

第2章

「競技プログラミング入門」 by 井上誠大 (73)

2.1 まえがき&競技プログラミングとは (小冊子の内容)

こんにちは、73 回生の井上誠大です。73 回生なので中学 3 年生ですね…(自覚が…) 去年は旧中 1 らしくゲーム制作をしていたのですが、かなり苦労しました>< 去年はプログラミング初心者の登竜門 (?), TE*RIS を作っていたのです。が、判定はガバガバ、描画もかなり簡素なものになってしまい、いかにゲーム作りが大変か思い知らされました^^; 昔の話はこのくらいにして、今年は部誌を書くことになりました!わーい! NPCA の部誌を書くのは初めてどころか、入っていた他の部でも部誌はまだ書いたことがないので、不慣れなところ、日本語が多少不自由なところもあると思いますがよろしくをお願いします。

僕は、今回の部誌で「競技プログラミング」について書きます。ひとまず「競技プログラミング」って何ぞや…と思われる方が相当数だと思うので、ここではそれを簡単に説明しようと思います。

競技プログラミングは、簡単に言うとプログラミングを競技化したものです。と言ってもお前それ語順入れ替えただけやろ、と怒られそうなのでもうちょっと具体的に言います。短時間(だいたい 1.5-8 時間) で何問か出題されるプログラミングやアルゴリズムに関する問題をプログラムを組んで解く、というのが競技プログラミングです。

プログラミングだのアルゴリズムなど難しそうな語句が並んでいますが、実際はプログラムを組む最低限の知識と環境、多少の算数力があれば充分楽しめます。実際、僕の場合、ゲームや Web 系のものを開発するための知識をつけるのには挫折しましたが、競技プログラミングはすぐに慣れました。

具体的にどういった問題が出題されるのか、そのバリエーションは様々で、例えばある値を計算する、あることが可能か、条件を満たすものは何通りか、条件を満たす確率はどのくらいか、最短の距離はどのくらいか、中にはゲームの必勝法を考察するものまであります。全体的に算数や数学らしい内容です。

実際の競技は意外と短時間で、いかに速く解法を閃き、正確かつ迅速に正解を導き出すかも重要な要素になります。しかし、あまり速く解きすぎても誤答なら解答時間、つまり速度に多少のペナルティが科せられたりもします。

競技は主にインターネット上でオンラインで開催されるので、世界中のライバルと競うことも可

能です。問題文が英語…なんてこともあったりしますが今回扱う競技プログラミングのサイトは大半が日本語なので心配不要です。

今回の部誌では、

- [Aizu Online Judge](#)
- [AtCoder](#)

この2つの競技プログラミングの Web サイトのうち初心者でも充分解くことができるような簡単な問題について解説しながら、競技プログラミングの基本中の基本を書きました。僕は C 言語使いなのですが、その言語の基本も併せて多少は解説を挟むので、プログラミングに関する知識を僅かしか持っていない、そんな方でも内容を理解できるように努力したつもりです><

早速実際に C 言語を利用して競技プログラミングの基本中の基本になる問題を解いていこうと思います。

ここから扱う問題が掲載されている、主に Aizu Online Judge のアカウントは登録・設定から情報を入力すれば新規アカウントが作成できます。ここからはアカウントを持っている前提で書いてます>< それと、この解説がわかりづらかった場合やここに解説していない問題を解きたい、という場合は Introduction 内の多数の問題には公式で解説が付いているのでそちらを参照してください。

2.2 Hello World

http://judge.u-aizu.ac.jp/onlinejudge/description.jsp?id=ITP1_1_A

この問題、要は"Hello World"を出力する問題です。まず、解答例を書きます。(次の章以降も先に正解を書いてからそれを解説します)

```
1 #include<stdio.h>
2 int main(void){
3     printf("Hello World\n");
4     return 0;
5 }
```

1行目

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

これはおまじないだと思ってください。(これを詳しく解説するまでこの部誌では到達しません><)

2行目と4,5行目

```
int main(void){
...
    return 0;
}
```

C 言語ではこの{}の中の...の部分が実行される、という認識で構いません。

そして3行目

```
printf("Hello World\n");
```

この命令は"で囲まれた文章を表示するものです。今回のプログラムだと、Hello World と出力しろ、という命令になります。最後の\nは改行を表します。この後ろ向きのスラッシュ、バックス

