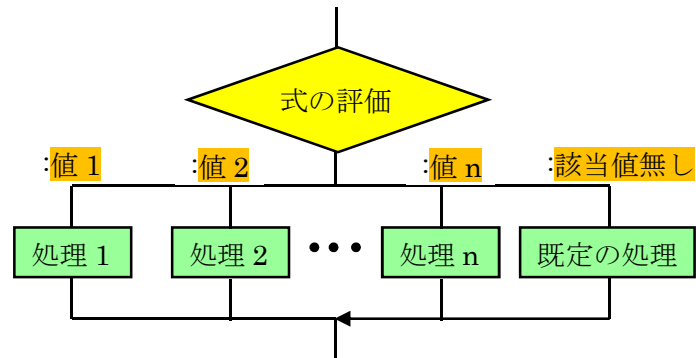


2013 年 5 月 23 日 (木) 実施

多分岐のプログラム

前回は多段階の 2 分岐を組み合わせ
て 3 種類以上の場合分けを実現したが、
式の値の評価によって、一度に多種類
の場合分けを行う多分岐の利用によ
って見通しのよいプログラムを作成
できる場合がある。(流れ図は右図)



switch 文

C 言語で多分岐のプログラムを実現するための文として、**switch 文**が用意されている。switch 文の構文は次のようになる。

```

switch ( 制御式 )
{
    case 事例式 1: [文 1-1] [文 1-2] ... [break;]
    case 事例式 2: [文 2-1] [文 2-2] ... [break;]
    ...
    case 事例式 n: [文 n-1] [文 n-2] ... [break;]
    [default: [既定文 1] [既定文 2] ...]]
}
    
```

ここで、[]内は省略可能である。switch 文の機能は、制御式を評価してその値が事例式のいずれかの値に一致したとき、その case ラベルに続く文を break 文に出会うまで実行する。break 文に到達すると switch 文から抜け出す。もし、一致した case ラベルに続く文で break 文が省略されている場合には、その下の case ラベルに続く文も break 文に出会うまで実行する。

また、switch 文では、制御式を評価してその値が事例式のいずれの値にも一致しないときは、default ラベルが書かれていれば、それに続く文を実行する。(default は要求されているものが抜け落ちているという意味)

例題 1 (switch 文を用いた多分岐 -int 型変数の評価-)

次のソースプログラムをテキストエディタで入力して、prog6-1.c の名前を付けて保存する。それを翻訳・編集して実行形式のファイルを作成し、5 回実行して、それぞれの実行の際に 60 点未満、60 点台、70 点台、80 点台、90 点以上の点数を入力せよ。

```

/* prog6-1.c */
#include <stdio.h>
#include <stdlib.h>
    
```

```
int main(void)
{
    int tensuu, rank;

    printf("点数を 0 点から 100 点の範囲の整数で入力してください: ");
    scanf("%d", &tensuu);

    if (tensuu < 0 || tensuu > 100)
    {
        printf("入力された点数は範囲外です。¥n");
        exit(1);
    }

    rank = tensuu/10;
    switch(rank)
    {
        case 6: printf("点数が%d点なので, 成績評価は[可]です。¥n", tensuu); break;
        case 7: printf("点数が%d点なので, 成績評価は[良]です。¥n", tensuu); break;
        case 8: printf("点数が%d点なので, 成績評価は[優]です。¥n", tensuu); break;
        case 9:
        case 10: printf("点数が%d点なので, 成績評価は[秀]です。¥n", tensuu); break;
        default: printf("点数が%d点なので, 成績評価は[不可]です。¥n", tensuu);
    }

    return 0;
}
```

【解説】

1. rank が 9 の場合の case ラベルに続く文はなく, break 文が省略された場合に相当する。したがって, rank が 9 及び 10 の場合に同一の文が実行される。
2. default ラベルに続く文は rank が 6, 7, 8, 9, 10 の何れでもない場合に実行される。

例題 2 (switch 文を用いた多分岐 —char 型配列要素の評価—)

次のソースプログラムをテキストエディタで入力して, prog6-2.c の名前を付けて保存する。それを翻訳・編集して実行形式のファイルを作成し, 次の様に 6 回, 実行せよ。

