

Primeira Prova de Linguagens de Programação
- DCC024B -
Sistemas de Informação

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. Isso inclui “Quanto tempo falta para acabar a prova”. Para a pergunta: “Posso ir ao banheiro”, a resposta é sim (deixe o telefone celular sobre a mesa para ir ao banheiro).
- A prova termina uma hora e quarenta minutos após seu início.
- Seja honesto e lembre-se: **você deu sua palavra de honra.**

Alguns conselhos:

1. Escreva sempre algo nas questões, a fim de ganhar algum crédito parcial.
2. Se não entender a questão, e já tiver gasto sua pergunta, escreva a sua interpretação da questão junto à resposta.
3. A prova não é difícil, ela é divertida, então aproveite!

Tabela 1: Pontos acumulados (para uso do instrutor)

Questão 1	Questão 2	Questão 3	Extra

Questão extra (0.5): O disco “*Acabou Chorare*” foi considerado, pela revista *Rolling Stones*, como o álbum mais importante da Música Brasileira até 2007. A qual cantor ou banda pertence o disco?

1. Linguagens de programação frequentemente usam “instruções invisíveis”. Essas são instruções usadas para prevenir erros em tempo de execução. Por exemplo, a operação de divisão, em SML gera a exceção `Div [divide by zero]`. Cada operação de divisão é guardada por um condicional que verifica se o dividendo é zero. Caso o seja, a exceção aritmética é lançada. Quanto mais dinâmica a linguagem, mais instruções invisíveis ela possui. Python usa muitas instruções desse tipo. C, por outro lado, não as usa.

(a) (2 Pontos) Dê exemplo de alguma outra operação, além da divisão vista acima, que é guardada por condicionais invisíveis em SML/NJ.

(b) (2 Pontos) Dê exemplo de alguma operação em Java que seja guardada por condicionais invisíveis.

(c) (3 Pontos) A linguagem C não usa instruções invisíveis. Há vantagens nessa abordagem. Cite uma vantagem **usando somente uma palavra**:

(d) (1 Pontos) Há, contudo, várias desvantagens na abordagem de C. Cite uma dessas desvantagens.

(e) (2 Pontos) Use um exemplo que ilustre a desvantagem mencionada no item anterior.

2. Considere a função `max` abaixo, implementada em SML:

```
fun max [e] = e
  | max (h::t) = if max t > h then max t else h
```

(a) (2 Pontos) Qual o tipo da função `max`?

(b) (2 Pontos) A função `max` possui um pior caso exponencial. Como deve ser a entrada (assumindo uma lista com n elementos) para forçar o caso $O(2^n)$?

(c) (2 Pontos) Reescreva a função `max`, para que ela seja linear no número n de elementos da lista de entrada.

(d) (2 Pontos) A função `max`, ao ser compilada, provoca um aviso do tipo `match nonexhaustive`. Por que?

(e) (2 Pontos) Escreva uma função `maxL` de uma linha, que receba uma lista de listas, e retorne o máximo de cada lista. Exemplo:

```
maxL [[2, 3], [3, 4, 5]];
val it = [3,5] : int list
```

3. Esta questão refere-se ao programa abaixo, implementado em uma linguagem de programação hipotética, cuja sintaxe é baseada em Python:

```
def f(x) = x + 2
def h(g) = g(false)
h(f)
```

Explique como seria o sistema de tipagem da linguagem hipotética considerando cada um dos casos de tentativa de execução do programa acima. Especificamente, identifique se o sistema de tipagem é estático ou dinâmico e se há presença de algum dentre os seguintes tipos de polimorfismo: coerção, sobrecarga, paramétrico e/ou subtipagem. Justifique suas respostas.

- (a) (2 Pontos) O programa não compila com o erro:

```
$> Linha 2. Tipo esperado: 'int'. Tipo encontrado: 'bool'
```

- (b) (3 Pontos) O programa compila, mas durante a execução ele termina de modo anormal, com a mensagem:

```
$> Atribuição inválida. Tipo esperado: 'int * int'. Tipo encontrado: 'bool * int'
```

- (c) (2 Pontos) O programa original compila, e retorna o valor 2. Porém, o programa abaixo não compila:

```
def f(x) = x + 2
def h(g) = g(f)
h(f)
```

- (d) (3 Pontos) O programa original compila e retorna o valor 2. O programa abaixo também compila:

```
def f(x) = x + 2
def h(g) = g(f)
h(f)
```

Contudo, o programa acima termina de forma anômala, com a mensagem:

```
$> Operador +: int*int -> int não definido para type<fun>.
```