プログラミング演習
～条件分岐～

1　目的

C 言語における条件分岐を理解し，その時々の状況に応じて処理の仕方や出力の仕方を変えてみる．例題としてバンディット問題 [1] を用いて実践し理解を深める．

2　製作対象 条件分岐を用いたバンディット操作

条件分岐を用いたバンディット試行プログラムを作成する．今回は，bandit00.o のバンディットを対象とする．

今回の学習目的である条件分岐の表現方法は swich 命令と if 命令の二つがある．

switch 命令による条件分岐は 20 行目から 33 行目までであり，if 命令による条件分岐は 38 行目から 42 行目までである．

switch 命令の代表的な使い方は

- switch(式または変数){
  case 値 1:
  式または変数が，値 1 であれば実行
  break;
  }

である．ここで，プログラムの 30 行目にある default: は，switch の式または変数が case で考えている値のどれとも合致しない場合に，実行される．つまり，case が”値 1 の場合”，”値 2 の場合”であるのに対して，default は”その他”に相当する．

if 命令の代表的な使い方は

- if(条件 1) 条件 1 がなりたつときに実行する命令
- if(条件 1) 条件 1 がなりたつときに実行する命令
  else 条件 1 がなりたたないときに実行する命令
- if(条件 1) 条件 1 がなりたつときに実行する命令
  else if(条件 2) 条件 1 がなりたたず，かつ条件 2 がなりたつときに実行する命令
  else 条件 1 も条件 2 もなりたたないときに実行する命令

である．ここで条件 (条件式ともいう) とは,「二つの値が同じ」や「ある値が他の値より大きい」など正しいか正しくないかがはっきり分かれる式である．正しい，は，条件式が「成り立つ」ともいい，0 以外の数値で表されることもある．正しくない，は，条件式が「成り立たない」ともいい，数値的には 0 で表されることもある．主なものとして以下のようなものがある．

- a == b (a と b が等しければ正しい (1)，等しくなければ正しくない (0))
- a ! = b (a と b が等しくなければ正しい (1)，等しければ正しくない (0))
- a > b (a が b よりも大きければ正しい (1)，a が b 以下であれば正しくない (0))
- a >= b (a が b 以上であれば正しい (1)， a が b よりも小さければ正しくない (0))
- <, <= も同上
- 1 (もしくは 0) 常に条件はなりたっている (なりたっていない)，と言っているのと同じ意味

ちなみに，実行する命令が複数行にわたる場合には，{ と } でくくってやればよい．

2.1 準備

ディレクトリ ”programming03”を作成する．今回の演習では，プログラムの作成や必要ファイルのダウンロードは，programming03 ディレクトリで行う．
はたおり虫より，以下のファイルをダウンロードする．

- bandit.h
- bandit00.o

2.2 プログラム

バンディット関係の関数は，ガイダンス資料 [1] に説明があるので一読すること．

```
1  /*********************************************************
2            人によるバンディットの操作分岐
3  *********************************************************/
4
5  #include <stdio.h>  /* システムの用意した入出力         */
6  #include <stdlib.h> /* システムの用意した便利関数       */
7  #include "bandit.h" /* ユーザが用意したバンデット関連 */
8
9  int main(){
10   int select_arm, num_arm;
11   double reward;
12
13   init_bandit();           /* バンディットの初期化 */
14   num_arm = get_arm_num(); /* バンディットが持つ腕の数を取得 */
```

```
15
16      /* バンディットの腕の選択 */
17      printf("バンデットの腕の番号を選択してください [1-%d]:",num_arm);
18      scanf("%d", &select_arm);
19
20      /* 選択した腕を表示 */
21      switch(select_arm){
22      case 1:
23        printf("腕 1 を選択\n");
24        break;
25      case 2:
26        printf("腕 2 を選択\n");
27        break;
28      case 3:
29        printf("腕 3 を選択\n");
30        break;
31      default:
32        printf("範囲外\n");
33        break;
34      }
35
36      /* バンディット動作 */
37      reward = bandit(select_arm);
38
39      if(reward > 0.0){ /* なにがしかの報酬を得ている = 当たり */
40        printf("当たりです\n");
41      }else{          /* 報酬を得ていない = はずれ */
42        printf("はずれです\n");
43      }
44
45      return 0;
46    }
```

2.3 コンパイル

上記のプログラムを作成し，ファイル名 gameplay.c として保存してコンパイルを行う．コンパイル後の名前を gameplay とする．

> gcc -o gameplay gameplay.c bandit00.o

注意することは，ダウンロードしたファイル bandit.h と bandit00.o を使用するので作成したファイルと同じディレクトリに両ファイルを保存しておくこと．

2.4　動作実験

作成したプログラムを動作させる．腕を選択し，当たり/外れを観察する．

2.5　調べてみよう

if 命令，switch 命令の使い方を教科書等で詳しく調べてみよう．

2.6　やってみよう

2.6.1　バンディットの挙動によって条件分岐

バンディット bandit00.o は腕を 3 本持ってる．この時，腕番号の選択時に 4 を入力したり，bandit(4) などで 4 本目の腕を選択すると，バンディット bandit() は「おかしな腕が選択させているよ」とエラー値として-1.0 を返す．バンディットからの返答はプログラムの 37 行目を見ると分かるように，変数 reward に保存される．そのため，変な腕を選択すると，reward にエラー値-1.0 が保存されている．

現在は，if 命令の中で，reward>0.0 でない場合，つまり reward≤0.0(0.0 以下) の場合は報酬を得ていないものとして,「はずれ」としているが，厳密には，

- reward = 0.0 の場合．これは「はずれ」
- reward = -1.0 の場合．これは「エラー」

と分ける必要がある．上記のように厳密に分けたプログラムをつくってみよう．

2.6.2　switch 命令と if 命令の書き換え

switch 命令と if 命令の書き換えは，基本的にどちらももう一方の命令に書き換えることができる．そこで，プログラム 21 行目から 34 行目の switch を if で，またプログラム 39 行目から 43 行目までの if を switch で書き換えてみよう．

3　参考文献

参考文献

[1] プログラミング演習～ガイダンス～

[2] プログラミング演習～基礎文法～