Trabalho Prático de Linguagens de Programação (v. 2.0)
Universidade Federal do Amazonas
Departamento de Ciência da Computação
Marco Cristo

Introdução

Neste trabalho, **trios** de alunos devem comparar **duas** linguagens e explorar dois diferentes paradigmas (Python-Paradigma1 x Outra-Paradigma2). Vocês definem a segunda linguagem, os paradigmas e o problema. As únicas restrições são que a solução seja focada em paradigmas distintos e o problema seja de tamanho moderado.

Objetivos

a) Implementar uma aplicação de tamanho pequeno/moderado (de 100 a 200 linhas de código) em Python e em outra linguagem de sua escolha (note que a aplicação vai depender da linguagem escolhida – por exemplo, para uma linguagem a aplicação pode envolver uma interface gráfica e para a outra, uma interface textual).
b) Escrever um relatório de avaliação das linguagens (com parte da avaliação baseada em seu projeto).
c) Entregar códigos e relatório.

Detalhamento

O relatório escrito deverá conter, minimamente, os seguintes itens:

1) Parte I – **Introdução** (no máximo, duas páginas)

   - Breve história das linguagens, incluindo seu projetista e objetivos de projeto;
   - Uso atual: ela ainda é usada e para quê?
   - Paradigmas suportados (e qual foi explorado no trabalho?)
   - Principais características (por exemplo, linguagem de propósito geral ou de *script*, compilada, interpretada? Suporta programação genérica? Etc)
   - Em geral, como ela poderia ser descrita em termos de segurança, portabilidade, eficiência, estensibilidade e expressividade.

2) Parte II – **Descrição da segunda linguagem** (com pequenos exemplos, tentando manter o máximo de seis páginas)

   - Mecanismos de fluxo de controle (seqüência, seleção, iteração, suporta/encoraja recursão?);
   - Tipos de Dados (tenha a certeza de incluir tipos básicos – inteiro, lógico, caractere, etc -- e compostos -- registros, vetores & matrizes, ponteiros, listas, arquivos, mapas, etc. Determine se ela é uma linguagem de tipagem estática ou dinâmica, forte ou fraca)
   - Procedimentos (mecanismos de passagem de parâmetros suportados e possibilidades polimórficas; suporta concorrência?)

3) Parte III - **Avaliação** (1 a 2 páginas)

Baseado na sua experiência com as linguagens e, considerando as restrições impostas pelo paradigma usado, tempo e recursos disponíveis, forneça uma breve avaliação comparativa das linguagens. Procure responder os seguintes itens nesta avaliação:

- Descreva o seu projeto, detalhando seus objetivos e especificidade do projeto para cada linguagem.
- Como as duas linguagens se compararam em termos de segurança, portabilidade, eficiência, estensibilidade e expressividade? Elas corresponderam ao que foi descrito no início?
- Quais aspectos das linguagens são (foram) mais difíceis de aprender e usar?
- Como você compara os ambientes usados, as mensagens de erro dadas e a facilidade de construir a solução?

4) Parte IV – **Código Fonte** – Os códigos do aplicativo-alvo nas duas linguagens, com breves exemplos de execução.

Observações

- Seu relatório deve ser mais que uma compilação das listas de informações sugeridas acima. Dê exemplos e faça comentários, para que eu possa ver um pouco de você no texto e não uma mera colagem de informações obtidas na web. Se você estiver na dúvida sobre se está no caminho certo, pode mostrar o trabalho antes da data de entrega.

- Cite todas as fontes que você usou na pesquisa.