ラッシュは円¥マークと同じ方法で入力できます。(競技プログラミングでは大抵出力の最後に改行が求められたりするので、これを忘れないように癖づけることを推奨します。)

今回はプログラムを試しに実行するのに Ideone を利用します。

<http://ideone.com/>

まず、上の大きい四角にプログラムを貼り付け、下の押すといろいろ出て来る中から popular の中にある C を選び、最後に Run ボタンを押します。(アカウントは不要ですが、あるとプログラムが保存できて便利です)

The screenshot shows the Ideone website interface. At the top, there is a text area for source code with the following C code:

```

<> enter your source code or insert template or sample or your template
1 #include<stdio.h>
2
3 int main(void){
4     printf("Hello World\n");
5     return 0;
6 }
7

```

Below the code area is an input field labeled "enter input (stdin)". A dropdown menu is open, displaying a list of programming languages. The menu is divided into two sections: "popular" and "others".

popular		others		Scheme (guile 2.0.13)	
Bash	Pascal	Ada95	Common Lisp	Java	Prolog
C	Perl	Assembler 32t	Common Lisp	JavaScript	Python
C#	PHP	Assembler 32t: D		JavaScript	Python 3 nbc
C++	Python	Assembler 64t: D		Kotlin	R
C++14	Python 3	AWK	D	Lua	Racket
Haskell	Ruby	AWK	Dart	Nemerle	Rust
Java	SQLite	BC	Elixir	Nice	Scala
Objective-C	Swift	Branf**k	Erlang	Nim	Scheme
Pascal	VB.net	C	F#	Node.js	Scheme
		C++ 4.3.2	Fantom	Objective-C	Scheme
		C++14	Forth	Ocaml	Smalltalk
		C99	Fortran	Octave	TCL
		Clips	Go	Oz	Text
		Clojure	Gosu	Perl	Unlambda
		Cobol	Groovy	Pico Lisp	VB.NET
		COBOL 85	Icon	Pike	Whitespace
		CoffeeScript	Intercal	Prolog	

At the bottom of the interface, there is a toolbar with buttons for "C", "stdin", "Run", and "fewer options". The "Run" button is highlighted in green.

すると、こんな感じの画面が出てきます。stdout と書かれた場所にあるのが出力で、問題無く出力されているのが確認できました。



The screenshot shows a code editor interface with the following elements:

- Top navigation: edit, fork, download, template, ?
- Code editor: A C program with 7 lines of code:

```
1. #include <stdio.h>
2.
3. int main(void) {
4.     printf("Hello world\n");
5.     return 0;
6. }
7.
```
- Success message: A green bar with the text "Success" and a "comments (0)" link.
- Input/Output section: A table with two rows:
 - stdin: Standard input is empty
 - stdout: Hello world

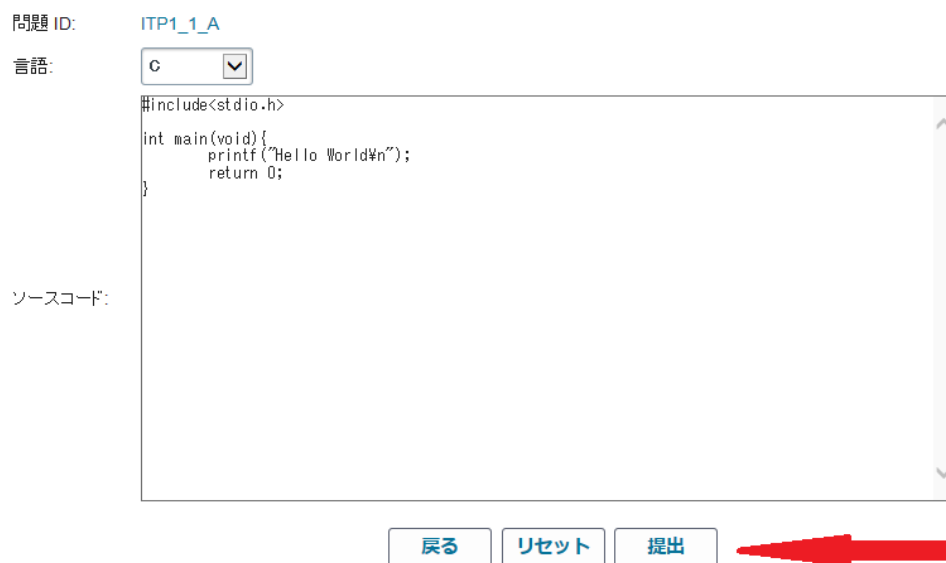
さて、解答が書き上がりましたが、書き上げただけでは勿論何も起こりません。解答を提出しなければなりません。

