

# プログラミング演習

## ～基礎文法～

### 1 目的

C 言語における基礎文法の理解を目的とし、バンディット問題 [1] 等を例題にしながら実践し理解を深める。

### 2 製作対象 プログラムの雛形

プログラムの最も基本的な雛形の作成を行う。3 行目はヘッダファイルと呼ばれるものである。stdio.h は standard(標準的な)input/output(入出力)に関する機能を使うためのものである。

7 行目がメインプログラムの開始位置であり,9 行目が終了の位置である。特に 9 行目は正常終了したことを意味する。

#### 2.1 準備

ディレクトリ "programming02" 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、programming02 ディレクトリで行う。

#### 2.2 プログラム

```
1  /*****
2      雛形
3  *****/
4
5  #include <stdio.h>
6
7  int main(){
8
9      return 0;
10 }
```

## 2.3 コンパイル

上記のプログラムを作成し、ファイル名 `pg-sample.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 `pg-sample` とする。

```
> gcc pg-sample.c -o pg-sample
```

## 2.4 動作実験

作成したプログラムを動作させる。今回は雛形であり、何も”仕事”をしないので、なにも起こらない。

```
> pg-sample
>
```

# 3 製作対象 入力-演算-出力

プログラムとは、人間の代わりに計算等の仕事を行ってくれるものである。具体的に言うと、外部(人間や他のプログラムなど)からの入力に従って演算を行い、結果を出力する。この基本的なプログラムを 3.1 に示す。

6 行目は整数を一時的に記憶するための変数である。

10 行目はキーボードからの入力を変数(ここでは `input`)に代入するための関数(かっこ”`()`”)を持つものである。

外部から情報を入力するための関数として、ここでは `scanf` を使用しているが、この他にも様々な関数が存在する。

12 行目が演算である。ここでは掛け算を行い、入力された変数 `input` を 2 倍にして変数 `output` に代入している。このような四則演算 `+*/(÷)` の組み合わせで演算を行う他に、色々な関数を使用することも演算である(その意味では 10 行目の `scanf`, 14 行目の `printf` も入出力に特化した演算を行っているといえる)。

14 行目は外部への出力を行っている。ここでは画面への出力であり、`input x 2 = ...` という文字列と、計算結果である `output` を表示する。

外部へ出力するための関数として、ここでは `printf` を使用しているが、この他にも様々な関数が存在する。

## 3.1 プログラム

```
1  /*****
2      入力-演算-出力
3  *****/
4
5  #include <stdio.h>
6
7  int main(){
8      int input, output;
9
```

```
10     scanf("%d", &input);
11
12     output = input * 2;
13
14     printf("input x 2 = %d\n", output);
15
16     return 0;
17 }
```

## 3.2 コンパイル

上記のプログラムを作成し、ファイル名 `in-pr-out.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 `inprout` とする。

```
> gcc in-pr-out.c -o inprout
```

## 3.3 動作実験

作成したプログラムを動作させる。すると入力待ちになるので整数を入れてリターンキーを打つと結果が返ってくる。

```
> inprout
```

3 (自分で入力)

input x 2 = 6 (プログラムから結果が返ってくる)

## 3.4 調べてみよう

今回使った基本的な入出力関数は `scanf` と `printf` である。これについて教科書等で、使い方を詳しく調べてみよう。

## 3.5 やってみよう

### 3.5.1 心の準備

動作がおかしくなってしまった時の最終手段

- `\C-c` (Ctrl キーを押しながら `c` を押す)
- 先生を呼ぶ

### 3.5.2 様々な入力

作成したプログラム `inprout` に色々な入力をしてみよう

- 1,2,-3,50 といった整数
- 5.5, 8.90, 11.22593 といった実数
- hello といった文字・文字列

### 3.5.3 様々な演算

作成したプログラムを改良して様々な演算を実現してみよう

- 入力した値に 10 を加えるもの
- 入力した値から 3 を引くもの
- 入力した値を 5 でわるもの
- 二つの数値を入力出来るようにして、それぞれ  $+$   $-$   $\times$   $\div$  するもの

### 3.5.4 様々な出力

作成したプログラムを改良してわかりやすい出力を実現してみよう

- 入力 (scanf) する前に, "入力待ちだよ" と分かるようにしてみる
- 表示する数値を答えだけでなく, 入力された数値も表示してみる (input  $x 2 = 6$  から input(3)  $x 2 = 6$  など)

### 3.5.5 扱える変数

作成したプログラムは整数しか扱えない. 小数点を含む実数を扱えるようにしてみよう

## 4 製作対象 バンディット操作

人がバンディットを試行 (プレイする) 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.

バンディットは一番最初に一回だけ初期化という準備をしなければいけない. これが 14 行目の `init_bandit()` である.

バンディットは, いくつか腕を持っており, その腕の一つを選択することで当たった割合に応じて報酬が得られる. バンディットはそれぞれ持っている腕の数が違うが, 15 行目の `get_arm_num()` 関数でその数を知ることができる. 今回の bandit00.o のバンディットは, 3 本の腕を持っているので, 変数 `arm_num` に 3 が保存され, 腕番号 1,2,3 のどれかを選ぶことになる. 各腕を選んだ結果は当たりか外れかのどちらかだけであり, 当たれば報酬として 1 がもらえ, 外れれば報酬が得られない (0 の報酬が与えられる). その, 腕の選択と, 選択された腕でに応じた報酬の計算を行っている部分が, 22 行目である.

```
reward = bandit(select_arm);
```

このうち, 選択した腕の番号 (1,2,3) がカッコの中の `select_arm` である. また, その腕を選んだ結果 (当たり (1)/はずれ (0)) が `reward` に代入される.

その得られた報酬をスコアとして, 22 行目で表示している. ここで, 報酬は実数 (double 型) で表される. そのため `printf` での整数表示 "%d" から実数表示 "%lf" になっていることに注意する.

## 4.1 準備

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

## 4.2 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
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ィット関連 */
8
9  int main(){
10     int select_arm;
11     int arm_num;
12     double reward;
13
14     init_bandit(); /* バンディットの初期化 */
15     arm_num = get_arm_num(); /* バンディットが持つ腕の数を取得 */
16
17     /* バンディットの腕の選択 */
18     printf("バンディットの腕の番号を選択してください [1-%d]:", arm_num);
19     scanf("%d", &select_arm);
20
21     /* バンディット動作 */
22     reward = bandit(select_arm);
23
24     printf("結果は… %lf のスコアです\n", reward);
25
26     return 0;
27 }
```

## 4.3 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。

## 4.4 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。

## 4.5 やってみよう

### 4.5.1 腕の番号入力大丈夫?

`bandit00.o` のバンディットでは、腕は3本である。そのため、選択できる腕は1,2,3であるが、これ以外を選択したらどうなるか?

## 5 参考文献

### 参考文献

[1] プログラミング演習～ガイダンス～