Sugestões de projetos possíveis (para que vocês não digam que eu não sugeri nada ☺)

- Eliza (ver **Anexo I**, para uma definição mais detalhada)
- Gerador de cardápios e listas de compras (ver **Anexo II**, para uma definição mais detalhada)
- Mini-Logo textual (ver **Anexo III**, para uma definição mais detalhada)
- Calculadora básica com interface gráfica ou textual baseada em comandos interpretados
- Jogo da Senha contra computador (http://pt.wikipedia.org/wiki/Mastermind)
- Um programa que trace funções matemáticas em um intervalo (ex: exibe o gráfico de uma função y = x * x, com -10 < x < 10, onde função e intervalos são fornecidos – a avaliação da função é particularmente simples em linguagens funcionais, incluindo Python – ver *eval*)
- Mini-jogo da batalha naval, em que o adversário é o computador (regras em http://pt.wikipedia.org/wiki/Batalha_naval_(jogo))
- Programa de balanço financeiro (fazer depósitos, saques, consultar saldo, obter extrato, fazer transferências entre contas, usando arquivos pra gravas as informações)
- Agenda Telefônica (opções de consulta, modificação e remoção de dados, com dados gravados e lidos pra arquivos)
- Simulador de jogos de algum esporte (entre times e estatísticas usadas na simulação – por exemplo, força da defesa e ataque – e [a] gere placares para todos os jogos, rodada a rodada; [b] emita relatórios sobre a classificação dos times, incluindo várias estatísticas como número de pontos, vitórias, etc) [você pode encontrar uma descrição do algoritmo que gera as rodadas em http://en.wikipedia.org/wiki/Round-robin_tournament]
- Blackjack (http://pt.wikipedia.org/wiki/Blackjack)
- Um RPG simples baseado em texto (http://www.kirsle.net/creativity/javascript/jsrpg.html)
- Pong (http://www.youtube.com/watch?v=LPkUvfL8T1I)
- Sistema de votação eletrônica na Web

Anexo I – Projeto ELIZA

(cerca de umas 400 linhas em C++, grande parte delas para definição das expressões regulares)

Introdução

Joseph Weizenbaum, em 1966, desenvolveu um programa chamado ELIZA. ELIZA deveria ser uma paródia a um terapeuta Rogeriano cuja técnica consiste em repetir sentenças dos pacientes. Quando era executado, o programa pedia uma entrada de um usuário humano, que a digitava em linguagem natural em um terminal. ELIZA então fornecia respostas aparentemente inteligentes. Uma sessão típica era assim:

```
eliza> Oi, como vai você?
human> Estou me sentindo cansado hoje.
eliza> Porque você está se sentindo cansado hoje?
human> Meus projetos estão me matando.
eliza> Seus projetos estão lhe matando?
human> Eles nos fazem trabalhar muito.
eliza> Você pode ser mais específico?
human> O meu professor.
eliza> Fale-me mais sobre o seu professor.
human> Ele fala pra caramba.
eliza> Você acha que isso se deve à infância?
```

Como você pode ver, ELIZA se comporta como se compreendesse o que o usuário diz. Realmente, o programa está simplesmente procurando por certos padrões na entrada do usuário. Por exemplo, na conversa acima, ELIZA observa que a primeira sentença inicia com “Estou me sentindo”. Ao ver tal padrão, ELIZA é programado a construir uma resposta que inicia com `Porque você está se sentindo'. O resto de sua resposta simplesmente repete o restante da sentença do usuário, ”cansado hoje”, que é precisamente como o usuário disse que se sentia.

O programa real era mais complicado que isso pois tinha memória e algum contexto. Ainda assim é possível gerar respostas interessantes apenas casando padrões. Este é o seu trabalho neste projeto.

Expressões Regulares

Uma sentença é uma seqüência de literais, ou palavras. Uma expressão regular é uma forma concisa de descrever uma seqüência de literais ou um conjunto dessas sentenças. As expressões regulares em ELIZA são seqüência que usam três tipos de componentes:

- Um literal específico.
- Um literal qualquer (representado por `_?` neste texto).
- Uma seqüência de literais (representado por `_*` neste texto).

Os dois últimos são chamados *curingas*. Quando dizemos que uma expressão regular casa com uma entrada, nós queremos dizer que toda a palavra na entrada corresponde a uma palavra ou curinga na expressão regular, na ordem correta.