まず、この丸をつけた所をクリックします。



The screenshot shows the submission page for the 'Getting Started - Hello World' problem. It includes the problem title, a time limit of 1 sec, a memory limit of 65536 KB, and a red circle highlighting the submission icon (a document with an arrow) in the top right corner.

すると、このような画面が出てきます。今回は、言語に C を選択して、ソースコードの所に書いたプログラムを貼り付け、最後に矢印で指したところから提出します。



The screenshot shows the submission interface for the 'Hello World' problem. It includes the following elements:

- 問題 ID: ITP1_1_A
- 言語: A dropdown menu with 'C' selected.
- ソースコード: A text area containing the C program code:

```
#include<stdio.h>
int main(void){
    printf("Hello World\n");
    return 0;
}
```
- Buttons: Three buttons at the bottom: '戻る' (Back), 'リセット' (Reset), and '提出' (Submit). A red arrow points to the '提出' button.

そこから現れる画面で自分の提出の所にこのようなチェックが入れば正解です。おめでとうございます!

結果

✓ : Accepted

2.3 X Cubic

http://judge.u-aizu.ac.jp/onlinejudge/description.jsp?id=ITP1_1_B

この問題は、入力から数値を受け取り、その3乗を出力する問題です。

```
1 #include<stdio.h>
2 int main(void){
3     int x;
4     scanf("%d",&x);
5     printf("%d\n",x*x*x);
6     return 0;
7 }
```

1,2,6,7行目は先ほどと変わりません。

3行目

```
int x;
```

これは int 型の変数 x を宣言する、という命令です。はい、(プログラミング未経験であれば)何言ってるか分かりませんよね。

まず、変数、これは数字などを入れておける箱、と思ってください。(次のページの図も参考にしてください) int 型、とは整数、厳密にいうと ± 21 億くらいまでの整数を入れておける箱の種類のことです。つまり、x って名前の整数を入れておく箱を用意するよ〜、という命令になります。ちなみに (とは言ったもののここはとても重要) この x に何らかの数値を入れておきたい場合は、

```
int x = 5;
```

や

```
int x;
x = 5;
```

などとすると x に 5 が入った状態になります。このように、x = y; と書くと変数 x に数値 y または変数 y の値が入った状態になり、これを代入と呼びます。

4行目

```
scanf("%d",&x);
```

() の中身の左側、%d は int 型、つまり整数の箱 1 つを表し、右側、&x は箱 x のある場所を表します。& を忘れないようにしましょう (厳密にはポインタというのですが、今回は簡単のため詳しい説明はしません><) この scanf という命令にこの情報を教えることで、この命令は箱 x に、入力から整数 1 つを受け取れ、というものになります。