- 1) prog6-2 +
- 2) prog6-2 -
- 3) prog6-2 *
- 4) prog6-2 /
- 5) prog6-2 %
- 6) prog6-2 a

```
/* prog6-2.c */
#include <stdio.h>
#include <stdlib.h>

int nyuryokul(void);
int nyuryoku2(void);
void wa(int, int);
void sa(int, int);
void seki(int, int);
void shou(int, int);
void amari(int, int);

int main(int argc, char *argv[])
{
    int x, y;

    if (argc != 2)
    {
        printf("利用法 : ¥n prog6-2 +¥n prog6-2 -¥n prog6-2 *¥n prog6-2 /¥n");
        printf(" prog6-2 %%¥n");
        exit(1);
    }
    else if (argv[1][1] != '¥0')
    {
        printf("演算子は 1 文字のみです。¥n");
        exit(1);
    }
    else if (argv[1][0] != '+' && argv[1][0] != '-' && argv[1][0] != '*'
        && argv[1][0] != '/' && argv[1][0] != '%')
    {
        printf("演算子は, +, -, *, /, %%のいずれかです。¥n");
        exit(1);
    }

    switch(argv[1][0])
    {
        case '+': x = nyuryokul(); y = nyuryoku2(); wa(x, y); break;
        case '-': x = nyuryokul(); y = nyuryoku2(); sa(x, y); break;
        case '*': x = nyuryokul(); y = nyuryoku2(); seki(x, y); break;
        case '/': x = nyuryokul(); y = nyuryoku2(); shou(x, y); break;
        case '%': x = nyuryokul(); y = nyuryoku2(); amari(x, y); break;
    }

    return 0;
}

int nyuryokul(void)
{
    int a;
    printf("1 つ目の整数を入力してください : ");
    scanf("%d", &a);

    return a;
}
```

```
int nyuryoku2(void)
{
    int a;
    printf("2つ目の整数を入力してください: ");
    scanf("%d", &a);

    return a;
}

void wa(int a, int b)
{
    printf("%d + %d => %d\n", a, b, a+b);
}

void sa(int a, int b)
{
    printf("%d - %d => %d\n", a, b, a-b);
}

void seki(int a, int b)
{
    printf("%d * %d => %d\n", a, b, a*b);
}

void shou(int a, int b)
{
    printf("%d / %d => %d\n", a, b, a/b);
}

void amari(int a, int b)
{
    printf("%d %% %d => %d\n", a, b, a%b);
}
```

【解説】

C 言語では 1 行に複数の文を並べて記述することが出来る。

【注意】 Windows 上でも gcc で翻訳・編集した場合、前回の prog5-2.c の注意と同様、プログラム実行時に、prog6-2 * とした場合、'*' が任意の文字列を表すワイルドカードとして扱われるため、積を呼び出すための第 1 プログラム引数を 'x' とする等の工夫が必要である。

演習 1

第 1 プログラム引数として、A または a を入力すると「あ」、I または i を入力すると「い」、U または u を入力すると「う」、E または e を入力すると「え」、O または o を入力すると「お」、それ以外の文字を入力すると「変換ルールがありません。」と画面に表示するプログラムの空欄 1) ____, 2) ____ を埋めてソースプログラムを完成させ、テキストエディタで入力して、ex6-1.c の名前を付けて保存する。それを翻訳・編集して実行形式のファイルを作成し、実行せよ。

```
/* ex6-1.c */
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    if (argc != 2)
    {
        printf("利用法 : %n ex6-1 A%n ex6-1 a%n ex6-1 I%n ex6-1 i%n");
        printf(" ex6-1 U%n ex6-1 u%n ex6-1 E%n ex6-1 e%n");
        printf(" ex6-1 O%n ex6-1 o%n");
        exit(1);
    }
    else if (argv[1][1] != '0')
    {
        printf("引数は 1 文字のみです。 %n");
        exit(1);
    }

    switch( 1 )
    {
        case 'A':
        case 'a': printf("あ%n"); break;
        case 'I':
        case 'i': printf("い%n"); break;
        case 'U':
        case 'u': printf("う%n"); break;
        case 'E':
        case 'e': printf("え%n"); break;
        case 'O':
        case 'o': printf("お%n"); break;
        2 : printf("変換ルールがありません。 %n");
    }

    return 0;
}
```

演習 2 (余裕のある人向け)

第 1 プログラム引数として、7 桁の学籍番号を入力するとその 3 文字目を読み取り、学科名を画面に表示するプログラムの空欄 1) _____ ~ 8) _____ (同一番号には同一内容が入る)を埋めてソースプログラムを完成させ、テキストエディタで入力して、ex6-2.c の名前を付けて保存する。それを翻訳・編集して実行形式のファイルを作成し、実行せよ。(提出するファイルは **ex6-2.c の完成版**とする)

ただし、学科名は次のルールで決められているものとする。また、strlen は文字列の長さを図る関数である。

3 文字目が 1 : 商経学部商学科
3 文字目が 2 : 商経学部経済学科
3 文字目が 3 : 商経学部経営学科
3 文字目が 4 : 政策情報学部政策情報学科
3 文字目が 5, 6, 7, 8, 9 : 大学院
3 文字目が a : サービス創造学部サービス創造学科

```
/* ex6-2.c */
#include <stdio.h>
#include <string.h> /* strlen を用いるため */
#include <stdlib.h> /* exit を用いるため */

int main(int argc, char *argv[])
{
    if (argc != 2)
    {
        printf("利用法 : %n ex6-2 xxxxxxx(7桁の学籍番号)%n");
        exit(1);
    }
    else if (strlen(argv[1]) != 7)
    {
        printf("学籍番号は 7 文字です。 %n");
        exit(1);
    }

    switch(argv[1][2])
    {
        case '1': printf("1) _____ %n"); 2) _____;
        case '2': printf("3) _____ %n"); 2) _____;
        case '3': printf("4) _____ %n"); 2) _____;
        case '4': printf("5) _____ %n"); 2) _____;
        case '5':
        case '6':
        case '7':
        case '8':
        case '9': printf("6) _____ %n"); 2) _____;
        case 'a': printf("7) _____ %n"); 2) _____;
        8) _____: printf("該当する学科がありません。 %n");
    }

    return 0;
}
```

提出物 :

- 1) 例題 1, 2 の出力結果をコピーして貼り付けたテキストファイル **res6.txt**
- 2) 演習 1 のソースプログラムのファイル **ex6-1.c** の**完成版**