Assim, a expressão regular “_? are _*” casa com as seguintes sentenças:

- ``we are" (_? casa com “we” e _* casa com nada)

- ``they are here" (_? casa com “they” e _* casa com ``here")
- ``we are what we are" (_? casa com “we” e _* casa com ``what we are")

Mas não casa com:

- ``here we are" (``here" não casa)
- ``we think we are going" (embora o _* case com ``going", não há curinga que case com o início da sentença, “we think”, já que o _? casa com uma palavra apenas)

Como Funciona o ELIZA

Sua implementação do Eliza deve incluir uma lista de regras. Cada regra na lista será um par consistindo de uma expressão regular e um modelo para a resposta a ser dada. Usando as regras, o seu programa irá fazer o seguinte:

1. Dada a entrada do usuário, ela será comparada com as expressões regulares de cada uma das regras, até casar com alguma. Para tanto, normalmente são retirados todos os sinais de pontuação da entrada (exceto o ? que identifica perguntas), ela é transformada para uma forma padrão (só letras minusculas, por exemplo) e separada em palavras.
2. Ao casar com uma regra, da regra casada serão extraídas as porções da entrada do usuário que correspondem a cada curinga na regra (exceto o ?).
3. Então, a resposta ao usuário sera construída de acordo com o modelo, usando as porções extraídas.

Digamos que o usuário digite:

```
Eu gosto de C++ porque OO é o máximo
```

Primeiro, é necessário determinar se esta entrada casa com uma regra. Assuma que há uma regra cuja expressão regular é ``Eu gosto de _? porque _*" e o modelo de resposta é “É realmente por causa de _2 que você gosta de _1?”. Você irá retornar esta regra, extrair as porções associadas aos curingas (_? = “C++” e _* = “OO é o máximo”) e incluí-los no modelo de acordo com os índices dados, correspondentes aos curingas. Ou seja, _2 corresponde ao _* e _1 ao _?, neste caso. Assim, a resposta a ser dada é “É realmente por causa de OO é o máximo que você gosta de C++?”.

Note que pode ser necessária alguma mudança de contexto ao substituir índices pelo conteúdo dos curingas. Por exemplo, dadas a regra “Eu gosto _*” -> “Porque você gosta _1?” e a entrada “Eu gosto do meu carro”, a resposta deveria ser “Porque você gosta do seu carro?”. Neste caso, “meu” foi substituído por “seu”. Mudanças de contexto devem envolver pronomes (eu -> você; seu -> meu, meu -> seu, minha -> sua, etc) e alguns verbos simples (sou -> é, tenho -> tem, etc).

No caso de uma entrada casar mais de uma regra, escolha a primeira. Isso permite que você coloque regras mais específicas no início e mais gerais no fim.

Objetivo

Sua tarefa é criar um programa Eliza que consuma entradas do usuário e forneça respostar de acordo com um conjunto interno de regras (várias regras são dadas em anexo).

Algumas sugestões de mudanças de contexto (sempre considere também a ordem inversa)

Meu -> seu
Minha -> sua
Eu -> você
Sou -> é
Fui -> foi
Estou -> está
Tenho -> tem
Tive -> teve
Fomos -> eram
Meus -> seus
Minhas -> suas
Teus -> meus

Algumas sugestões de regras (em ordem de avaliação)

Expressão regular	*Modelo de resposta*
_* oi _*	Como vai você? Por favor, me fale do seu problema.
_* computador _*	Computadores te incomodam?
	O que você acha sobre máquinas?
	Porque você menciona computadores
	O que você acha que máquinas tem a ver com o seu problema?
_* nome _*	Não estou interessado em nomes
_* desculpe _*	Por favor, não se desculpe
	Desculpas não são necessárias
	Como você se sente quando se desculpa?
_* eu lembro _*	Você normalmente lembra _2?
	Lembrar _2 traz alguma outra lembrança à sua mente?
	Que outras coisas você lembra?
	Porque você lembra _2 nesse momento?
	O que na situação atual faz você lembrar _2?
	Qual a conexão entre lembrar _2 e eu?
_* você lembra _*	Você acha que eu me esqueceria _2?
	Porque você acha que eu deveria lembrar _2 agora?
	E que tal _2?
	Você mencionou _2?
_* se _*	Sério?
	De verdade? se _2?
_* eu sonhei _*	Realmente? _2?
	Você já sonhou _2 enquanto acordado?
	Você já havia sonhado _2 antes?