5行目

```
printf("%d\n",x*x*x);
```

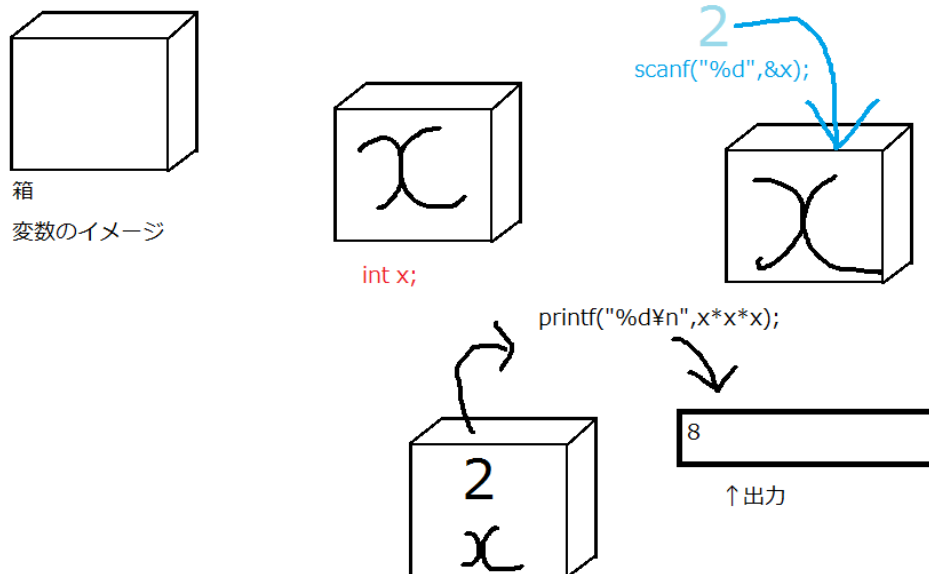
勘の良い方ならお気づきだと思いますが、さっきの Hello World の時に使った命令の進化形です。() の中身の左側、%d は int 型、つまり整数の箱 1 つを表しその後の\n は改行です。右側 x*x*x は x を 3 つ掛けたもの、つまり x の 3 乗です。

ちなみに、

- a+b で a と b の和
- a-b で a から b を引いた差
- a*b で a に b を掛けた積
- a/b で a を b で割った商
- a%b で a を b で割った余り

を表します。計算の順序は $3+2*4$ なら $2*4$ が優先されて 11 になります。

この printf という命令で、箱 x に入っている数値の 3 乗を出力して最後に改行せよ、という意味になります。出力したからと言って x の中身が無くなる訳ではありません。



先ほどの Ideone でプログラムに入力を渡したい場合は、stdin の枠内に書き込みます。今回は "2" と書き込んで実行してみます。

```

edit fork download template ? copy
1. #include<stdio.h>
2.
3. int main(void){
4.     int x;
5.     scanf("%d",&x);
6.     printf("%d\n",x*x*x);
7.     return 0;
8. }

```

Success

comments (0)

```

stdin copy
2
stdout copy
8

```

無事に2の3乗、8が出力されていることが確認できたので先ほどのように提出します。

2.4 Rectangle

http://judge.u-aizu.ac.jp/onlinejudge/description.jsp?id=ITP1_1_C

この問題は、縦 a(cm) 横 b(cm) の長方形の面積と周の長さを求める問題です。面積は $a*b$ 、周の長さは $2*(a+b)$ で求められます。それさえ分かれば殆ど先ほどと同じコードで正解できます。

```

1 #include <stdio.h>
2 int main(void){
3     int a,b;
4     scanf("%d%d",&a,&b);
5     printf("%d %d\n",a*b,2*(a+b));
6     return 0;
7 }

```

3行目

```
int a,b;
```

, で区切って書くと複数の変数を同時に宣言することができます。

4行目

```
scanf("%d%d",&a,&b);
```

入力が空白区切りで与えられるので、このように書くことでデータを2つ受け取ることができます。

5行目

```
printf("%d %d\n",a*b,2*(a+b));
```

2つの %d を空白で区切ると空白区切りの出力ができます。出力させたいものは、区切りで書きます。他にも "[%d]" と書くとと数値を [] で囲んだり "a:%d" と書くと a:12345 のように表示させたりできます。(ここからは実行は省略します><)

復習問題 Watch

ここまでの内容で解ける問題です。

http://judge.u-aizu.ac.jp/onlinejudge/description.jsp?id=ITP1_1_D

是非自力で解いてみてください(ここでは解答のみを挙げますが解説は公式にあります)

```

1 #include <stdio.h>
2 int main(void){
3     int h,m,s;
4     scanf("%d",&s);
5     m = s/60;
6     s = s%60;
7     h = m/60;
8     m = m%60;
9     printf("%d:%d:%d\n",h,m,s);
10    return 0;
11 }
```

