

復習用課題 (rev07.c)

倍々に増えるお小遣い

1日目…1円、2日目…2円、3日目…4円，，，と毎日，前日の倍の金額を15日間もらえるとする。各日にもらった金額とその日までの累計金額を下記のフォーマットどおりに画面に表示せよ。
#等比級数の公式は使わないこと。

実行画面(例)

```
emerald.fml.mech.tuat.ac.jp> cc rev07.c
emerald.fml.mech.tuat.ac.jp> a.out
1日目 もらった金額 1円, 合計金額 1円
2日目 もらった金額 2円, 合計金額 3円
3日目 もらった金額 4円, 合計金額 7円
4日目 もらった金額 8円, 合計金額 15円
5日目 もらった金額 16円, 合計金額 31円
emerald.fml.mech.tuat.ac.jp>
```

[HINT] 整数の出力桁数をn桁に合わせる… %nd
(%とdの間にそろえたい桁数を書く(例%6d))

Step 10 関数

課題1 プログラムの入力と実行

```
/* sample10-1.c */  
#include <stdio.h>
```

```
int sum(int a,int b)  
{  
    int ans;  
    ans=a+b;  
    return ans;  
}
```

```
main()  
{  
    int x,y,z;  
    x=1;  
    y=2;  
    z=sum(x,y);  
    printf("%d\n", z)  
}
```

- (1) ひとつのプログラムの中で、繰り返し現れる手続きをひとまとめにする
- (2) 良く使う手続きを関数にしておくと、別のプログラムにも流用できる
Step 11 ヘッダーファイル参照

関数の「戻り値(整数)
(引数、ここでは2つの整数)

戻り値の代入

引数は1対1で対応させる必要がある。
関数の中で定義された変数は、
その関数の中のみで有効
(sumのなかではa, b; mainのなかではx, y)

関数に2つの整数を与え、戻り値をzに代入

Step 10 関数 (その2)

<参考> sample10-2.c

関数の中で別の関数(だけでなく自分自身も)を呼び出しありOK

課題2 (ex10-3.c)

(p. 45, 演習(3))

x (整数)の n (整数)乗を求める関数を含むプログラムをつくり,
2の15乗を計算しなさい

```
/* ex10-3.c 解答例 */
#include <stdio.h>

int power(int x, int n)
{
    int i,ans;
    ans=1;
    for(i=0;i<n;i++)
    {
        ans=ans*x;
    }
    return ans;
}

main()
{
    int x,n,a;
    printf("input x => ");
    scanf("%d", &x);
    printf("input n => ");
    scanf("%d", &n);
    a=power(x,n);
    printf("%dの%d乗は%d\n", x,n,a);
}
```

Step 11 ヘッダー

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

```
#include "my.h"
```

```
#define DOZEN 12
```

```
main( )  
{  
...  
}
```

課題4 p.48 sample11-3.c
を作成し, 実行せよ

標準関数ライブラリー用
ヘッダーファイルの中にも
予め定義されている定数がある

標準関数ライブラリー
(**stdio.h**)を読み込む
自作ヘッダーファイルを読み込む

プログラム全体で使う定数名
(**DOZEN**)とその中身(**12**)を定義

課題3 p. 46 sample11-1.c

sample10-1.cで作った2つの整数の和を求める関数の部分だけ
を, 別ファイル **sum.h** に保存し, 以下のプログラムを実行せよ

```
/* sample11-1.c */  
#include <stdio.h>  
#include "sum.h"  
main()  
{  
    int x,y,z;  
    x=1;  
    y=2;  
    z=sum(x,y);  
    printf("%d\n", z)  
}
```

Step 12 ライブラリー

stdio.h (標準入出力ライブラリー) で定義されているおもな関数

printf scanf fprintf fscanf fopen fclose

puts fputs gets fgets putchar getchar fputc putc fgetc getc ungetc ...