_* eu sonhei com _*	Como você se sente em relação a _2 na verdade?
_* sonho _*	O que este sonho sugere a você?
	Você sonha com freqüência?
	Que pessoas aparecem em seus sonhos?
	Você não acha que sonhos têm algo a ver com o seu problema?
_* minha mãe _*	Quem mais na sua família _2?
	Fale-me mais sobre a sua família
_* meu pai _*	Seu pai?
	Ele influencia você fortemente?
	O que mais vem à sua mente quando você pensa no seu pai?
_* eu quero _*	Qual a importância de ter _2?
	Porque você quer _2?
	Acho que você terá _2 logo
_* estou feliz _*	Eu tenho alguma influência nisso?
	O que te faz feliz?
	Você pode explicar o porque de feliz?
_* estou triste _*	Sinto que você se sinta assim
	Estou certo de que não é prazeroso estar assim
_* são como _*	Que semelhança você vê entre _1 e _2
_* é como _*	De que forma _1 é como _2
	Que semelhança você vê?
	Será que há realmente alguma coisa em comum?
	Como?
_* parece _*	De que forma?
	Que similaridades há?
_* mesmo _*	Que outras conexões você observa?
_* eu era _*	Você era realmente?
	Talvez eu soubesse que você fosse _2
	Porque você está me dizendo que era _2 agora?
_* eu era _*?	E se você estivesse _2 ?
	Você acha que era _2
	E qual o siginificado de ser _2
_* eu sou _*	E como você é _2
	Você quer ser _2 ?
_* eu sou _*?	Você acredita que você seja _2
	Você gostaria de ser _2
	Você gostaria que eu dissesse que você é _2
	E qual o significado de ser _2
_* sou _*	Porque você está dizendo "SOU"?
	Nao entendi
_* você é _* ?	E qual o seu interesse em saber se sou _2 ou não?
	Você iria preferir se eu não fosse _2
	Talvez eu seja _2 em suas fantasias
_* você é _*	O que faz você pensar que eu sou _2 ?
_* por causa _*	Essa é a razão?
	Que outras razões você acha que poderiam haver?
	E isto explica tudo?
_* você era _* ?	Talvez eu fosse _2

	O que você acha?
	E se eu fosse _2
_* eu não posso _*	Talvez você pudesse _2 agora
	E se eu pudesse _2 ?
_* eu sinto _*	Você sempre sente _2 ?
_* eu sentia _*	Que outras coisas você sente?
_* porque você não _*	Você _2?
	Você acredita que eu não _*
	Talvez eu vá _2 depois
_* sim _*	Você parece uma pessoa bem positive
	Tem certeza
	Entendo
_* não _*	Porque não?
	Você está sendo um pouco negative
	Você diz não só pra ser negativo?
_* alguém _*	Você pode ser mais específico?
_* todos _*	Com certeza não todos
	Pode pensar em alguém em particular?
	Quem por exemplo?
	Você está pensando em alguém em particular?
_* sempre _*	Você pode dar um exemplo específico
	Quando?
	Sobre o que você está pensando?
	Realmente – sempre
_* o que _*	Porque você pergunta?
	Essa pergunta é interessante para você?
	O que você quer saber de verdade?
	O que você acha?
	O que vem à sua mente quando pergunta isso?
_* talvez _*	Você não parece muito certo
_* são _*	Você acha que eles poderiam não ser _2
	Possivelmente eles são _2
Bye	Tchau.
_*	Muito interessante.
	Não sei se entendi você direito
	O que isso sugere a você?
	Por favor, continue.
	Continue
	Você quer mesmo falar sobre isso?

Anexo II – Gerador de Cardápios e Listas de Compras
(cerca de umas 110 linhas em AWK)

Introdução

Pessoas que controlam a nutrição costumam ter várias opções balanceadas de cardápio. Para realizar as várias refeições de forma adequada, é necessário ter um conjunto mínino de mantimentos. Assim, para estas pessoas, dada uma série de opções de refeições, seria interessante que um programa pudesse gerar um cardápio variado para um conjunto de dias (por exemplo, uma semana) junto com uma lista de itens necessários para a realização daquele cardápio.

