P1 – Compiladores!!

VOCE NÃO COLOCA MATCH DO ANALISADOR LEXICO!!!!

NAO TEM MATCH !!

o match é um operador sintetico, ele é um varredor de caractéries

o match é um interface entre o analisador sintático e lexico.

Você usa LEXEME!!!!!!!!!

analisador lexico não sabe que analisador sintático existe!!!!!!!!

ps: string em C termina em 0.!!

Não precisa declarar as bibliotecas, só faça a função!!!

Questão do Ponto flutuante é um boa questão para a prova!!!!!!!!1

1) Apresente uma expr . Regular para um padrão de identificadores, que tenha letra como prefixo, podendo vir com um ou mais underscore (‘\_’), prefixando esta mesma letra, e que tenha como sufixo zero ou mais letra, underscore ou digito.

→ já esta na cara que existe um alfabeto mínimo de prefixo (a-z, A-Z).

→ tenha letra com prefixo, podendo vir com um ou mais underscore.

→ Este padrão, começa com 0 ou mais underscore prefixando uma letra

**R: ID = \_ \*[a-z A-Z] [ \_ a-z A-Z 0-9]\***

2) Implemente uma função predicado ( em c ) para emular o padrão regular do exercício anterior :

R:

int isId( FILE \*tape)

{

int head, n\_und = 0;

while ( (head = getc(tape)) == ‘\_’) n\_und++;

if (isalpha (head)){

while(isalnum(head = getc(tape)|| head == ‘\_’);

ungetc (head,tape);

return ID;

}

ungetc(head,tape);

while (n\_und){

ungetc(‘\_’,tape);

nud--;

}

return 0;

}

3) Apresente uma expr. regular, para um padrão de ponto flutuante C.

→ O ponto está em apostrofo, é qualquer codigo asc diferente de 0 do unix.!!

→ Interrogação significa que pode vir com ou sem sinal!!!

→ o lado destacado não é preciso

R:Float=(([1-9][0-9]\*|0)‘.’[0-9]+ |‘.’[0-9]+ )(E(‘+’|’-’)?[0-9]+)?

| ( [1-9] [0-9]\* | 0) E (‘+’| ‘-”) ? [0-9]+

FltPoint = ( uint ‘.’ digit\* | ‘.’ digit) exp? | uint exp

ELE NÃO PEDE o PREDICADO EM PROVA!!!!!!!!!!!!!!!!

Questão For em C

Tem ambiguidade no For???

Forstmt → for (expr; expr; expr) stmt

stmt → ‘{‘ stmt {stmt } ‘}’| E | IF …. | While…. | expr;

forsmt

for ( expr ; expr ; expr ) stmt

Em vista não temos ambiguidade, em posto de vista de sintaxe !!!!!!!

Qual a ordem de execução dessas expr :

for ( expr ; expr ; expr ) stmt n

1 2 4 3 0

5 7 6 1

………………………………. (3n+2) (3n+4) (3n+3)

→ traduzindo do for para While ( tratando de regra semantica, de forma informal)

for ( expr ; expr ; expr ) stmt

1 2 3

expr1;

while(expr2) {

stmt

expr3;

}

Questão 18 da LISTAAAAAA – passar para forma EBNF \_ Lauden

stmt → ifhead elifseq elseclose endif | other

ifhead → if expr then stmt

elifseq → elif expr then stmt elifseq| E

elseclose → else stmt | E

–

EBNF

| ifhead → if expr then stmt

| elifseq → {elif expr then stmt} // acontece 0 ou mais vezes

| elseclose → [eles stmt]

R: stmt → if expr then stmt {elif expr then stmt}[eles stmt] endif|other

Questão 11

Slash //////////

/\* < comment> \*/

void stmt( comments (FILE \* buff)

{ int head;

if (head = getc(tape)) == ‘/’) {

if( (head = getc(tape) ) == ‘\*’){

\_while: while ( (head = getc ( tape) ) != ‘\*);

if ( (head = getc(tape) != ‘/’) goto \_while;

return;

} ungetc (head,tape);

}

}