

学籍番号	学年	氏名
------	----	----

計算数学 (2004 年度後期, 1 月 26 日実施) 期末試験略解 (松本 眞)

問題 1. 次のプログラムについて、以下の問題に答えよ。

```
#include <stdio.h>
#define BUNBO 7

unsigned long
next_state(unsigned long x)
{
    return((10 * x) % BUNBO);
}

main(void)
{
    unsigned long state, kosu;
    int i;
    printf("漸化式による数列を計算します。初期値 a を入力ください。 \n");
    printf(" a=");
    scanf("%d",&state);
    printf("何個もとめますか? 個数を入力ください。 \n 桁数=");
    scanf("%d",&kosu);
    for (i=0; i<kosu; i++) {
        printf("%d ", state);
        state = next_state(state);
    }
    printf("\n");
}
```

1. このプログラムで初期値 1 を入力し個数 3 を入力するとどのような計算を行いどのような出力を行うか。簡潔に説明せよ。

入力により、state に 1 が入り、kosu に 3 が入る。その後、for 文で kosu 回次を繰り返す。  
 「state を整数値で出力し、state に next\_state(state) を代入する。」  
 ここで、プログラム上部で定義された関数 next\_state は、もらった値を 10 倍して 7 で割った余りを求めるものである。よって、state の値は

$$1 \rightarrow 10 \equiv 3 \rightarrow 30 \equiv 2$$

と変化する。よって出力は  
 1 3 2  
 となる。

2. 漸化式

$$x_{n+1} = \begin{cases} x_n/2 & x_n \text{ が偶数} \\ 3x_n - 1 & x_n \text{ が奇数} \end{cases}$$

で定まる数列を求めるプログラムに変更するには、関数 next\_state をどのように書き換えれば良いか、記せ。

```
unsigned long
next_state(unsigned long x)
{
    if (x % 2 == 0) {return(x/2);}
    else {return(3*x-1);}
}
```

3. 上の 2. で書き換えたプログラムを実行し、初期値 5 を入力し、個数 7 を入力すると出力結果はどうなるか、記せ。なぜそうなるか、簡単な説明を沿えよ。

結局、偶数なら 2 で割り、奇数ならば 3 倍して 1 を引く、という操作を 7 回繰り返すことに他ならない。  
5 14 7 20 10 5 14  
なる出力を得る。

問題 2. 次のプログラムについて、以下の問題に答えよ。

```
#include <stdio.h>

void bunkatsu(int n, int depth)
{
    int i,j;

    printf("+%d\n",n);
    for (i=1; i<=n-1; i++) {
        for (j=0; j< depth; j++) printf(" ");
        printf("+%d", i);
        bunkatsu(n-i,depth+1);
    }
}

main(void)
{
    int n;
    printf("n 個の元の分割を全て求めます。n を入力ください。 \n");
    printf(" n=");
    scanf("%d",&n);
    bunkatsu(n,0);
}
```

- このプログラムは再帰的呼び出しを行うものである。実行して 3 を入力すると、main 関数から関数 bunkatsu が呼び出されるが、このとき bunkatsu に渡される値は何と何か。  
n に 3 が入り、bunkatsu(n,0) が呼び出されるのだから 3 と 0 が渡される。
- bunkatsu(2,0) が呼び出されたとき、このプログラムはどのように動作しどのような出力を行うか、順をおって説明せよ。
  - bunkatsu(2,0) が呼ばれると、まず printf 文で +2(改行) が出力される。  
その後、i=1 から n-1=1 まで i を動かしながら次の動作をする。(この場合 i=1 しかおきない)
  - depth 回スペース二個分を打つ。(この場合 depth=0 だから何もしない。)
  - +1 を出力(改行なし)
  - bunkatsu(1,1) が呼ばれる。
  - printf 文で +1 を出力
  - n-1=0 なので次の for 文は素通りして終わり、bunkatsu(1,1) が終了
  - bunkatsu(2,0) に戻るが、そのまま for 文を抜けて終了
  - 結果  
+2  
+1+1  
を得る。

問題 3. 次のプログラムについて、以下の設問に答えよ。

```
#include <stdio.h>
main(void)
{
    long i,n;
    double a=0, b;
    printf("何回繰り返す?");
    scanf("%d",&n);
    b=1;
    for (i=1; i<=n; i++) {
        a += 1 / b;
        b = b * i;
    }
    printf("e is nearly %.20f\n", a);
}
```

1.  $b=b*i$  を実行した後で、次の二つのループ不変性質が成立することを証明せよ。

$$(A) b = i! \quad (B) a = \sum_{j=0}^{i-1} 1/j!$$

帰納法を用いる。(A) について：最初のループ実行後には  $b=1, i=1$  で成立している。ある時点で成立しているとして、ループをもう一回まわる。ループの先頭では帰納法の仮定から  $b=(i-1)!$  となっている。その後  $b=b*i$  の実行により  $b=i!$  となる。

(B) について。最初のループ実行後には  $a=1, i=1$  で成立している。ある時点  $i-1$  で成立しているとして、ループをもう一回まわると

$$a = \sum_{j=0}^{i-1} 1/j! + 1/i!$$

となり、性質が保たれる。

2. この時の  $e$  の近似誤差を、繰り返しの回数  $n$  を用いて  $O(f(n))$  の形に表せ。きっちりとした証明を与えよ。

ループを抜けたときに  $i=(n+1)$  となってぬけるのだが、それはぬける直前に  $i$  が 1 増やされるのであって、上のループ不変性質が成立している段階では  $i=n$  である。

(最初の略解では、ここを間違えて、以下正解を  $O(\frac{1}{(n+1)!})$  としてしまいました。何度も間違えて申し訳ありません。心からお詫び申し上げます。ご指摘いただいた斉藤睦夫さんに感謝いたします。松本 眞)

$$a = \sum_{j=0}^{n-1} 1/j!$$

となる。近似誤差の評価は中間試験と同様の方法で  $O(\frac{1}{n!})$  となる。