問２　コンパイラの処理内容（ソフトウェア）　(H22春 FE 午後問2)

【解答】

［設問１］　オ
［設問２］　イ
［設問３］　ウ
［設問４］　ウ

【解説】

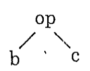
コンパイラの構文解析を題材にした問題である。２分木，深さ優先検索，後置表記法（逆ポーランド表記法）が理解できていれば容易に解答できる。構文解析手法，構文記法を理解できていない場合でも，本文中に解析方法，解析順序が書かれているので，その順番に求めれば解答を導き出せる。ただし，設問２の考え方は難しく，整理して解かないと時間を要する。問題にある構文解析手法についてはよく出題されるので，確実に習得しておくとよい。

［設問１］

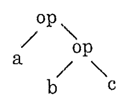
与えられた２項演算子の式が，深さ優先で探索するどの２分木に該当するかが問われている。探索順序は，２分木において深さ優先で節を探索し，帰り掛けにその親の節の演算子を評価するとしている。与えられた式の計算順序から，構文木である２分木を考えてみる。

式：$\underbrace{\overbrace{\text{a op }\underbrace{\text{(b op c)}}_{①}}^{②}\text{ op d}}_{③}$

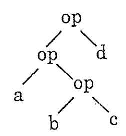
この式において括弧が先に計算されるので，①の部分の(b op c)が２分木の最下部となる。最下部の２分木は，子の節がｂとｃ，親の節が演算子 op となる。



次に，②の部分の a op (b op c)が計算される。２分木は，左の子の節がａ，右が(b op c)の部分木，親の節が op となる。



最後に，③の部分の a op (b op c) op d が計算される。右が a op (b op c)の部分木，左の子の節にｄ，親の節が op となる。



したがって，（オ）が正解である。
ちなみに，各選択肢の構文木から式に直すと，次のようになる。
ア：a op (b op (c op d))
イ：(a op b) op (c op d)
ウ：(a op d) op (b op c)
エ：a op ((b op c) op d)

［設問２］

与えられた２項演算子の式を見ただけでは，どの演算から評価すればよいのか分からない。この式を演算子に着目して分解し，〔式の構文規則〕に当てはめて考えていく。
まず，左の op1 に注目して考えてみる。op1 が含まれる定義は「式」の定義の「式 op1 項」で，op1 の左側は「式」と定義されている。式の op1 の左側は「a」である。「式」→「項」→「因子」→「名前」→「a」の順にそれぞれ含まれて定義されているので，対応している。次に，op1 の右側は「項」と定義されているが，この「項」の範囲は次の①～③の三つのパターンが考えられる。

式：$\underbrace{\text{a}}_{\text{式}}\text{ op1 }\underbrace{\overbrace{\underbrace{\text{b}}_{①}\text{ op2 c}}^{②}\text{ op2 (d op1 e)}}_{\text{右側の項？ ③}}$

「項」の範囲が①の場合は，左の op2 の左側を「a op1 b」として演算を評価することとなり，その op1 を含む定義は「式」だけである。しかしながら，op2 が含まれる定義は「項」の定義の「項 op2 因子」で，op2 の左側は「項」が定義されている。「項」は「因子」と「項 op2 因子」が定義され，「因子」は「名前」と「(式)」しか定義されておらず，「式」の定義がない。したがって，①の「項」の範囲は適切ではない。

式：$\underbrace{\text{a op1 b}}_{\text{項でない}}\text{ op2 …}$

「項」の範囲が②の場合は，右の op2 の左側を「a op1 b op2 c」として演算を評価することになるが，op1 が含まれているので「式」の定義が入ることになる。op2 の左側は「項」が定義され，前述と同様に「項」は「式」の定義を含まないので，②の「項」の範囲も適切ではない。

式：$\underbrace{\text{a op1 b op2 c}}_{\text{項でない}}\text{ op2 …}$

「項」の範囲が③の場合は，左の op1 の右側を「b op2 c op2 (d op1 e)」として演算を評価することになる。op1 の右側は「項」が定義されている。左の op2 に着目して「項」を考えると，op2 の右側は「因子」となり，「因子」は「名前」か「(式)」に定義されているだけなので，式の op2 の右側である「c op2 (d op1 e)」には該当しない。

