

UNIVERSIDADE FEDERAL DE RORAIMA
PROFESSOR (A) : Herbert Oliveira Rocha .
DISCIPLINA DE LINGUAGEM DE PROGRAMAÇÃO.
ALUNO : Ibukun Chife Didier Adjitche

RESOLUÇÃO DA 1ª LISTA DE EXERCÍCIO

1º/ CONCEITO E DIFERENÇAS. ENTRE LINGUAGEM DE ALTO-NÍVEL E DE BAIXO-NÍVEL

São chamados de linguagem de Alto-Nível, linguagens de programação cujas , o nível abstração é elevado, longe do código comum das máquinas , mas, próximo à linguagem humana. O que se revela contrário ao que diz respeito às linguagens de Baixo-Nível. Uma linguagem de baixo nível seria uma linguagem baseada na arquitetura do computador, muito distante da linguagem humana, apenas bem compreendida pela máquina. Assim O programador de uma linguagem de alto nível não precisa conhecer características do processador, como instruções e registradores.

O PAPEL E A IMPORTÂNCIA DO COMPILADOR NO PROCESSO DE CRIAÇÃO DE PROGRAMAS DE COMPUTADOR

Um compilador é um programa que serve de tradução do código fonte de uma linguagem de alto nível para uma linguagem de programação de baixo nível (por exemplo, Assembly ou código de máquina.) Uma função importante do compilador diz respeito à detecção e informe de erros que detecta no programa fonte durante o processo de tradução.

2º) PROGRAMA ESCRITO NA LINGUAGEM C, C++, PYTHON E PERL

Os códigos se encontram no repositório LP_Lista_1-2017_IBUKUN como os Q02.

No terminal o comando **chmod +x Q02.(extensão)** foi aplicado para todos os arquivos para dar a permissão de execução.

Arquivos em C.

```
gcc Q02.c -o Q02.exe -lm // compilação
```

```
$ ./ Q02.exe // Execução.
```

Arquivos em C++.

```
gcc Q02.c -o Q02cpp.exe -lm // compilação
```

```
$ ./ Q02cpp.exe // Execução.
```

Arquivos em Perl.

```
$ ./ Q02.pl // Execução.
```

Arquivos em Python.

```
$ python3 Q02.py // Execução.
```

3º) PROGRAMA ESCRITO NA LINGUAGEM C, C++, PYTHON E PERL

Os códigos se encontram no repositório LP_Lista_1-2017_IBUKUN como os Q03.

No terminal o comando **chmod +x Q03.(extensão)** foi aplicado para todos os arquivos para dar a permissão de execução.

Arquivos em Python.

```
$ python3 Q03.py // Execução.
```

4º) Se achar o arquivo dessa questão no repositório LP_Lista_1-2017_IBUKUN como [downscript.sh](https://www.downscript.sh).

UNIVERSIDADE FEDERAL DE RORAIMA
PROFESSOR (A) : Herbert Oliveira Rocha .
DISCIPLINA DE LINGUAGEM DE PROGRAMAÇÃO.
ALUNO : Ibukun Chife Didier Adjitche

Temps total effectif : 7,1s

Téléchargés : 67 fichiers, 2,0M en 1,9s (1,04 MB/s)

5°) DESCRIÇÃO DAS CATEGORIAS DE LINGUAGENS DE PROGRAMAÇÃO E APRESENTAÇÃO DE DOIS EXEMPLOS

(A) **Imperativas:** As linguagens imperativas são orientadas a ações, onde a computação é vista como uma sequência de instruções que manipulam valores de variáveis. Elas são caracterizadas por tres conceitos : variaveis ,atribuição e iteração. Ex: C,Java,Perl,JavaScript, Visual BASIC, C++.

(B) **Funcional:** Ela se baseia no conceito de função. Pode-se pensar na programação funcional como simplesmente avaliação de expressões. Seu Funcionamento é da forma seguinte : O programador define uma função para resolver um problema e passa essa função para o computador.Uma função pode envolver várias outras funções em sua definição. O Computador funciona então como uma calculadora que avalia expressões escritas pelo programador através de simplificações até chegar a uma forma normal de aplicação de funções sob parametros. Ex: LISP, SCHEME, HASKELL, OCAML.

(C) **Logica:** Essa linguagem possui base na Lógica Matemática e qual o sentido de pegar esta base para a programação lógica é trazer o estilo da lógica matemática à programação de computadores. Ex: Prolog, Popler, Connivere QLISP.

(D) **Marcação/Hibrida:** são linguagens que estabelecem o formato , a maneira de exibicao e padroe de documento qualquer. Ex HTML, XML, SGML , XSLT...

6°) DEFINIÇÃO DE UMA GRAMÁTICA NO FORMATO BNF PARA UMA OPERAÇÃO DE MULTIPLICAÇÃO E ESPONENCIAÇÃO

```
<PROGRAM> ::= <MAIN>
<MAIN> ::= BEGIN <STMT_LIST> END
<STMT_LIST> ::= <STMT> | <STMT_LIST>
<STMT> ::= <VAR> = <DIGIT> ; | <VAR> = <DIGIT> ; <STMT>
<STMT> ::= <VAR> = <EXPR> ; | <VAR> = <EXPR> ; <STMT>
<EXPR> ::= <VAR> | <VAR> <OP> <VAR> | (<VAR> <OP> <VAR>) <OP> <EXPR>
<VAR> ::= A | B | C
<OP> ::= * | ^
<DIGIT> ::= [0-9]+
```

7°) DEFINIÇÃO DE UM PARADIGMA DE PROGRAMAÇÃO

Um paradigma de programacao é a visao, o conceito pelo qual o programador projeta et implementa um projeto na vida pratica et como ele usa os elementos da linguagem.