Como tal programa iria funcionar

As opções de refeições podem ser informadas através de uma lista que descreve cada tipo de refeição, um identificador de opção, um item da refeição e a sua quantidade correspondente (quantidade, unidade e multiplicador). Por exemplo:

```
café,1,leite desnatado,250,ml,1
café,1,pão integral,2,fatia,0.4
café,1,queijo minas,1,fatia,0.5
café,1,peito de peru,2,fatia,0.5
café,1,mamão,0.5,unidade,1.5
café,2,iogurte desnatado natural batido com mel,250,ml,1.5
café,2,pão integral,2,fatia,0.4
café,2,queijo minas,1,fatia,0.5
café,2,maça/pera,1,unidade,1
almoco,1,salada colorida,1,unidade,1
almoco,1,filé grande peixe grelhado,1,unidade,1
almoco,1,arroz integral,4,colher de sopa,1
almoco,2,salada colorida,1,unidade,1
almoco,2,filé peito de frango na chapa,1,unidade,1
almoco,2,purê de batata,1,porção,1
```

A lista acima descreve duas refeições (almoço e café) com duas opções cada (opções 1 e 2). Por exemplo, a primeira opção de almoço corresponde a um filé de peixe grelhado com arroz integral e salada colorida. Na primeira linha, o primeiro item da primeira opção de café é um copo de 250 ml de leite desnatado. Neste caso, o multiplicador é 1, a quantidade é 250 e a unidade é ml.

Dada esta lista e um número de dias o programa deveria gerar um cardápio para os dias e uma lista de compras correspondente. Por exemplo, para sete dias, o cardápio poderia ser algo como:

```
Dia 1:
almoco: salada colorida (1 unidade) + filé peito de frango na chapa (1 unidade) + purê
de batata (1 porção)
jantar: leite morno (250 ml) + mel (1 colher de sopa) + torrada (2 unidade) + queijo
minas (2 fatia)
lanche-tarde: iogurte desnatado (1 copo) + torrada (2 unidade) + requeijão light (1
fatia) + peito de peru (1 fatia)
café: iogurte desnatado natural batido com mel (250 ml) + pão integral (2 fatia) +
queijo minas (1 fatia) + maça/pera (1 unidade)
lanche-pos-treino: barra de cereal light (1 unidade) + água (300 ml)

Dia 2:
almoco: salada colorida (1 unidade) + filé peito de frango na chapa (1 unidade) + purê
de batata (1 porção)
jantar: leite morno (250 ml) + mel (1 colher de sopa) + torrada (2 unidade) + queijo
minas (2 fatia)
```

```
lanche-tarde: chá mate light (1 caixinha) + pão integral (2 fatia) + queijo minas (1
fatia) + peito de peru (2 fatia)
café: iogurte desnatado natural batido com mel (250 ml) + pão integral (2 fatia) +
queijo minas (1 fatia) + maça/pera (1 unidade)
lanche-pos-treino: iogurte light ou desnatado (molico. corpus. activia0) (250 ml)

...
```

A lista de compras correspondente seria:

```
QTDE   UNIDADE        ITEM
-----------------------------------------
 2     livre          abóbora
 8     colher de sopa arroz integral
 4     colher de sopa arroz sete cereais
 3     fatia          atum
 5     colher de sopa aveia em farelo
 2     unidade        barra de cereal light
 2     prato fundo    canja de galinha
 3     caixinha       chá mate light
 2     livre          ervilhas
 1     concha média   feijão
 1     unidade        filé grande peixe na chapa
 2     unidade        filé médio de carne vermelha
 4     unidade        filé peito de frango na chapa
 6     colher de sopa granola light
 1     concha média   grão de bico
 750   ml             iogurte desnatado natural batido com mel
 752   ml             iogurte desnatado
 500   ml             iogurte light ou desnatado (molico. corpus. activia0)
 1     unidade        laranja/maça/pera/abacaxi1/morangos5/tangerina
 1500  ml             leite desnatado
 500   ml             leite morno
 1     concha média   lentilha
 5.5   unidade        mamão
 11    unidade        maça/pera
 2     colher de sopa mel
 3     unidade        omelete de 2 claras e 1 gema
 2     livre          palmito
 14    fatia          peito de peru
 4     porção         purê de batata
 24    fatia          pão integral
 11    fatia          queijo minas
 2     fatia          requeijão light
 7     unidade        salada colorida
 2     prato fundo    sopa de legumes
 750   ml             suco de acerola
 2     caixinha       suco de soja
 8     unidade        torrada
 2     medidor        whey protein
 600   ml             água
```

