

■ 入力、scanf、if、switch.

printf () などによるプログラムからの結果の取り出し (人間に見える形にする) を出力と呼んでいます。これに対してプログラムに値や指示を与えることを入力と呼びます。ここではそうした方法のうちの一つ、キーボードから文字入力を行う例を示します。

□ scanf による文字入力

右のプログラムを入力して実行してください。(グラフィクスプログラムではないので、コンパイルは egg コマンドではなく cc コマンドが良い。)

実行するとキーボードからの入力を待つ状態になります。以下のように 230 と入力すると、消費税を計算して結果を画面に出力します。

```
% ./a.out
230
total = 241
%
```

```
#include <stdio.h>

/*
 消費税計算をする 473088 榎田裕一郎
*/

main() {
  int price, total;

  scanf("%d", &price);
  total=price * 1.05;
  printf("total = %d\n", total);

  return 0;
}
```

scanf("%d", &price); で price 変数に入力された値 (この例では 230) が入ります。

書式 :

```
scanf("書式", &変数 1, &変数 2, ... &変数 n);
```

"書式"には入力する変数の数だけ % に続けて入力する形式を %d や %f の変換文字列で指定します。変換文字列は printf のそれと同じで、%d は十進数の整数、%f は float 型の実数に、%lf が double 型の実数に対応します。変数の前の & については今は説明しません(*1)。

つまり二つの変数に値を設定したい場合は、以下のように書きます。

```
scanf("%d %f", &price, &tax);
```

二行に分けて書く方法もあります。

```
scanf("%d", &price);
scanf("%f", &tax);
```

どちらの方法でも、実行時の入力の方法は同じです。二行に分けて入力することもできます。

```
% ./a.out
230 1.05
total = 241
%
```

```
% ./a.out
230
1.05
total = 241
%
```

□ 課題 1.

上記の例のように税率も scanf で入力するようにして実行し、動作を確認してください。

*1 &記号、つまりポインタについては「プログラミング B」で説明します。

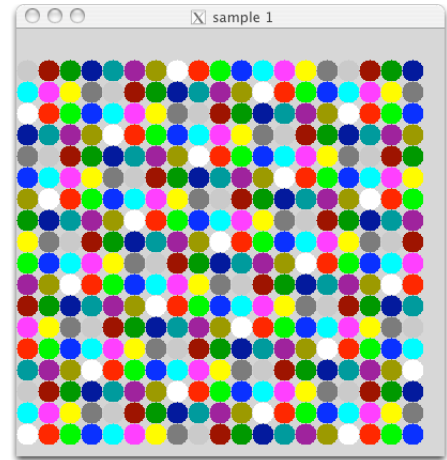
□ 課題 2.

入力に応じて処理を変える

先週の小さな四角を描くプログラムに手を加えて、入力に対応して四角ではなく丸を描くように機能追加してください。

実行すると以下のように 1 か 2 のどちらかを入力するように求め、1 を入力すれば四角、2 を入力すれば円を描くようにしてください。(右図は 2 を指定した場合)

```
% ./a.out
drawing type? (1-2) = 2
%
```



以下の点に注目してプログラムを修正して下さい。

押さえて欲しいところ：

- printf() でプロンプトとなる文字列を出力。(改行なし)
- scanf() で描画図形のタイプを入力し、int 型変数 type に格納。
- type の値によって描く図形のタイプを if 文で変更。
- fillrect() で四角形 (塗りつぶし) を描く。
- fillarc() で円 (塗りつぶし) を描く。

修正の例

```
printf("drawing type? (1-2) = ");
scanf("%d",&type);
.. (中略) ..
if(type == 1) {
    fillrect(win, x, y, 10.0, 10.0); /* 四角を描く */
} else {
    fillarc(win, x, y, 10.0, 10.0, 0.0, 360.0, 1); /* 円を描く */
}
```

fillarc() 関数の引数の役割は以下の通りです。(詳細な仕様は <<ガイド 9.3>> 参照)

```
fillarc(win, x, y, w, h, s, e, d);
```

x,y: 描く円の中心の座標位置 (x,y)

w: 描く円の水平方向の径

h: 描く円の垂直方向の径

s,e: 描きはじめの角度と、描き終わりの角度。0.0 と 360.0 で円に、それ以外で扇形になる。

d: 描く方向 (右回りが 1 で左回りが -1)

d は整数型、それ以外は全て実数、float 型です。

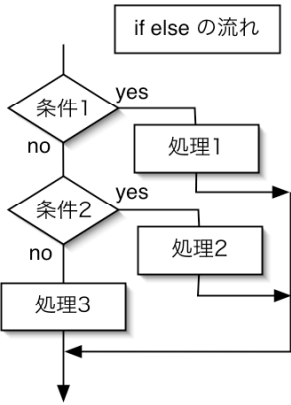
□ さまざまな if 文の書き方

1, 2 以外の値を入力した場合の例外処置を下記のように含めました。1 でも 2 でもない場合は描画ウィンドウを閉じてプログラムを途中で終了します。else if 句の使い方に注目して下さい。

```
if(type == 1) {
    fillrect(win, x, y, 10.0, 10.0); /* 四角を描く */
} else if(type == 2) {
    fillarc(win, x, y, 10.0, 10.0, 0.0, 360.0, 1); /* 円を描く */
} else {
    printf(" 1 か 2 で指定してください。 \n");
    gclose(win);
    return 0;
}
```