math.h (数学ライブラリー) で定義されている関数

exp ldexp frexp

modf fmod

log log10

pow

sqrt

sin asin sinh

cos acos cosh

tan atan atan2 tanh

ceil floor fabs

このほか、

stdlib.h, time.h, string.h, ctype.h

にさまざまな関数が定義されている。

それぞれのヘッダーファイルで定義されている
すべての関数の意味、引数、戻り値の型については、
http://cham.ne.jp/piro/p_stdfunc.html

などを参照

重要 math.hを使うときは、リンクオプションが必要。具体的には、
コンパイルするときに「`gcc *.c -lm`」と書く（はいふん・える・えむ）

Step 12/13 数学ライブラリーの利用

課題5 (ex12.c) 三角関数表を作れ

右の図にならって、
角度θ, ラジアン, \sin , \cos , $\sin^2 \theta + \cos^2 \theta$
の5項目を1行にして、
 $0^\circ < \theta < 90^\circ$ の間を3°おきに表示

[HINT] %3d, %7.4f を使う

```
emerald.fml.mech.tuat.ac.jp> a.out
DEG RAD SIN COS SIN2+COS2
 0 0.0000 0.0000 1.0000 1.0000
 3 0.0524 0.0523 0.9986 1.0000
 6 0.1047 0.1045 0.9945 1.0000
 9 0.1571 0.1564 0.9877 1.0000
12 0.2094 0.2079 0.9781 1.0000
15 0.2618 0.2588 0.9659 1.0000
18 0.3142 0.3090 0.9511 1.0000
21 0.3665 0.3584 0.9336 1.0000
24 0.4189 0.4067 0.9135 1.0000
27 0.4712 0.4540 0.8910 1.0000
30 0.5236 0.5000 0.8660 1.0000
33 0.5760 0.5446 0.8387 1.0000
36 0.6283 0.5878 0.8090 1.0000
39 0.6807 0.6293 0.7771 1.0000
42 0.7330 0.6691 0.7431 1.0000
45 0.7854 0.7071 0.7071 1.0000
48 0.8378 0.7431 0.6691 1.0000
51 0.8901 0.7771 0.6293 1.0000
54 0.9425 0.8090 0.5878 1.0000
57 0.9948 0.8387 0.5446 1.0000
60 1.0472 0.8660 0.5000 1.0000
63 1.0996 0.8910 0.4540 1.0000
66 1.1519 0.9135 0.4067 1.0000
69 1.2043 0.9336 0.3584 1.0000
72 1.2566 0.9511 0.3090 1.0000
75 1.3090 0.9659 0.2588 1.0000
78 1.3614 0.9781 0.2079 1.0000
81 1.4137 0.9877 0.1564 1.0000
84 1.4661 0.9945 0.1045 1.0000
87 1.5184 0.9986 0.0523 1.0000
90 1.5708 1.0000 -0.0000 1.0000
emerald.fml.mech.tuat.ac.jp>
```

Step 13/14 数学ライブラリーの利用 / 例外処理

sample14.c

19行目 `return -1;` main関数の戻り値を-1にする(必須ではない)

20行目 `exit(0);` プログラムを途中で終わらせる命令

余裕がある人への

課題6 (ex14-2.c)

二次方程式の解を求めるプログラムを作れ

sample14.c をもとに、

あたられた $ax^2 + bx + c = 0$ について、

- (1) 方程式が2つの実数解を持つ場合 (sample14にすでにある)
- (2) 方程式が2つの虚数解を持つ場合 (sample14にすでにある)
- (3) 方程式が重解を持つ場合 (新しく付け加える)
- (4) $a = 0$ について (sample14では全部強制終了)
 - (a) $bx + c = 0$ に答えがある場合 ($b \neq 0$) は
1つの答えを出力してプログラム正常終了
 - (b) $bx + c = 0$ に答えがない場合 ($b = 0$) は強制終了

にすべて対応するプログラムを完成させること