



知能情報工学演習I 第12回 (C言語第6回) 課題の回答

岩村雅一

masa@cs.osakafu-u.ac.jp

前回の課題1

- 球の体積を計算するマクロを作り、球の半径(小数とする)を入力したとき、球の体積を返すプログラムを作成しなさい。

```
#include<math.h>
#include<stdio.h>
#define V(r) 4.0/3.0*M_PI*r*r*r

int main(void){
    float i;
    printf("半径を小数で入力: ");
    scanf("%f",&i);

    printf("半径%fの球の体積は%fです。
    ¥n",i,V(i));

    return(0);
}
```

マクロ中の
rをiに置き換える

課題1で実際にあった間違い

- 体積の公式を間違っている。
 - $\frac{4}{3} \pi r^2$
- 係数が整数扱い ($\frac{4}{3}=1$) になっている
 - `#define V(r) 4/3*M_PI*r*r*r`
- 球の体積を計算するマクロを作っていない。
(題意に反する)
- 厳密に言えば題意に反する (OKとしました)
 - `#define volume(r) (4*r*r*r)/3`
 - `vol=volume(r)*3.14159265358979323846;`

間違いではないが...

- M_PIは math.h で定義されているので、インクルードすればよい。

```
#define M_PI 3.14159265358979323846
```

```
→ #include <math.h>
```

- 関連コメント:関数の中でパイを使う場合でも math.h は必要みたいです。

コメント(課題1)

- 今回一つ目はとりあえずなんとかかなりました。これをやることでやっと少しですがグローバル関数がわかりました。それよりも最近USBメモリを無くしてしまいブルーです。もうブルー畑中です。慰めてください.....
 - 心中お察し申し上げます

前回の課題2

□ 階乗(1からnまでの自然数の積)を計算する関数を作り、順列と組み合わせを表示しなさい

■ 順列

$${}_n P_r = n \cdot n - 1 \cdot n - 2 \cdots n - r + 1 = \frac{n!}{n - r !}$$

■ 組み合わせ

$${}_n C_r = \frac{{}_n P_r}{{}_r P_r} = \frac{n!}{r! \cdot n - r !}$$

前回の課題の回答例1 (for文を使った場合)

```
#include <stdio.h>

/* 階乗を計算する関数 */
int fact(int x) {
    int i, fact = 1;
    for(i = 2; i <= x; i++) {
        fact *= i;
    } fact = 1 * 2 * 3 * 4 * ...
    return(fact);
} 0!や1!もok
```

```
int main (void){
    int n, r, p, c;

    printf("n: "); scanf("%d", &n);
    printf("r: "); scanf("%d", &r);


    p = fact(n) / fact(n-r);
    c = fact(n) / fact(n-r) / fact(r);

    printf("nPr = %d\n", p);
    printf("nCr = %d\n", c);

    return(0);
}
```

前回の課題の回答例2 (関数の再帰的呼び出し)

main関数は説明
のために消去



```
#include <stdio.h>
```

```
/* 階乗を計算する関数 */
```

```
int fact(int x) {  
    if (x==1 || x==0) {  
        return(1);  
    } else {  
        return(x*fact(x-1));  
    }  
}
```

```
int main (void){  
    int n, r, p, c;  
  
    printf("n: "); scanf("%d", &n);  
    printf("r: "); scanf("%d", &r);  
  
    p = fact(n) / fact(n-r);  
    c = fact(n) / fact(n-r) / fact(r);  
  
    printf("nPr = %d\n", p);  
    printf("nCr = %d\n", c);  
  
    return(0);  
}
```


前回の課題の回答例2 (関数の再帰的呼び出し)

例: fact(2)の場合

```
#include <stdio.h>
```

```
/* 階乗を計算する関数 */
```

```
int fact(int x) {  
    if (x==1 || x==0) {  
        return(1);  
    } else {  
        return(x*fact(x-1));  
    }  
}
```

||
fact(1)
||
1

fact(1)の計算

```
/* 階乗を計算する関数 */
```

```
int fact(int 1) {  
    if (x==1 || x==0) {  
        return(1);  
    } else {  
        return(1*fact(1-1));  
    }  
}
```

前回の課題の回答例2 (関数の再帰的呼び出し)

```
#include <stdio.h>
```

```
/* 階乗を計算する関数 */
```

```
int fact(int x) {
```

```
    if (x==1 || x==0) {
```

```
        return(1);
```

```
    } else {
```

```
        return(x*fact(x-1));
```

```
    }
```

```
}
```

fact(4)

= 4 * fact(3)

= 4 * 3 * fact(2)

= 4 * 3 * 2 * fact(1)

= 4 * 3 * 2 * 1

課題2で実際にあった間違い

- 階乗を計算する関数がない。順列と組み合わせを計算する関数のみがある。
 - OKとしました。
- 順列の公式を間違っている

$${}_n P_r \neq \frac{n!}{r!}$$

間違いではないが...

- 階乗の関数を複数作っている人がいた。引数を変えれば、関数は1つでOK。

- 階乗: $N(n)/NR(n-r)$

$${}_n P_r = \frac{n!}{n-r!}$$

- 組み合わせ: $N(n)/R(r)/NR(n-r)$

$${}_n C_r = \frac{n!}{r! (n-r)!}$$

- 定石:

- 「 $d=e*a; e=d;$ 」は $e=e*a$ または $e*=a$ と書ける

コメント(その1)

- 課題2ではunsigned int を使っても対応しきれない順列が出てきました。(20P20 とか。)これらを表示する方法を調べてみようと思います。

(同様のコメント多数)

- 解決方法1: 階乗を使用しない。

$${}_n P_r = n \ n - 1 \ n - 2 \ \cdots \ n - r + 1$$

- 解決方法2: 対数を使用する。

- $a/b \rightarrow \log(A) - \log(b)$

- 解決方法3: 処理系に依りますが、(unsigned) long もしくは (unsigned) long long というのもあります。
- 解決方法4: それ以上の桁数が必要ならば、無限精度を扱えるライブラリ(GNU MP)などがあります。

コメント(その2)

- 課題2を出すなら、授業でint型の最大値を越える数を扱う場合にintmaxを使わなければならないということを教えておいてほしかったです。
 - 初耳でした。調べてみたところ、intmax_tは、処理系によって、long intかlong long intに置き換えられるようです。

そのほか

- LaTeXでソースの貼り付け方を忘れた人がいる(読めません)

問題1 `include<stdio.h> include<math.h> define taiseki(r,a) a*(M_PI)*r*r*r*r*4/3`
`int main(void) float a, r, v; printf("半径を入力してください。 : "); scanf("a = 1; v = taiseki(r, a); pri`

問題2 `include<stdio.h> int kaijo(int x); int main(void) int a,b,c,d,e,n,r;`
`printf("自然数を入力してください。 "); scanf("printf("自然数を入力してください。さっきより小さい数で。"); scanf("a=kaijo(n); b=kaijo(r); c=kaijo(n-r); d=a/c; e=a/(b*c); printf("printf("printf("return 0;`

`int kaijo(int x) int i,a; a=1; for(i=1;ij=x;i++) a=a*i; return a;`