8°) APRESENTAÇÃO DE UM AUTÔMATO PARA RECONHECER UMA COMPARAÇÃO ENTRE NÚMEROS REAIS

RegEx: (-?[0-9]+(\\.[0-9]+)?) (=|<>|[<>]=?) (-?[0-9]+(\\.[0-9]+)?)

NB: A ESQUEMA SE APRESENTAR NO REPOSITORIO COMO Q08.

9º) DÉFINIÇÃO DA ANÁLISE SEMÂNTICA E DESCRIÇÃO DAS CATEGORIAS

A semântica trata da análise do significado das expressões, das instruções e das unidades de programa. A semântica é importante para que os programadores saibam precisamente o que as instruções de uma linguagem fazem.

Semântica Operacional: Descreve o significado de um programa ao executar suas instruções em uma máquina, real ou simulada. As alterações que ocorrem no estado de uma máquina, quando determinada instrução é executada, define o significado desta. O estado de um computador é definido pelos valores de todos os seus registradores e de suas localizações de memória.

Semântica Axiomática: Método para provar a exatidão dos programas que mostra a computação, descrita por sua especificação. Cada instrução de um programa tanto é precedida como seguida de uma expressão lógica que especifica restrições a variáveis. As expressões lógicas são usadas para especificar o significado das instruções. As restrições são descritas pela notação do cálculo de predicados.

Semântica Denotacional: Ela se baseia na teoria de função recursiva. Para cada entidade da linguagem deve ser definido tanto um objeto matemático como uma função que relacione instâncias daquela entidade com as deste. Os objetos representam o significado exato de suas entidades correspondentes. A dificuldade no uso deste método está em criar os objetos e as funções de correspondências. Os objetos matemáticos denotam o significado de suas entidades sintáticas correspondentes.

10º) DETERMINAÇÃO DAS PRÉ-CONDIÇÕES DOS SEQUENTES PROGRAMAS

PROGRAMA – A $\{y > 2.3\}$ $x = 122 * y - 144$ $\{x > 144\}$	$x > 144 \Rightarrow 122 * y - 144 > 144$ $y > (144 + 144) / 122$ $y > 2.3$ $\{y > 2.3\}$	PROGRAMA – B $\{x < 9\}$ $y = 5 * x - 5$ $x = y + 5$ $\{x < 45\}$	$x < 45 \Rightarrow y + 5 < 45 \Rightarrow y < 40 \Rightarrow$ $5 * x - 5 < 40 \Rightarrow x < (40 + 5) / 5$ $x < 9$ $\{x < 9\}$
PROGRAMA – C $\{y > 4\}$ if (x < 200) y = y + 2; else y = y - 2; $\{y > 2\}$	$y > 2 \Rightarrow y + 2 > 2 \vee y - 2 > 2$ $y > 0 \vee y > 4$ uma pré-condição que satisfaz o programa é $\{y > 4\}$ pois um valor de y entre]0,4] não satisfaz a pré-condição.	PROGRAMA – D $\{i \leq N\}$ while i < N do i = i + 1 end $\{i == N\}$	Temos três possibilidades: *i==N: o laço não será executado mas a pós-condição será atendida. *i>N Não execução do laço inclusive a pois-Condicion não atendida. *i<N o laço será executado inumeraveis vezes até atender a pós-condição que será a sua parada.
PROGRAMA – E $i = 1; sn = 0; n = 32767;$ $\{a \leq 0\}$ while (i <= n){ sn = sn + a; i = i + 1; } $\{sn == n * a\}$	Analisaremos o programa em três casos: *i<n: Se i<n o laço será executado n vezes: 1º) $sn + a == n * a \Rightarrow sn == a(n-1)$ pois antes o laço $sn=0$ podemos dizer $0 == a(n-1)$. Isto posto que $a=0$ pois que $n-1 \neq 0$. 2º) notamos que seja o valor de a a pré-condição será sempre verificada. Ex: si $a=0$ o laço será executado n vez ao final dara 0 como sn. Se a for 1 o laço será inclusive executado a*n o que será realmente o valor equivalente da soma sucessiva do a no laço. Concluimos que uma pré-condição será $\{a \in \mathbb{R}\}$. *i=n o laço será executado apenas uma vez. E obviamente o sn será igual ao valor de a incrementado uma vez. E a pré-condição será atendida que seja $\{a \in \mathbb{R}\}$.		

UNIVERSIDADE FEDERAL DE RORAIMA
PROFESSOR (A) : Herbert Oliveira Rocha .
DISCIPLINA DE LINGUAGEM DE PROGRAMAÇÃO.
ALUNO : Ibukun Chife Didier Adjitche

	<p>*$i > 0$ o laço não será executado mas a pré-condição que deixará a pos-condição satisfeita é a atribuição do $a = 0$, utilizando os valores de base $sn=0$ e $n = 32767$. Das três possibilidades do programa E para que a pos-condição seja satisfeita diremos que $\{a \leq 0\}$ será a pré-condição</p>
--	--