Objetivo

Sua tarefa é criar um programa que, dada uma lista de opções de refeições e um intervalo de dias, gere cardápios aleatórios para os dias com uma lista de compras correspondente.

Observação

Abaixo, é dado um exemplo de uma lista de opções de refeições:

```
café,1,leite desnatado,250,ml,1
café,1,pão integral,2,fatia,0.4
café,1,queijo minas,1,fatia,0.5
café,1,peito de peru,2,fatia,0.5
café,1,mamão,0.5,unidade,1.5
café,2,iogurte desnatado natural batido com mel,250,ml,1.5
café,2,pão integral,2,fatia,0.4
```

café,2,queijo minas,1,fatia,0.5
café,2,maça/pera,1,unidade,1
café,3,leite desnatado,250,ml,1
café,3,maça/pera,1,unidade,1
café,3,mamão,1,unidade,3
café,3,aveia em farelo,1,colher de sopa,1
café,3,iogurte desnatado,250,ml,1.3
café,3,granola light,2,colher de sopa,0.6
café,3,maça/pera,1,unidade,1
lanche-pos-treino,1,leite desnatado,250,ml,1
lanche-pos-treino,1,maça/pera,1,unidade,1
lanche-pos-treino,1,mamão,1,unidade,1
lanche-pos-treino,1,aveia em farelo,1,colher de sopa,1
lanche-pos-treino,1,whey protein,1,medidor,1
lanche-pos-treino,2,iogurte light ou desnatado (molico. corpus. activia0),250,ml,1
lanche-pos-treino,3,barra de cereal light,1,unidade,1
lanche-pos-treino,3,água,300,ml,1
lanche-pos-treino,4,laranja/maça/pera/abacaxi1/morangos5/tangerina,1,unidade,1
almoco,1,salada colorida,1,unidade,1
almoco,1,filé grande peixe grelhado,1,unidade,1
almoco,1,arroz integral,4,colher de sopa,1
almoco,2,salada colorida,1,unidade,1
almoco,2,filé peito de frango na chapa,1,unidade,1
almoco,2,purê de batata,1,porção,1
almoco,3,salada colorida,1,unidade,1
almoco,3,filé médio de carne vermelha,1,unidade,1
almoco,3,arroz integral,4,colher de sopa,1
almoco,3,feijão,1,concha média,1
almoco,4,salada colorida,1,unidade,1
almoco,4,filé médio de carne vermelha,1,unidade,1
almoco,4,arroz integral,4,colher de sopa,1
almoco,4,lentilha,1,concha média,1
almoco,5,salada colorida,1,unidade,1
almoco,5,filé grande peixe na chapa,1,unidade,1
almoco,5,arroz sete cereais,4,colher de sopa,1
almoco,5,grão de bico,1,concha média,1
lanche-tarde,1,suco de soja,1,caixinha,1
lanche-tarde,1,pão integral,2,fatia,1
lanche-tarde,1,peito de peru,2,fatia,1
lanche-tarde,2,suco de soja,1,caixinha,1
lanche-tarde,2,pão integral,2,fatia,1
lanche-tarde,2,atum,2,fatia,1
lanche-tarde,3,chá mate light,1,caixinha,1
lanche-tarde,3,pão integral,2,fatia,1
lanche-tarde,3,queijo minas,1,fatia,1
lanche-tarde,3,peito de peru,2,fatia,1
lanche-tarde,4,iogurte desnatado,1,copo,1
lanche-tarde,4,torrada,2,unidade,1
lanche-tarde,4,requeijão light,1,fatia,1
lanche-tarde,4,peito de peru,1,fatia,1
jantar,1,salada de alface americana,1,unidade,1
jantar,1,tomate cereja,1,unidade,1
jantar,1,pepino,1,unidade,1
jantar,1,filé de peito de frango,1,unidade,1
jantar,1,chá verde,1,copo,1
jantar,2,sopa de legumes,1,prato fundo,1
jantar,2,canja de galinha,1,prato fundo,1
jantar,2,palmito,1,livre,1
jantar,2,abóbora,1,livre,1
jantar,2,ervilhas,1,livre,1
jantar,3,leite morno,250,ml,1
jantar,3,mel,1,colher de sopa,1
jantar,3,torrada,2,unidade,1
jantar,3,queijo minas,2,fatia,1
jantar,4,suco de acerola,250,ml,1
jantar,4,omelete de 2 claras e 1 gema,1,unidade,1
jantar,4,atum,1,fatia,1
jantar,4,pão integral,2,fatia,1
jantar,5,suco de abacaxi,250,ml,1
jantar,5,pão integral,2,fatia,1
jantar,5,queijo minas,1,fatia,1
jantar,5,peito de peru,2,fatia,1
jantar,6,mingau de aveia feito com leite desnatado,250,ml,1
jantar,6,maça/pera,1,unidade,1
jantar,6,canela em pó,1,livre,1

