令和５年６月６日

令和６年度大分大学理工学部第３年次編入学試験問題【筆記試験】

(共創理工学科　知能情報システムコース)

試験問題の配点は，1．が 50 点，2．が 50 点です。

1．次の英文を読んで各問いに答えなさい。

profound: 深い　　transaction: 取引　　decentralized: 分散した
ledger: 台帳　　deployment: 配置，展開　　benign: 害のない
spawn: ～を発生させる　　non-fungible token: 非代替性トークン

(1) 下線部(A)が指す内容を日本語で答えなさい。
(2) 下線部(B)を和訳しなさい。
(3) 下線部(C)の具体的な例をすべて本文中の表記そのままで答えなさい。
(4) ブロックチェーンと AI のメタバースとの融合で生まれる新たな課題について，本文中に記述がある例を日本語で答えなさい。

2. 以下は，簡単な数式の表現方法や計算方法について書かれた文章である。文章中及び文章から参照している図の中の下線部(A)～(J)を埋めなさい。

数式やプログラムの式を記述する方法の一つに，演算子を被演算数の後に記述する後置記法がある。後置記法における式の構文とその値は，以下の規則に従って決まる。

- 被演算数は式であり，その値はその数そのものである。
- 二つ続く式の後ろに演算子を置いた並びは式であり，その値は，その二つの式の値に対してその演算子を適用した結果となる。

なお，ここでは簡単のため，被演算数は一桁の正の整数のみを，演算子は加算を表す+と乗算を表す*のみを考える。例えば，一般的にプログラム等で用いられる記法で記述された式 1+2 を後置記法で記述すると 12+となり，2*5+6*7 を後置記法で記述すると (A) となる（結合の強さは加算よりも乗算のほうが上とする)。また後置記法で記述された式 397+* の値は (B) となり，13+45+*の値は (C) となる。

中置記法（演算子を二つの被演算数の間に記述する記法）から後置記法への並べ替えを行うアルゴリズムには，スタックを用いた以下のようなものがある。なお，ここでは括弧や空白文字を含む式は考えない。

1. スタックにどの演算子よりも結合の弱い演算子（ここでは¥0 とする）を積む
2. 中置記法の式を左から読みながら
   - 被演算数を読んだら，それをそのまま出力する
   - 演算子を読んだら，スタックの上にそれより結合の強いあるいは結合の強さの等しい演算子があればあるだけそれをスタックから降ろして出力してから，読んだ演算子をスタックに積む
3. 式の終わりになったら，スタックの先頭要素が¥0 になるまでスタックから降ろして出力する

このアルゴリズムを実装した C 言語のプログラムは次ページのようになる。このプログラムを実行すると (J) が出力される。

```
#include <stdio.h>

char stack[8]; /* スタック */
int sp = 0;

void push(char c) { /* スタックに c を積む */
  stack[sp++] = c;
}

char pop() { /* スタックから降ろし，その値を返す */
  return stack[ (D) ];
}

char peek() {  /* スタックの先頭を返す */
  return stack[ (E) ];
}

int priority(char c) {
  switch (c) {
  case '*': return 1;
  case '+': return 0;
  default: return -1;
  }
}

int main() {
  char expr[] = "5+6*7+8";
  char c;
  int i=0;

  push('¥0');
  while (c = expr[i++]) {
    if (c == '*' || c == '+') {
      while (priority( (F) ) <= priority(peek())) {
        printf("%c", (G) );
      }
      (H) ;
    } else {
      printf("%c", c);
    }
  }
  while (peek() != '¥0') {
    printf("%c", (I) );
  }
  printf("¥n");
}
```