Segunda Prova de Linguagens de Programação
- DCC024 -
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. São perguntas: "Posso fazer uma pergunta?"ou "Quanto tempo falta?"
- 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!

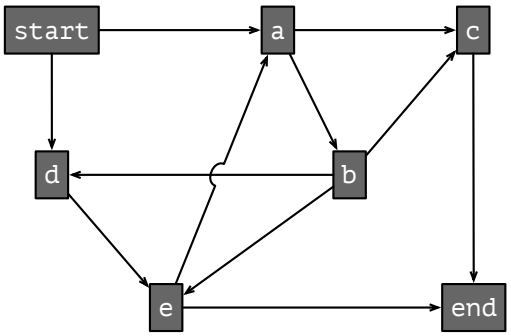
Tabela 1: Pontos acumulados (para uso do instrutor)

Questão 1	Questão 2	Questão 3	Questão 4	Extra

Questão Extra (0.5 Pontos): "*Magister bonus discipulos suos curiosos facit.*". Então: qual o nome atual do país em que ficaria a Cacuânia, se ela de fato houvesse existido?

1. É possível representar grafos direcionados em Prolog usando uma relação `edge(V1, V2)`, que representa a existência de uma aresta do vértice `V1` para o vértice `V2`. A figura abaixo mostra um exemplo:

```
edge(start, a).
edge(start, d).
edge(a, b).
edge(b, c).
edge(a, c).
edge(b, d).
edge(b, e).
edge(d, e).
edge(e, a).
edge(c, end).
edge(e, end).
```



   (a) (3 Pontos) Implemente um predicado `path(L)`, que seja verdade quando `L` for uma lista de vértices que denote um caminho no grafo. Em outras palavras, se $\texttt{L} = v_1, v_2, \ldots, v_n$, então deve existir uma aresta $v_i \rightarrow v_{i+1}$ para todo $i, 1 \leq i \leq n-1$. Por exemplo, considerando o grafo acima:

```
path([a, b, e, end]).
true ;
path([b, a, e, end]).
false.
```

   Note que uma lista vazia é um caminho. Também é um caminho uma lista com um elemento só.

   (b) (2 Pontos) Implemente um predicado `last(L, E)`, que seja verdade quando `E` for o último elemento da lista L. Por exemplo:

```
last([a, b, e, end], X).
X = end.
path([a], X).
X = a.
path([], X).
false.
```

   (c) (2 Pontos) Implemente um predicado `cycle(L)`, que seja verdade quando `L` formar um ciclo, isto é: um caminho tal que exista uma aresta do último para o primeiro elemento. Assuma que todo ciclo precisa ter no mínimo um vértice, e que não existe ciclo de um vértice v para ele mesmo, a menos que o predicado `edge`(v, v) seja verdadeiro. Exemplos:

```
?- cycle([e, a, b]).
true ;
?- cycle([a]).
false.
```

   (d) (2 Pontos) Implemente um predicado `solution(L)`, que seja verdade se: (i) `L` for uma lista que forme um caminho de vértices; (ii) o primeiro vértice de `L` for `start` e (iii) o último vértice de `L` for `end`.

   Exemplos:

```
?- solution(L).
L = [start, a, b, c, end] ;
?- solution([start, d, e, end]).
true ;
```

2. Esta questão refere-se ao programa abaixo, escrito na linguagem de programação C:

```
01 #include <stdio.h>
02 #include <stdlib.h>
03
04 int foo(int** a, i, j) {
05   return a[i][j];
06 }
07
08 int bar(int b[2][3], i, j) {
09   return b[i][j];
10 }
11 int main() {
12   int** a = (int**)malloc(2 * sizeof(int*));
13   a[0] = (int*)malloc(3 * sizeof(int));
14   a[1] = (int*)malloc(3 * sizeof(int));
15   a[0][0] = 0; a[0][1] = 1; a[0][2] = 2;
16   a[1][0] = 3; a[1][1] = 4; a[1][2] = 5;
17   int b[2][3] = {{0, 1, 2}, {3, 4, 5}};
18   int c[3][2] = {{10, 20}, {30, 40}, {50, 60}};
19   printf("A: %d\n", foo(a, 1, 1));
20   printf("B: %d\n", foo(b, 1, 1));
21   printf("C: %d\n", bar(a, 1, 1));
22   printf("D: %d\n", bar(b, 1, 1));
23   printf("E: %d\n", bar(c, 1, 1));
24 }
```

Para responder as questões seguintes, procure pensar em como seria a implementação das funções `foo` e `bar` em linguagem de máquina. Você não precisa conhecer linguagem de máquina: basta imaginar como cada função implementa o acesso à memória usando operações de carregamento de dados.

(a) (2 Pontos) O que imprime a chamada à função `printf` na linha 19?

(b) (2 Pontos) A chamada à função `printf` na linha 20 causa um erro do tipo "falha de segmentação". Por que isso ocorre?

(c) (2 Pontos) Duas execuções diferentes do programa acima podem causar a chamada da função `printf` na linha 21 a imprimir dois valores diferentes. Em outras palavras, o comportamento da linha 21 não é bem definido. O que causa esse comportamento indefinido?

(d) (2 Pontos) O que imprime a chamada à função `printf` na linha 22?

(e) (2 Pontos) O que imprime a chamada à função `printf` na linha 23? Não responda tão rápido! Essa linha possui comportamento determinístico, porém é bem possível que esse comportamento não seja o mais intuitivo.

3. Em cada um dos itens abaixo, você devera reescrever, em Python, um programa mostrado em alguma outra linguagem de programação. Você deve também reconhecer a linguagem em que está escrito o programa original. Veja que uma tradução perfeita pode não ser possível. Quando for o caso, busque escrever o programa mais próximo em que puder pensar.

```
- (fn x => fn y => x + y) 2 3;
val it = 5 : int
```

(1 Ponto) Linguagem de origem:

_____

(2 Pontos) Código Python:

```
void foo(int v[], int N) {
  for (int i = 0; i < N/2; i++) {
    int aux = v[i];
    v[i] = v[N-i-1];
    v[N-i-1] = aux;
  }
}
```

(1 Ponto) Linguagem de origem:

_____

(2 Pontos) Código Python:

```
class P {
  private double x, y;
  public P(double x, double y) {
    this.x = x;
    this.y = y;
  }
  public boolean positive() {
    return x > 0 && y > 0;
  }
}
```

(1 Ponto) Linguagem de origem:

_____

(2 Pontos) Código Python:

(1 Ponto) O último programa não pode ser traduzido para Python de forma exata. O que falta em Python nesse caso?

4. Historicamente, o projeto de linguagens de programação é influenciado por vários fatores. Nesta questão abordaremos alguns desses fatores.

   (a) (5 Pontos) O protocolo HTTP, que permitiu o surgimento da WWW, foi criado no início dos anos 90. A partir deste protocolo surgiu a rede de computadores hoje conhecida como "Internet". Como o surgimento da internet influenciou o projeto e implementação de linguagens de programação?

   (b) (5 Pontos) No final da década de 70, e mais notadamente durante a década de 80, assistimos ao surgimento de programas computacionais muito grandes: `Electric Pencil`, `WordPerfect`, `Multiplan`, `Lotus 1-2-3`, `MS Excel`, `MS PowerPoint`, `MS Word`, etc. Como o crescimento dos sistemas computacionais (em termos de números de arquivos e linhas de código) influenciou o projeto e implementação de linguagens de programação?