

Prova Final de Linguagens de Programação
- DCC024B -
Ciência da Computação

Nome: _____

“Eu dou minha palavra de honra que não trapacearei neste exame.”

Número de matrícula: _____

As regras do jogo:

- A prova é sem consulta.
- Quando terminar, não entregue nada além do caderno de provas para o instrutor.
- Quando escrever código, a sintaxe correta é importante.
- Cada estudante tem direito a fazer uma pergunta ao instrutor durante a prova. Traga o caderno de provas quando vier à mesa do instrutor.
- A prova termina uma hora e quarenta minutos após seu início. O instrutor avisará quando faltarem somente 15 minutos para o final do exame.
- Seja honesto e lembre-se: **você deu sua palavra de honra.**

Alguns conselhos:

- Escreva sempre algo nas questões, a fim de ganhar algum crédito parcial.
- Se não entender a questão, e já tiver gasto sua pergunta, escreva a sua interpretação da questão junto à resposta.
- Serão avaliadas somente as seis melhores respostas. Então sinta-se livre para abandonar alguma questão devido ao tempo.
- A prova não é difícil, ela é divertida, então aproveite!

Tabela 1: Pontos acumulados (para uso do instrutor)

Questão 1	Questão 2	Questão 3	Questão 4	Questão 5	Questão 6	Questão 7

1. Arranjos bi-dimensionais são armazenados *por linha* em Java e em C, conforme ilustrado na figura abaixo:

Uma matriz como esta: será alocada em memória assim:
00 01 02	... 00 01 02 10 11 12 20 21 22 ...
10 11 12	
20 21 22	

É claro que arranjos não precisam ser, necessariamente armazenados desta forma. Poderíamos, por exemplo, armazenar matrizes *por coluna*. Em Fortran matrizes são armazenadas desta forma.

- (a) (5 pontos) Se por alguma ventura, a partir de hoje o comitê da linguagem C decidisse que arranjos deveriam ser armazenados por coluna, haveria algum programa que poderia deixar de funcionar? Por “deixar de funcionar” eu quero dizer, produzir resultados diferentes do esperado. Em caso negativo, explique o porquê, e em caso positivo forneça um exemplo.

- (b) (5 pontos) E no caso de Java, se a forma de alocação mudasse, haveria algum programa que deixaria de estar correto?

2. Alex, Bruno, Carlos e Daniel são quatro colegas de aula. Eles precisam reunir-se para terminar um trabalho, porém todos são bastante ocupados, e arranjar um horário pode ser difícil. Estes alunos têm os seguintes dias livres:

Estudante	Dia livre
Alex	{ segunda, terça, quinta }
Bruno	{ terça, quarta, sexta }
Carlos	{ segunda, terça, sexta }
Daniel	{ segunda, quarta, sexta }

- (a) (4 pontos) Escreva um programa, em Prolog, que determine se estes estudantes têm algum dia livre em comum.
- (b) (3 pontos) Escreva um programa, em Prolog, que determine se algum grupo de três estudantes possui algum dia livre em comum. Cuidado para não dizer que um estudante possui um dia livre com ele mesmo, isto é, procure distinguir um estudante de outro. Sinta-se livre para reutilizar predicados do exercício anterior. Não é preciso re-escrevê-los.
- (c) (3 pontos) Escreva uma busca para determinar quais dias da semana possuem três ou mais estudantes livres. Dica: lembre-se do predicado `findall(E, prep(E), L)`. Este predicado é verdade se `L` for uma lista contendo todos os elementos `E` para os quais `prep(E)` é verdade. Não é preciso re-escrever predicados das questões anteriores, caso queira reutilizá-los.

3. Existem muitos tipos diferentes de coletores de lixo. Alguns são capazes de modificar o conteúdo de ponteiros ou referências, por exemplo.

(a) (5 pontos) Por que um coletor de lixo modificaria o endereço apontado por uma referência? Note que nem todo coletor faz isto, mas o coletor de Java, por exemplo, o faz.

(b) (5 pontos) Por que é mais difícil prover um coletor de lixo assim para C do que para a linguagem Java? Ilustre seu argumento com algum programa escrito em C.

4. Muitas linguagens possuem uma construção sintática chamada *list comprehension*, que permite criar listas usando uma sintaxe muito parecida com aquela que usamos, em matemática, para criar conjuntos. Por exemplo, o conjunto dos quadrados dos números naturais pares menores que 10 poderia ser especificado assim:

$$sqp = \{x^2 \mid x \in \{1, 2, 3, 4, 5, 6, 7, 8, 9\} \wedge x \bmod 2 = 0\}$$

Construtores de listas como este, é claro, são um açúcar sintático em linguagens funcionais. Podemos especificá-los usando, por exemplo, uma combinação de filtros e mapas. Esta possibilidade tanto é verdade que a verificaremos neste exercício.

