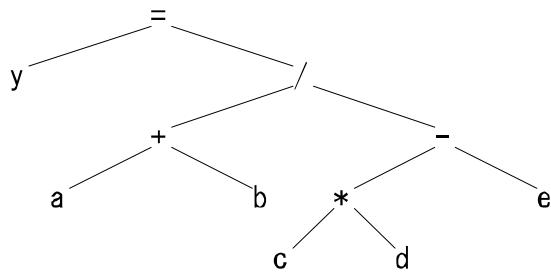


## コンパイラ

### 12回目 中間言語とコード生成 課題

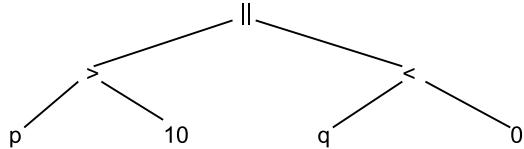
課題1 構文解析の結果、出力された式  $y = ((a + b) / ((c * d) - e))$  の構文木を下に示す。この構文木の後置記法表現、三つ組表現、四つ組表現を示しなさい。



課題2 次に論理式とその構文木を示す。表の命令をもつスタック機械でこれを評価するための機械語コードを作成しなさい。最初、スタックは空であり、最終的にこの論理式の結果である真(1)または偽(0)のみがスタックに残ることとします。

(論理式)  $p > 10 \parallel q < 0$

(構文木)



学籍番号

氏名

解答欄 :

## (スタック機械命令)

命令 (オペコード)	パラメータ (オペランド)	機能	意味
PUSH	para	プッシュ	paraが変数であればその中の値をスタックへ積み、paraが数値であればその値をスタックへ積む。
POP		ポップ	スタックのトップの値を取り出す。
ASSIGN	var	アサイン	スタックのトップをポップし、その値を変数varへ書き込む。
JUMP	label	分岐	ラベルlabelへ飛ぶ。
FJUMP	label	分岐	スタックのトップをポップし、0であればラベルlabelへ飛ぶ。
TJUMP	label	分岐	スタックのトップをポップし、0でなければラベルlabelへ飛ぶ。
INV		符号反転	スタックのトップをポップし、その値の符号を反転する。結果をスタックのトップへプッシュする。
ADD		加算	スタックのトップと2番目をポップし、それらを加算する。結果をスタックのトップへプッシュする。
SUB		減算	スタックのトップと2番目をポップし、2番目からトップを減ずる。結果をスタックのトップへプッシュする。
MULT		乗算	スタックのトップと2番目をポップし、それらを乗算する。結果をスタックのトップへプッシュする。
DIV		除算	スタックのトップと2番目をポップし、2番目をトップで割る。結果をスタックのトップへプッシュする。
MOD		剰余	スタックのトップと2番目をポップし、2番目をトップで割ったときの余りを計算する。結果をスタックのトップへプッシュする。
GTOP	>		スタックのトップと2番目をポップし、2番目がトップより大きければ1、そうでなければ0をスタックのトップへプッシュする。
GEOP	>=		スタックのトップと2番目をポップし、2番目がトップ以上であれば1、そうでなければ0をスタックのトップへプッシュする。
LTOP	<		スタックのトップと2番目をポップし、2番目がトップより小さければ1、そうでなければ0をスタックのトップへプッシュする。
LEOP	<=		スタックのトップと2番目をポップし、2番目がトップ以下であれば1、そうでなければ0をスタックのトップへプッシュする。
EQOP	==		スタックのトップと2番目をポップし、2番目とトップが等しいならば1、そうでなければ0をスタックのトップへプッシュする。
NEOP	!=		スタックのトップと2番目をポップし、2番目とトップが等しくないならば1、そうでなければ0をスタックのトップへプッシュする。
ANDOP	&&		スタックのトップと2番目をポップし、2番目とトップで一方または両方が0ならば0、そうでなければ1をスタックのトップへプッシュする。
OROP			スタックのトップと2番目をポップし、2番目とトップで両方が0ならば0、そうでなければ1をスタックのトップへプッシュする。

課題3 次に各コードを上表の命令をもつスタック機械で実行するための機械語コードを作成しなさい。最初、スタックは空とします。

- (1) if (a > b) c = a;
- (2) if (a == 0) b = 1; else b = 2;
- (3) while(a > 0) a = a - 1;

学籍番号

氏名

解答欄 :