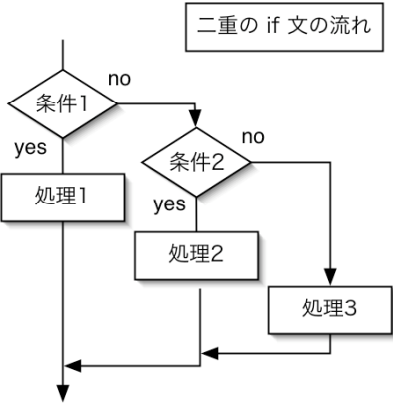
(下の例では一つですが、else if は幾つでも続けて書けます。)

```
if(条件 1) {
    処理 1;
} else if(条件 2) {
    処理 2;
} else {
    処理 3;
}
```



同じ意味の処理を if 文を入れ子にして表現することもできます。入れ子も何重にでもできます。

```
if(条件 1) {
    処理 1;
} else {
    if(条件 2) {
        処理 2;
    } else {
        処理 3;
    }
}
```

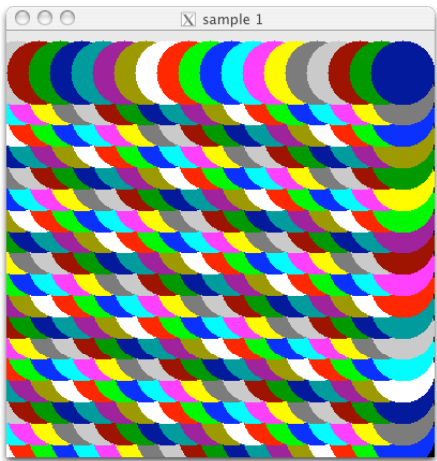


□ 課題 3.

- 上記の例外処置を組み込み、
- type だけでなく、長さも入力し、
- 四角形の辺の長さ、円の径をその長さで描画する。

つまり以下のような実行時の入力によって右図のような結果となるように。(プロンプトに続いて 2 30.0 と入力した。)

```
% ./a.out
drawing type (1-2) and size = 2 30.0
%
```



□ switch 文による場合分け

if else を使った条件分岐を、switch 文を使ってより分かりやすく記述できる場合があります。

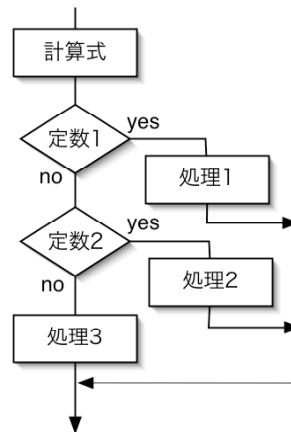
```
switch( type ) {
case 1:
    fillrect(win, x, y, w, w); /* 四角を描く */
    break;
case 2:
    fillarc(win, x, y, w, w, 0.0, 360.0, 1); /* 円を描く */
    break;
default:
    printf(" 1 か 2 で指定してください。 \n");
    gclose(win);
    return 0;
}
```

switch() のカッコ内に変数や計算式を書き、その結果がどのケースに該当するかによって処理を分岐させます。

書式 :

```
switch( 式 ) {
case 定数 1:
    処理 1;
    break;
case 定数 2:
    処理 2;
    break;
.. (略) ..
default:
    処理 n;
}
```

switch の流れ



注意点 :

- case に続く条件部分には定数または定数式しか書けません。($x < 10$ など条件式は不可。)
- case による分岐は幾つでも書けます。
- どのケースにも該当しなかった場合は default: に続く処理を実行します。
- default は省略できます。(その場合該当しないケースでは何もせずに switch を通過します。)
- 各処理の最後には break; を付けてください(*1)。break はループを脱出するものとして登場していましたが、switch からの脱出にも使います。
- switch の直後の { と、対応する } を忘れないように。

□ 課題 4.

サンプルプログラムを加工し、switch を使って描く図形を 4 種類指定できるようにしてください。あと二つの図形には (中を塗りつぶさない) 四角を描く drawrect()、(中を塗りつぶさない) 円を描く circle() を利用すると良いでしょう。各関数の仕様については<<ガイド 9.3>> を参照。

*1 たとえば一番目の処理の最後に break; が存在しなかった場合、処理 1 が終わると無条件に処理 2 を実行します。なぜこのような言語仕様になっているのかはここでは説明しません。興味のある人はコンパイルされたコードを読むと分かるでしょう。