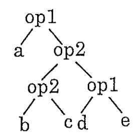
式：$\underbrace{\text{a}}_{\text{式}}\text{ op1 }\underbrace{\overbrace{\text{b}}^{\text{項}}\text{ op2 }\overbrace{\text{c op2 (d op1 e)}}^{\text{因子でない}}}_{\text{項}}$

次に，右の op2 に着目して「項」を考えると，「項」の定義に含まれる「項 op2 因子」に該当する。右の op2 の左側の「b op2 c」は「項」に該当し，「項」→「因子」→「名前」→「b」，「c」の順にそれぞれ含まれて定義されているので，対応している。右の op2 の右側の「(d op1 e)」は「因子」に該当し，「因子」は「(式)」を，「式」は「式 op1 項」を含み，また，「式」→「項」→「因子」→「名前」→「d」，「e」の順にそれぞれ定義されているので，対応している。

式：$\underbrace{\text{a}}_{\text{式}}\text{ op1 }\underbrace{\overbrace{\text{b op2 c}}^{\substack{\text{項 op2 因子}\\ \text{項}}}\text{ op2 }\overbrace{\text{(d op1 e)}}^{\substack{\text{(式 op1 項)}\\ \text{(式)}\\ \text{因子}}}}_{\text{項}}$

これらから，演算順序を考えると次のようになり，この演算順序を表している構文木は（イ）である。

式：$\underbrace{\text{a op1 }\overbrace{\underbrace{\text{b op2 c}}_{(1)}\text{ op2 }\underbrace{\text{(d op1 e)}}_{(1)}}^{(2)}}_{(3)}$

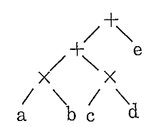


なお，この設問は設問３の設問文に「加算＋と乗算×は，それぞれ設問２の演算子 op1 と op2 に対応し，演算の優先順位や結合規則は，〔式の構文規則〕に従うものとする」とあり，単純に式：a＋b×c×(d＋e)において，乗算×が加算＋よりも優先順位が高いとして構文木を探すのと同じことになる。また，〔式の構文規則〕において op1 が「式」の規定に含まれ，op2 が「項」の規定に含まれるのは，優先順位として op2 の方が高いことを意味している。

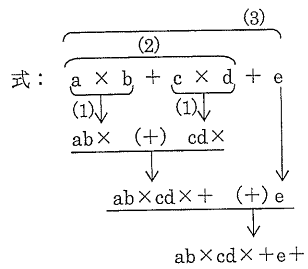
［設問３］

後置表記法（逆ポーランド表記法）は，コンピュータ内部での算術式の表記方法で，演算子を演算対象の後に記述して表記する。設問に沿って求める式の計算順序を考え，演算子 op を加算＋や乗算×に置き換えて構文木で表すと，次のようになる。

式：$\underbrace{\overbrace{\underbrace{\text{a × b}}_{(1)}\text{ ＋ }\underbrace{\text{c × d}}_{(1)}}^{(2)}\text{ ＋ e}}_{(3)}$



これを，深さ優先順序で探索すると，ab×cd×＋e＋となり（ウ）が正解である。
なお，後置表記法は演算子を後ろに置いて演算する順に表記するもので，表記手順は次のとおりである。



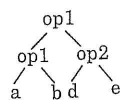
［設問４］

後置表記法（逆ポーランド表記法）は，スタックを利用して求めることができる。設問に沿って，式 a＋b＋c×d の＋と×をそれぞれ op1，op2 に置き換える。

式：　a op1 b op1 c op2 d

〔式の構文規則〕から，演算順序は次のとおりである。

式：$\overbrace{\underbrace{\text{a op1 b}}_{(1)}\text{ op1 }\underbrace{\text{c op2 d}}_{(1)}}^{(2)}$



なお，後置表記法の表記手順は次のとおりである。