2.5 Small, Large, or Equal

http://judge.u-aizu.ac.jp/onlinejudge/description.jsp?id=ITP1_2_A

2数の大小関係を求める問題です。

```

1 #include <stdio.h>
2 int main(void){
3     int a,b;
4     scanf("%d%d",&a,&b);
5     if(a < b){printf("a < b\n");}
6     else if(a > b){printf("a > b\n");}
7     else{printf("a == b\n",a,b);}
8     return 0;
9 }
```

5行目

```
if(a < b){printf("a < b\n");}
```

if(~){...}は、()の中の条件が正しければ{}の中を実行する文です。これをif文と呼びます。

- $a < b$ なら b が a より大きい
- $a \leq b$ なら b が a 以下
- $a == b$ なら a と b が等しい
- $a != b$ なら a と b が等しくない

時に{}の中が実行されます。

ちなみに、

- $a < b \ \&\& \ b < c$ なら $a < b$ かつ $b < c$
- $a < b \ || \ a < c$ なら $a < b$ または $a < c$

というように、&&で「かつ」、||で「または」を表すことができます。

6行目

```
else if(a > b){printf("a > b\n");}
```

else が付いている部分は、先ほどの条件を満たさなかった場合に実行される部分です。今回は $a \geq b$ の場合です。そのあとに先ほどと同じ、条件が $a > b$ になった文が書かれており、それを満たすと `{}` の中が実行されます。

7行目

```
else{printf("a == b\n",a,b);}
```

またしても条件を満たさなかった場合、つまり $a == b$ の場合、この `{}` の中が実行されます。

復習問題 Range

http://judge.u-aizu.ac.jp/onlinejudge/description.jsp?id=ITP1_2_B

$a < b < c$ かどうか判定する問題です。

```
1 #include <stdio.h>
2 int main(void){
3     int a,b,c;
4     scanf("%d%d%d",&a,&b,&c);
5     if(a < b && b < c){printf("Yes\n");}else{printf("No\n");}
6     return 0;
7 }
```

2.6 Print Many Hello World

http://judge.u-aizu.ac.jp/onlinejudge/description.jsp?id=ITP1_3_A

"Hello World"を1000回出力する問題です。

```
1 #include <stdio.h>
2 int main(void){
3     int i;
4     for(i = 1;i <= 1000;i++){
5         printf("Hello World\n");
6     }
7     return 0;
8 }
```

お察しの通り、1章のプログラムを少し書き換えただけで正解できます。

4行目、6行目

```
for(i = 1;i <= 1000;i++){
...
}
```

`for(~){...}`、これを `for` 文と呼びます。今回の場合は、 i を1から1000まで、 i を1ずつ加えながら `{}` の中身を実行という意味になります。

具体的なイメージとしては

```
printf("Hello World\n");//i == 1
printf("Hello World\n");//i == 2
printf("Hello World\n");//i == 3
...
printf("Hello World\n");//i == 1000
```

といった感じです。 $i++$ は i に1加える、 $i--$ なら i から1引くという意味になります。

2.7 Print Test Cases

http://judge.u-aizu.ac.jp/onlinejudge/description.jsp?id=ITP1_3_B

いくつか分からない入力を最後まで受け取って形式を変換する問題です。入力の終端が0で表されます。

```

1 #include <stdio.h>
2 int main(void){
3     int x,i=1;
4     while(scanf("%d",&x) , x != 0){
5         printf("Case %d: %d\n",i,x);
6         i++;
7     }
8     return 0;
9 }
```

4行目、7行目の

```

while(scanf("%d",&x) , x != 0){
    ...
}
```

while(~){...}を while 文と言います。() の中の条件を満たしている間{}の中を実行し続けます。今回は、x が0 でない間入力から x を受け取り続ける、という意味になります。入力が123 45 6 0 なら、

- (x != 0 なので)"Case 1: 123\n"と出力
- (x != 0 なので)"Case 2: 45\n"と出力
- (x != 0 なので)"Case 3: 6\n"と出力
- (x != 0 でなくなったので終了)

