



Instituto Federal de Educação, Ciência e Tecnologia de Brasília – Câmpus Taguatinga
Ciência da Computação – Estrutura de Dados e Algoritmos
Prova II – 1º/2018 – Listas, Filas, Pilhas, Deques e Filas de Prioridade
Prof. Daniel Saad Nogueira Nunes

Aluno: _____

Matrícula: _____

Data: 29 de maio de 2018

Duração: 150 minutos

Tabela de notas (uso exclusivo do professor)

Question:	1	2	3	4	5	6	7	8	Total
Pontos:	1½	1½	1½	1½	1½	1½	1½	1½	12
Score:									

Observações

- Esta prova tem o total de 3 páginas (incluindo a capa) e 8 questões.
- O número total de pontos é 12.
- Certifique-se de assinar todas as folhas de resposta bem como a capa da prova.
- Leia atentamente todas as questões da prova. A interpretação do problema é crucial para o desenvolvimento correto da resposta.
- Resoluções sem justificativa não serão consideradas.
- É vedado o uso de equipamentos eletrônicos, como celulares, notebooks entre outros.
- A prova será **anulada** e medidas disciplinares serão tomadas para os alunos que “colarem” durante a avaliação.

★ Certifique-se de assinar todas as folhas de resposta.

Questão 1 (1½ pontos)

Implemente a função `list_access`, que recebe uma lista e um inteiro k e retorna o dado na posição k de uma lista. Se o inteiro k estiver fora dos limites, você deverá imprimir um erro.

A sua função deverá possuir a seguinte assinatura:

```
void* list_access(list_t* list, int k);
```

Questão 2 (1½ pontos)

Implemente a função `list_reverse`, que recebe uma lista e retorna uma nova lista correspondendo à original invertida. Sua função deverá possuir a seguinte assinatura:

```
list_t* list_reverse(list_t* list);
```

Assuma que você possui todas as funções da biblioteca implementadas.

Questão 3 (1½ pontos)

Implemente a função `drop`, que recebe uma lista e retorna uma nova lista sem os k primeiros elementos. Ela deverá possuir a seguinte assinatura:

```
list_t* list_drop(list_t* list, int k)
```

Tome n o tamanho da lista. Se $k \geq n$, `drop` deve retornar uma lista vazia.

Assuma que você possui todas as funções da biblioteca implementadas.

Questão 4 (1½ pontos)

Implemente a função `filter`, que recebe uma lista e um predicado, e retorna uma nova lista com os elementos que possuem a propriedade dada pelo predicado. Um predicado nada mais é que uma função que retorna **verdadeiro** se o elemento tem uma determinada propriedade e falso caso contrário.

Sua função deverá possuir a seguinte assinatura:

```
list_t* filter(list_t* list, int (*fn)(void* data));
```

Assuma que você possui todas as funções da biblioteca implementadas.

Questão 5 (1½ pontos)

Implemente a função de inserção em uma pilha. Ela deverá possuir a seguinte assinatura:

```
void stack_push(stack_t* stack, void* data);
```

Utilize os tipos definidos pela biblioteca da disciplina.

Questão 6 (1½ pontos)

Implemente a função de remoção em uma fila. Ela deverá possuir a seguinte assinatura:

★ Certifique-se de assinar todas as folhas de resposta.

```
void queue_pop(queue_t* queue);
```

Utilize os tipos definidos pela biblioteca da disciplina.

Questão 7 (1½ pontos)

Faça uma função que receba uma expressão qualquer em uma string, e indique se a quantidade e ordem dos parênteses estão corretas, ou não. Se estiverem corretas, sua função retornará **verdadeiro**, e quando incorretas, falso. A sua função deverá possuir a seguinte assinatura:

```
int avalia_parenteses(char* str);
```

Exemplos de entrada e saída:

Entrada	Saída
a+(b*c)-2-a	1
(a+b*(2-c)-2+a)*2	1
(a*b-(2+c)	0
2*(3-a))	0
)3+b*(2-c)(0

Obrigatoriamente, você deverá utilizar pilhas para resolução desse problema. Assuma que todas as funções da biblioteca estejam implementadas.

Questão 8 (1½ pontos)

Suponha que você tenha um baralho com várias cartas com cada uma podendo assumir um valor de 1 a 13. Você deverá pegar a carta do topo, examiná-la e descartá-la. Se a carta que foi descartada era par, você não deve fazer nada, mas se ela era ímpar, você deverá pegar a carta que está no topo após o descarte e inseri-la no final do monte. Repita esse processo até sobrar uma única carta.

Implemente uma função que recebe uma lista de inteiros contento o monte, de modo que a cabeça da lista seja a carta que se encontra no topo, e imprima na tela a carta restante após o final do procedimento. Sua função deverá possuir a seguinte assinatura:

```
void simula_baralho(list_t* l);
```

Assuma que todas as funções da biblioteca estão implementadas.