- (a) (5 pontos) Escreva, em SML, uma função `filter`, cujo tipo seja, `('a -> bool) -> 'a list -> 'a list`. Esta função recebe um predicado `p`, e devolve a lista formada por todos os elementos `e` para os quais `(p e)` seja verdade. Por exemplo:

```
- filter (fn x => x mod 2 = 0) [1,2,3,4,5,6];  
val it = [2,4,6] : int list
```

- (b) (5 pontos) Escreva, em SML, uma função `lc`, cujo tipo seja `('a -> 'b) -> ('b -> bool) -> 'a list -> 'b list`. Esta função deve receber uma função de mapeamento `m`, um predicado `p` e uma lista `l`. A função deve retornar uma nova lista, formada pelos elementos cuja imagem (fornecida por `m`) seja verdade de acordo com `p`. Por exemplo, a nossa função que retorna o quadrado dos números pares entre 0 e 10, usada previamente, ficaria assim:

```
- lc (fn x => x * x) (fn x => x mod 2 = 0) [1,2,3,4,5,6,7,8,9];  
val it = [4,16,36,64] : int list
```

5. (10 pontos) Considere as duas classes abaixo, implementadas na linguagem Java:

```
class Mammal extends Animal {
    public void eat() {
        System.out.println("Mammal is eating");
    }
}

public class Animal {
    public void eat() {
        System.out.println("Animal is eating");
    }

    public static void main(String args[]) {
        Animal a = new Animal();
        Animal m = new Mammal();

        // Chamada 1:
        a.eat();

        // Chamada 2:
        m.eat();
    }
}
```

Existe alguma diferença, ainda que mínima, em termos de eficiência, entre a primeira chamada de método (**Chamada 1**) e a segunda (**Chamada 2**)? Em caso afirmativo, diga qual chamada é mais eficiente, e explique o por quê. Em caso negativo, justifique a sua resposta.

6. Consider a classe abaixo:

```
class MyInt {
    int i;
    MyInt(int k) {i = k;}
    void swap1(MyInt j) {
        MyInt tmp = j;
        j = new MyInt(i);
        i = tmp.i;
    }
    void swap2(MyInt j) {
        MyInt tmp = j;
        j.i = i;
        i = tmp.i;
    }
    void swap3(int j) {
        int tmp = j;
        j = i;
        i = tmp;
    }
}
```

Cada uma das próximas questões é completamente independente uma das outras. Estas questões devem ser respondidas com base nas definições abaixo:

```
MyInt m1 = new MyInt(3);
MyInt m2 = new MyInt(4);
```

- (a) (2 pontos) Qual é o valor de `m1.i` e `m2.i` depois da chamada `m1.swap1(m2)`?
- (b) (2 pontos) Qual é o valor de `m1.i` e `m2.i` depois da chamada `m1.swap2(m2)`?
- (c) (2 pontos) Qual é o valor de `m1.i` e `m2.i` depois da chamada `m1.swap3(m2.i)`?
- (d) (2 pontos) Qual é a política ¹ que java adota para passar tipos primitivos (`int`, `float`, `char`, etc) como parâmetros de métodos?
- (e) (2 pontos) Qual é a política que java adota para passar objetos como parâmetros de métodos?

¹As políticas existentes são passagem por: valor, referência, nome, expansão de matro, retorno, valor-retorno e avaliação preguiçosa

7. Um dos mecanismos de passagem de parâmetros chama-se passagem por nome. Na passagem por nome os parâmetros não são avaliados imediatamente. A avaliação acontece quando estes parâmetros são utilizados. Um detalhe importante, contudo, é que a avaliação acontece no contexto da chamada da função, e não no contexto da função chamada. Isto faz com que não ocorra o problema da captura de variáveis, como ocorre em expansão de macros. A idéia de passagem por nomes foi lançada em Algol, e viu uso também em Simula. Por outro lado, este mecanismo acabou abandonado, pois era difícil de ser implementado, e complicava o código dos programas. É claro, entretanto, que podemos fazer coisas maravilhosas com a chamada por nome. Considere, por exemplo, a função abaixo, escrito em Simula, e responda as perguntas abaixo:

```
Real Procedure Sigma (k, m, n, u);  
  Name k, u;  
  Integer k, m, n; Real u;  
Begin  
  Real s:=0;  
  k:= m;  
  While k <= n Do Begin s:= s + u; k:= k + 1; End;  
  Sigma:= s;  
End;
```

- (a) (5 pontos) Qual o valor de Z na chamada abaixo?

```
Z:= Sigma (i, 1, 4, i ** 2);
```

- (b) (5 pontos) O que o programa abaixo faz? Não precisa escrever o valor, somente explicar o que ele calcula:

```
a := io.read_integer();  
Z := Sigma (i, 1, 100, 1 / (i + a) ** 2);
```