Anexo III – Mini-LOGO
(cerca de 150 linhas em C++)

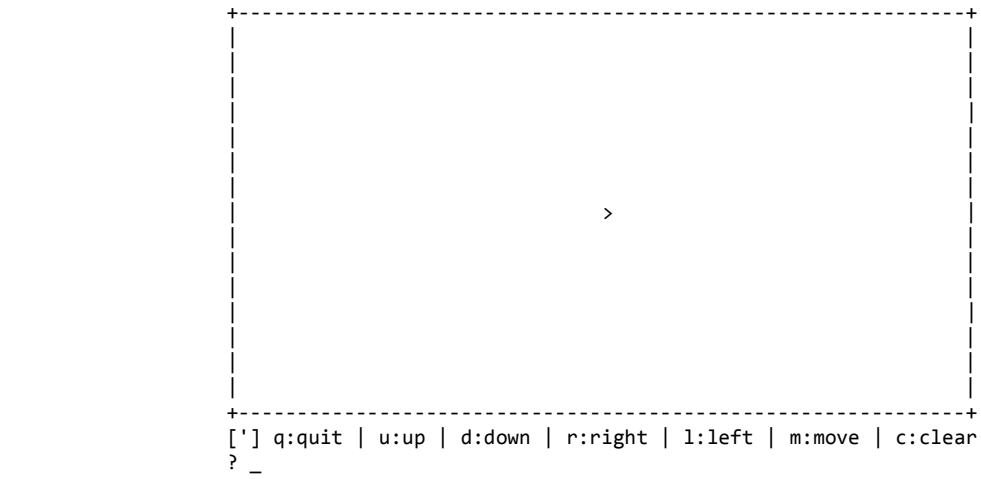
Introdução

Logo é uma linguagem de programação interpretada, criada inicialmente para o ensino de programação para crianças por meio de experiências concretas. Ela implementa, em certos aspectos, a filosofia construtivista, segundo a interpretação de Seymour Papert, co-criador da linguagem junto com Wally Feurzeig.

O ambiente Logo tradicional envolve uma tartaruga gráfica para responder aos comandos do usuário. Uma vez que a linguagem é interpretada e interativa, o resultado é mostrado imediatamente após digitar-se o comando – incentivando o aprendizado. Assim, o aluno aprende com seus erros. Se algo está errado em seu raciocínio, isto é claramente percebido e demonstrado na tela, fazendo com que o aluno pense sobre o que poderia estar errado e tente, a partir dos erros vistos, encontrar soluções corretas para os problemas. A maioria dos comandos, refere-se a desenhar e pintar.