というような感じで実行されます。

2.8 Reversing Numbers

http://judge.u-aizu.ac.jp/onlinejudge/description.jsp?id=ITP1_6_A

与えられる数値を逆順に出力します。

```

1 #include <stdio.h>
2 int main(void){
3     int a[128],n,i;
4     scanf("%d",&n);
5     for(i = 1;i <= n;i++){scanf("%d",&a[i]);}
6     for(i = n;i >= 1;i--){if(i != n){printf(" ");}printf("%d",a[i]);}
7     printf("\n");
8     return 0;
9 }
```

4行目

```
int a[128]...
```

また見慣れないものが出てきましたね。これは配列と言って、変数の箱をいくつか並べたものをイメージすると分かりやすいです。今回は [128] なので箱 128 個、0 番から始まって 127 番まで用意されます。

6行目

```
for(i = 1; i <= n; i++){scanf("%d",&a[i]);}
```

で1番からn番の箱に入力から数値を受け取り、

```
for(i = n; i >= 1; i--){if(i != n){printf(" ");}printf("%d",a[i]);}
```

でn番から1番の箱の順に読み込んで出力します。この問題は空白区切りで出力する問題ですが、for文{}内のif文で出力の最初以外に空白を出力して入力が区切られるようにしています。(無理矢理1行に収めたのは許してください><)

2.9 QQ

<http://judge.u-aizu.ac.jp/onlinejudge/description.jsp?id=0000&lang=jp>

九九を出力する問題です。

```
1 #include <stdio.h>
2 int main(void){
3     int i,j;
4     for(i = 1; i <= 9; i++){
5         for(j = 1; j <= 9; j++){
6             printf("%dx%d=%d\n",i,j,i*j);
7         }
8     }
9     return 0;
10 }
```

4~8行目、for文の中にfor文が書かれています。

- 1x1=1 (i = 1, j = 1)
- 1x2=2 (i = 1, j = 2)

...

- 1x9=9 (i = 1, j = 9)
- 2x1=2 (i = 2, j = 1)

...

- 9x9=81 (i = 9, j = 9)

という風に実行されます。

for文の中にfor文、while文の中にwhile文、for文の中にwhile文、while文の中にfor文…というような書き方もできます。

ここまではAizu Online Judgeの問題について解説しました。他にも様々なレベルの問題があり初心者でも十分楽しめるので是非いろいろ解いてみてください!

2.10 AtCoder に挑戦してみよう!

ここからはAtCoderという日本の競技プログラミングのサイトを紹介します。

<https://atcoder.jp/>

毎週土曜日の 21 時に大抵何らかのコンテストが行われます。コンテストには、コンテストのページから「~に参加する」ボタンを押すと参加できます。

せっかくなので、practice コンテストの問題を 1 問解いてみましょう。

http://practice.contest.atcoder.jp/tasks/practice_1

整数 a, b, c と文字列 s (文字の並び) が与えられるので、 $a+b+c$ と s を 1 行に出力する問題です。

```
1 #include <stdio.h>
2 int main(void){
3     int a,b,c;
4     char s[128];
5     scanf("%d%d%d",&a,&b,&c);
6     scanf("%s",s);
7     printf("%d %s\n",a+b+c,s);
8     return 0;
9 }
```

4 行目

```
char s[128];
```

`char` は文字を入れて置ける箱を表し、今回はその配列を用意します。

6 行目

```
scanf("%s",s);
```

`%s` は文字列を表します。文字列を文字の配列 s に受け取ります。今回は `&` は不要です。

具体的には受け取る文字列が `"test"` だとしたら、

- s の 0 番の箱に `'t'`
- s の 1 番の箱に `'e'`
- s の 2 番の箱に `'s'`
- s の 3 番の箱に `'\0'`
- s の 4 番の箱には文字列の終わりを表す記号が入られます。

8 行目

```
printf("%d %s\n",a+b+c,s);
```

文字列も整数の時と同じように出力できます。

2.11 あとがき

今回は競技プログラミングの簡単な、プログラミングの基礎を扱う問題を紹介しましたが、少しは競技プログラミングに興味を持って頂けたでしょうか？今回紹介した問題はまだまだ第一歩に過ぎません。まだまだこの先には沢山の難しい、楽しい問題が待ち構えています。是非挑戦して頂けると競技プログラミングをやっている身としてありがたいです。それでは、また来年の部誌にご期待ください！