

Segunda 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. Traga o caderno de provas quando vier à mesa do instrutor.
- A prova termina uma hora e quarenta minutos após seu início.
- 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.
- 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 Pontos): Cite uma bandeira que possui a cor rosa. Bandeiras válidas representam países com acento na assembleia das Nações Unidas, ou estados brasileiros ou capitais de estados brasileiros:

1. Nesta questão você irá explicar o que faz dois programas Prolog diferentes. Cada um deles utiliza a função **append**, cuja implementação é dada abaixo:

```
append([], L, L).  
append([H|T], L2, [H|R]) :- append(T, L2, R).
```

- (a) (2 Pontos) Explique, usando até **dez** palavras, o que faz o seguinte programa Prolog:

```
mystery1([]).  
mystery1([_]).  
mystery1([H|T]) :- append(Mid, [H], T), mystery1(Mid).
```

- (b) (2 Pontos) Complete a lista L para que o predicado abaixo seja verdade:

```
L = [__, __, __, __], mystery1(L).
```

- (c) (2 Pontos) Complete a lista L para que o predicado abaixo seja falso:

```
L = [__, __, __, __], mystery1(L).
```

- (d) Considere o predicado abaixo:

```
mystery2([], []).  
mystery2([H|T], FlatList) :-  
    mystery2(H, FlatH),  
    mystery2(T, FlatT),  
    append(FlatH, FlatT, FlatList).  
mystery2([H|T], [H|FlatT]) :- \+ is_list(H), mystery2(T, FlatT).
```

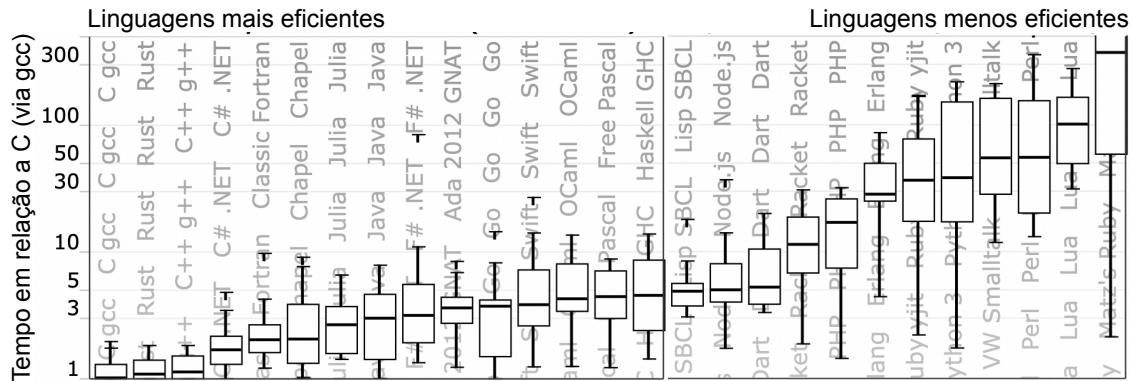
No que se segue, você deverá escrever três exemplos diferentes deste predicado que sejam verdade.

- i. (2 Ponto) Primeiro exemplo:

- ii. (1 Ponto) Segundo exemplo:

- iii. (1 Ponto) Terceiro exemplo:

2. O gráfico abaixo foi retirado da página “The Benchmark Game”. Esta página loga o tempo de execução de soluções de 10 problemas famosos, implementados em linguagens diferentes:



- (a) (2 Pontos) Por que o sistema de tipagem de C (e não estou falando da tipagem estática!) permite que esta linguagem seja tão eficiente?
- (b) (2 Pontos) Dê uma segunda razão, diferente daquela mencionada na questão acima, que justifique a grande eficiência da linguagem C.
- (c) (2 Pontos) Qual a característica comum de PHP, Python, Lua e Ruby que as tornam tão lentas?
- (d) (2 Pontos) Cite uma característica comum entre C, C++, Java e C#.
- (e) (2 Pontos) Cite uma diferença entre estes dois grupos de linguagens: {C, C++} e {Java, C#}.

3. Esta questão se refere à alocação de memória em consequência da execução do programa abaixo:

```
>>> def new_set(x): # Linha 1
...     s = set([x]) # Linha 2
...     return s     # Linha 3
>>> a = new_set(1)   # Linha 4
```

Lembre-se: existem três áreas em que valores ficam armazenados: na memória estática, no heap e na pilha.

- (a) (2 Ponto) Em qual área de memória fica alocado o conjunto $\{1\}$, que é criado pela função `new_set`, quando ela é chamada com o valor 1 na linha 4?
- (b) (2 Ponto) Em qual área de memória fica alocado a variável `s`, que é declarada na função `new_set`, na linha 2?
- (c) (2 Pontos) Quando a variável `s` criada na linha 2 é desalocada?
- (d) (2 Pontos) Quando o conjunto criado na linha 2 é desalocado?
- (e) (2 Pontos) Imagine agora que o programa acima fosse escrito em C++:

```
std::set<int>* new_set(int x) { # Linha 1
    std::set<int>* s = new std::set<int>(); # Linha 2
    s->insert(x); # Linha 3
    return s; # Linha 4
} # Linha 5
int main() { std::set<int>* a = new_set(1); } # Linha 6
```

Quando o objeto criado na linha 2 será desalocado?