Um ambiente Logo básico

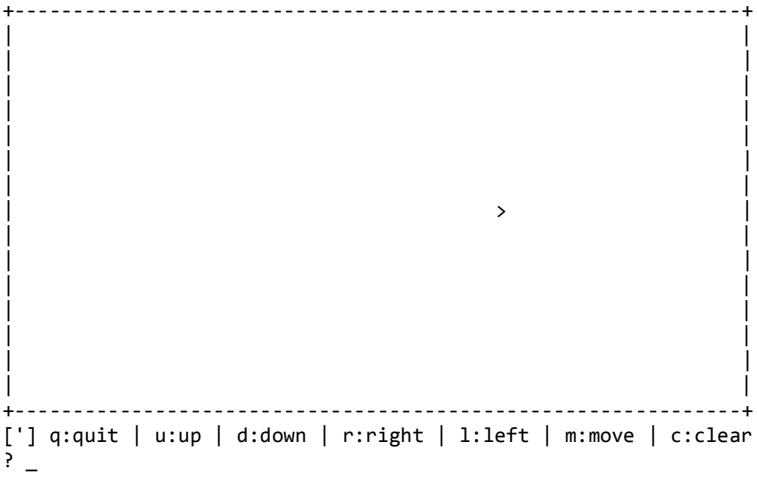
Os elementos básicos de Logo são a tartaruga e o seu mundo. A tartaruga tem (em sua barriga) uma caneta com a qual pode escrever no mundo. Os seguintes comandos podem ser dados para a tartaruga: erga a caneta (*up*), baixe a caneta (*down*), vire-se para esquerda (*left*), vire-se para a direita (*right*) ou mova-se *n* passos (*move*). Ao mundo, pode ser solicitado que todos os riscos feitos pela tartaruga sejam apagados. Abaixo, temos uma representação inicial deste mundo:



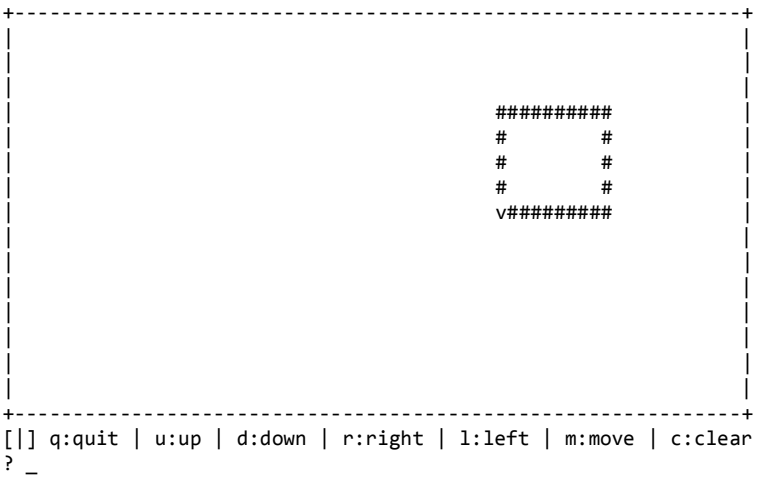
O mundo é representado por um quadro, onde a tartaruga (“›”) se encontra no meio. Nele, a representação da tartaruga (“›”) indica que ela está virada para a direita (ela poderia ser representada pelos símbolos “ᵛ”, “˂” e “^”, de acordo com a direção para a qual apontasse). Ela está com a caneta erguida, o que é representado pelo símbolo “`[']`” (se a caneta estivesse baixa e, portanto, a tartaruga estivesse pronta pra desenhar, isto seria representado por um “`[|]`”). Nesta situação, os comandos *up* (u), *down* (d), *right* (r), *left* (l) e *move* (m) podem ser dados para tartaruga. Qualquer desenho no mundo

pode ser apagado pelo comando *clear* (c). O comando q é usado se o usuário deseja sair do Logo.

Assim, o comando “m 10” resultaria em:



E a seqüência de comandos “d m 9 l m 4 l m 9 l m 4” resultaria em:



Objetivo

Sua tarefa é criar um programa que crie o mini-LOGO interativo descrito acima. A interface pode ser criada tanto em modo texto (como acima) quanto em modo gráfico.