

問題は 4 問ある。それとは別におまけ問題が 2 問ある。

問題 1. プログラム 1-1, 1-2 は、いずれも、非負整数 n を入力し $\sum_{i=1}^n i$ と $\sum_{j=1}^n \sum_{i=1}^j i$ と $\sum_{k=1}^n \sum_{j=1}^k \sum_{i=1}^j i$ をこの順に出力するプログラムである。プログラム 1-1 は

$$\sum_{i=1}^n i = \frac{n(n+1)}{2}, \quad \sum_{j=1}^n \sum_{i=1}^j i = \frac{n(n+1)(n+2)}{6}, \quad \sum_{k=1}^n \sum_{j=1}^k \sum_{i=1}^j i = \frac{n(n+1)(n+2)(n+3)}{24} \quad (\Delta)$$

を使って計算し、プログラム 1-2 は地道に足し算を繰り返して計算する。

空欄を埋めよ。

——— プログラム 1-1: $\sum_{i=1}^n i$ と $\sum_{j=1}^n \sum_{i=1}^j i$ と $\sum_{k=1}^n \sum_{j=1}^k \sum_{i=1}^j i$ を計算するプログラム (1) ———

```
#include <stdio.h>

int
main()
{
    unsigned long    n;
    scanf("%lu", &n);
    unsigned long    s1 = n * (あ) / 2;
    unsigned long    s2 = s1 * (い) / 3;
    unsigned long    s3 = s2 * (う) / 4;
    printf("%lu %lu %lu\n", s1, s2, s3);
    return 0;
}
```

——— プログラム 1-2: $\sum_{i=1}^n i$ と $\sum_{j=1}^n \sum_{i=1}^j i$ と $\sum_{k=1}^n \sum_{j=1}^k \sum_{i=1}^j i$ を計算するプログラム (2) ———

```
#include <stdio.h>

int
main()
{
    unsigned long    n;
    scanf("%lu", &n);
    unsigned long    s1 = 0;
    unsigned long    s2 = 0;
    unsigned long    s3 = 0;
    unsigned int      i;
    for (i = 1; i <= n; i++) {
        s1 += (え);
        s2 += (お);
        s3 += (か);
    }
    printf("%lu %lu %lu\n", s1, s2, s3);
    return 0;
}
```

問題 2. 平面上の点 (x, y) を原点を中心に角度 θ だけ反時計回りに回転させて移る点を (x', y') とおくと、

$$\begin{aligned} x' &= x \cos \theta - y \sin \theta \\ y' &= x \sin \theta + y \cos \theta \end{aligned} \quad (\curvearrowright)$$

と表すことができる。

プログラム 2-1, 2-2 は $(\curvearrowright)$ を C の関数として実装したものである。

- いずれのプログラムでも、角度はラジアンであり、double 型で表す。また、平面上の点を表すのに構造体 struct point を使い、そのメンバの意味はコメントに書かれた通りである。
- プログラム 2-1 とプログラム 2-2 は、関数のインターフェースが異なる。それぞれ、コメントとして書かれた通りである。

プログラム 2-1: 平面上の点を回転させる関数 (1)

```
#include <math.h>

struct point {
    double      x, y;          /* x: x 座標, y: y 座標 */
};

/* p: 元の点の座標, theta: 回転角, 返値: 移った先の点の座標 */
struct point
rotate(struct point p, double theta)
{
    double      x = (き);
    double      y = (く);
    double      cos_theta = cos(theta);
    double      sin_theta = sin(theta);
    struct point q;
    (け) = x * cos_theta - y * sin_theta;
    (こ) = x * sin_theta + y * cos_theta;
    return q;
}
```

プログラム 2-2: 平面上の点を回転させる関数 (2)

```
#include <math.h>

struct point {
    double      x, y;          /* x: x 座標, y: y 座標 */
};

/* p: 元の点の座標を指すポインタ, theta: 回転角, q: 移った先の点の座標の格納先を指すポインタ */
void
rotate(const struct point *p, double theta, struct point *q)
{
    double      x = (さ);
    double      y = (し);
    double      cos_theta = cos(theta);
    double      sin_theta = sin(theta);
    (す) = x * cos_theta - y * sin_theta;
    (せ) = x * sin_theta + y * cos_theta;
}
```

問題 3. プログラム 3-1 は、文字列の最後の文字を求める C の関数を実装しようと意図しながら間違えたプログラムである。どこが間違っているか指摘し、その間違いでどのような不具合が生じる可能性があるかを解説し、修正案を提案せよ。

プログラム 3-1: 文字列の最後の文字を求めようとして間違えたプログラム

```
#include <string.h>

/* 文字列が空でなければ最後の文字の文字コードを返し、空ならばヌル文字の文字コードを返す */
int
lastchr(const char *str)
{
    int      c = '\0';
    int      len = strlen(str);
    if (len > 0) {
        c = str[len - 1];
    }
    return c;
}
```

問題 4. たくさんの数の相乗平均を計算する C の関数が必要になり、今泉慎太郎と西園寺守は、それぞれ、プログラム 4-1, 4-2 のように書いた。数が多い場合は西園寺守版のほうが安全であるが、その理由を説明せよ。

プログラム 4-1: 相乗平均を計算する関数（今泉慎太郎版）

```
#include <stdlib.h>
#include <math.h>

double
gmean(const double *a, size_t size)
{
    size_t i;
    double prod = 1;
    for (i = 0; i < size; i++) {
        prod *= a[i];
    }
    return pow(prod, 1.0 / size);
}
```

プログラム 4-2: 相乗平均を計算する関数（西園寺守版）

```
#include <stdlib.h>
#include <math.h>

double
gmean(const double *a, size_t size)
{
    size_t i;
    double sum = 0;
    for (i = 0; i < size; i++) {
        sum += log(a[i]);
    }
    return exp(sum / size);
}
```

おまけ問題 1 (自信のない人のみ). 問題 1 の (Δ) を証明せよ。

おまけ問題 2 (自信のない人のみ). 自分でプログラミング言語を作るとしたらどんな名前をつけたいか。つけたい名前を記し、その名前の由来も説明せよ。