/wanderley>>. Neste sistema, cada questão corresponde a 200 (duzentos) pontos e, portanto, atingindo-se 1000 (mil) pontos você obteve pontuação máxima: 10,0 (dez) na avaliação;
3. As submissões das resoluções devem ser realizadas até às **14h do dia 15/06/2021 (terça-feira)**. Não é permitido o envio de resoluções por *e-mail*;
4. Se identificar equívocos numa determinada resolução, é possível reenviar nova versão até a data/hora limite da aplicação da avaliação.

Questão 01 (2,0 pontos). À luz das definições, conceitos e estudos realizados sobre as estruturas de dados denominadas de *Pilhas*, *Filas* e *Filas de Prioridade (heaps)*, bem como sobre a *Análise da Complexidade Assintótica de Tempo* de algoritmos, avalie cada uma das assertivas associando (**V**) para *verdadeira* ou (**F**) para *falsa*.

- (a) () Seja P uma pilha que, inicialmente, está vazia.
Considerando que as operações $\text{push}(P, x)$ e $\text{pop}(P)$ implementam, respectivamente, a inserção do elemento x e a remoção de um elemento nesta pilha, pode-se afirmar que a pilha gerará *underflow* se o número de operações pop executadas por determinado programa (em execução) for maior que o número de operações push ;
- (b) () Se uma certa fila F , não circular, é representada por um vetor de tamanho fixo N , com $N \in \mathbb{N}^*$, sobre o qual estão definidas duas operações: $\text{enfileirar}(F, x)$ e $\text{desenfileirar}(F)$ que, respectivamente, insere o elemento x na fila e remove um elemento da fila, pode-se afirmar que independentemente da maneira estas duas operações sejam implementadas, a complexidade computacional delas será igual a $\mathcal{O}(N)$.
- (c) () Uma *fila de prioridade* é uma estrutura de dados capaz de armazenar um aglomerado de dados – elementos ou células – de tal maneira que as operações fundamentais (inserção, remoção e consulta) sejam realizadas de maneira *eficiente* considerando-se a prioridade associada a cada um destes elementos. Neste contexto, o termo *eficiente* se refere ao número de passos, ou ações básicas, necessários para a completa execução da operação.

- (d) () É possível utilizar um vetor ordenado ou uma lista ordenada para implementar uma fila de prioridade. A primeira abordagem permite a realização da operação de remoção de maneira mais eficiente que a segunda abordagem, pois não há necessidade da realização de operações de movimentação de dados e, sim, apenas ajustes nos *ponteiros* (ou apontadores) para os dados.
- (e) () Dois algoritmos, \mathcal{A}_1 e \mathcal{A}_2 são capazes de corretamente resolver uma instância do problema \mathcal{A} , de tamanho $n \in \mathbb{N}^*$, com complexidades de tempo assintóticas expressas, respectivamente, por $n \cdot \log(n)$ e n^2 . Assim pode-se afirmar que o algoritmo \mathcal{A}_2 é mais eficiente que o algoritmo \mathcal{A}_1 no que diz respeito ao tempo de computação necessário para resolver o mencionado problema.

Observação: Você deverá elaborar, em C ou C++, um programa que recebe como entrada o número 1, que corresponde ao número da questão nesta avaliação e imprima como "*saída*", numa única linha, a sequência de letras V/F por você associada a cada um dos itens (a), (b), (c), (d), (e), nesta ordem. As letras devem ser maiúsculas, SEM QUALQUER ESPAÇO ENTRE ELAS, como no exemplo:

entrada: 1

saída : VFFVV

Questão 02 (2,0 pontos). Considere uma lista linear duplamente encadeada de números inteiros positivos \mathbb{V} , cujo tamanho máximo é de 10000 elementos.

Escreva um programa, em \mathbb{C} ou $\mathbb{C}++$, que seja capaz de realizar as seguintes operações:

C – ordenação dos elementos em ordem estritamente crescente a partir da primeira célula da lista;

D – ordenação dos elementos em ordem estritamente decrescente a partir da primeira célula da lista;

M – apresentar o maior elemento existente na lista;

m – apresentar o menor elemento existente na lista.

Você, então, já partindo para a resolução do problema, identificou que:

Entrada:

A primeira linha conterá os números que formam a lista a partir de sua primeira célula (valor mais à esquerda) até o último número (valor mais à direita). O último número será sempre igual a -1 (menos um) para indicar o encerramento dos elementos, ou seja, ele é um *flag* para indicar o final daquela lista.

A segunda linha conterá uma das operações anteriormente indicadas por uma das letras: C, D, M ou m.

Saída:

A lista ordenada de acordo com a operação selecionada (C ou D) ou, alternativamente, um único número (maior ou menor, de acordo com a operação selecionada - M ou m).

Exemplo 01:

ENTRADA	SAÍDA
1 3 2 5 6 8 7 16 20 9 13 29 -1	1 2 3 5 6 7 8 9 13 16 20 29
C	

Exemplo 02:

ENTRADA	SAÍDA
1 3 2 5 6 8 7 16 20 9 13 29 -1	29 20 16 13 9 8 7 6 5 3 2 1
D	

Exemplo 03:

ENTRADA	SAÍDA
1 3 2 5 6 8 7 16 20 9 13 29 -1	29
M	

Exemplo 04:

ENTRADA	SAÍDA
1 3 2 5 6 8 7 16 20 9 13 29 -1	1
m	

Observação: O programa deverá considerar que os valores fornecidos para a lista são números inteiros positivos, distintos, e no intervalo de 0 a 100000, inclusive extremos.

Questão 03 (2,0 pontos). Um dos grandes problemas brasileiros é o atendimento do SUS – Sistema Único de Saúde. Nele os pacientes que chegam para atendimento numa unidade, digamos X , são inseridos numa fila e passam, imediatamente, por uma triagem que os coloca numa *fila para atendimento*.

Na triagem, o atendente (enfermeiro ou enfermeira) anota o horário de entrada do paciente, quantos minutos ele tem até que sua condição de saúde se torne crítica. Sabe-se que, quando estão na fila para atendimento, os pacientes são atendidos em intervalos de 30 em 30 minutos. Considere, para simplificação, que são sempre nas horas “cheias” ou “meias horas”: 07h, 07h30min, 08h, e assim sucessivamente.

O início da triagem e do atendimento ocorre às 07h e, portanto, neste horário as filas de triagem e atendimento estão vazias. O primeiro paciente é atendido no instante em que chega na triagem. Considere também que o médico atende até o último paciente da fila (lembre-se: isto é para simplificar a solução, não refletindo a realidade!).

A principal preocupação aqui é se um paciente atingiu uma *condição crítica* antes de ter seu atendimento iniciado e, por isso, você convidado para desenvolver uma aplicação – em C ou C++ – que seja capaz de verificar na fila de atendimento quantos pacientes atingem a *condição crítica*.

Para facilitar o desenvolvimento, o líder da equipe lhe deu as seguintes orientações:

- (1^a) considere que a aplicação receberá os dados em *linhas de entrada*, cada linha contém certos dados;
- (2^a) a PRIMEIRA LINHA contém apenas um número inteiro n ($0 < n < 25$) que indica o número de pacientes que chegaram na *fila de triagem*;
- (3^a) as próximas n linhas apresentam os valores para a *hora* (h), *minuto* (m) e *tempo crítico* (tc) em que cada paciente chegou na triagem, sempre sabendo que o paciente da linha i chega antes, ou junto, com o paciente da linha $(i + 1)$ – nunca depois deste. Sabe-se também que $7 < h < 19$, $0 \leq m < 60$ e que $0 \leq tc \leq 720$. O tc é o número de minutos até que o paciente atinja a *condição crítica* em seu estado de saúde. Os números estão, dois a dois, separados por um ‘*espaço em branco*’.

O programa deverá imprimir – numa única linha de saída – o número de pacientes que atingiram a *condição crítica* ainda na fila de atendimento. Veja um exemplo:

1^a linha da entrada: 4
2^a linha da entrada: 7 0 20
3^a linha da entrada: 7 0 30
4^a linha da entrada: 7 30 20
5^a linha da entrada: 8 15 30

linha de saída : 1

Observação: A linha de saída deve ser terminada com um símbolo de *nova linha*.

Questão 04 (2,0 pontos). Escreva um programa, em \mathbb{C} ou $\mathbb{C}++$, que seja capaz de realizar a inserção de valores inteiros e positivos numa lista linear duplamente encadeada com nó descritor e, em seguida, apresentar o valor da *soma dos valores não repetidos nas células* da lista.

A entrada de dados para o programa será realizada no seguinte formato:

1ª linha da entrada: n – quantidade de elementos que serão inseridos na lista;

2ª linha da entrada: n elementos – valores da lista separados por um único “*espaço em branco*” entre eles.

A saída deve expressa numa única linha com o valor da soma dos *valores não repetidos* nas células da lista.

Veja um exemplo:

1ª linha da entrada: 10

2ª linha da entrada: 1 2 3 4 5 6 7 8 9 10

linha de saída : 55

Veja outro exemplo:

1ª linha da entrada: 6

2ª linha da entrada: 1 2 1 2 7 7

linha de saída : 10

Observações: (1ª) Considere que $n \leq 1000$, e que os valores estão no intervalo de 0 a 10000, inclusive extremos; (2ª) A resolução desta questão sem a utilização da estrutura de *lista linear duplamente encadeada com nó descritor* a invalida e, por consequência, será atribuída a nota 0,0 (zero) mesmo que os casos de teste sejam corretamente avaliados pelo *Sharif Judge System*.

Questão 05 (2,0 pontos) Uma operação comum em diversas áreas da computação científica é a multiplicação de números inteiros positivos com grande número de dígitos. Por exemplo, multiplicar dois números de 30 dígitos cada, o que pode gerar um número de até 60 dígitos.

Você está participando de uma equipe de desenvolvimento de uma aplicação científica que deve implementar a operação de multiplicação mencionada.

A aplicação deve ser desenvolvida utilizando a linguagem C ou C++, conforme a seguir especificado.

Entrada

A primeira linha da entrada conterá o número de casos de teste, t , a serem aplicados. Sabe-se que $1 \leq t \leq 50$.

A seguir são apresentadas t linhas, cada uma contendo os dois números inteiros a serem adicionados, digamos m e n , sabendo-se que eles terão no máximo 30 dígitos cada, mas que também poderão ser iguais a 0 (zero). A dupla de números está separada por um único espaço em branco.

Saída

A saída consiste de t linhas, cada uma com o resultado da operação de multiplicação dos pares de números correspondentes, na ordem em que foram fornecidos.

Exemplo 01:

ENTRADA	SAÍDA
1	1167220570724098896488008
9423891297239 123857601272	

Exemplo 02:

ENTRADA	SAÍDA
2	1167220570724098896488008
9423891297239 123857601272	1426349964088696127858578510648863253425
737238112845712940348123 1934720871365475	

Exemplo 03:

ENTRADA	SAÍDA
3	0
0 104759	0
105331 0	0
0 0	

"Why, sometimes I've believed as many as six impossible things before breakfast."
(Lewis Carroll, pseudonym of Charles Lutwidge Dodgson [1832 – 1898])



Figura 1: Autor de *Alice's Adventures in the Wonderland*.