

プログラミング I

Report#2

提出日:2006年某日(木)

所属 :工学部情報工学科

学籍番号: 065702G

氏名 : 新垣 智規

◎例題

◎ソースコード(balance.c)

```
/*
Program                :          balance.c
Student-ID             :          065702G
Author                 :          ARAKAKI, Tomonori
UpDate                :          2006/05/12(Fri)
Comments              :          Payment    &    Balance
*/

#include <stdio.h>

int main()
{
int price = 1234, pay = 10000;
int balance, amount;

/***** scanf */
printf("Price?  => "); scanf("%d",&price);
printf("Payment? => "); scanf("%d",&pay);
printf("----\n");

/***** scanf */
balance = pay - price;
printf("price = %d, ",price);
printf("payment = %d, balance = %d\n",pay,balance);
printf("----\n");

/***** 5000-yen */
amount = balance / 5000;
```

```
balance = balance % 5000;
printf("5000-yen note = %d\n",amount);

return(0);
}
```

◎実行結果(/a.out)

```
[nw0602:~] j06002% cc balance.c
[nw0602:~] j06002% ./a.out
Price?  => 1234
Payment? => 10000
-----
price = 1234, payment = 10000, balance = 8766
-----
5000-yen note = 1
```

◎考察

10000 円を払って、1234 円の品物を買ったときの、札や硬貨の数を出すプログラム。やっていることは分かるのだが、どのようにプログラムを組んでいいのかが分からないので、参考に例題をやってみた。

Report#1 の授業で習った `printf` 関数も例題に出てきており、表示されるであろう単語などは薄々想像できたが、例題の問題を打っているときはどのような実行結果が出るのかはいまいち感触が掴めなかった。コンパイルをして実行してみると「Price? =>」と「Payment? =>」という表示が出た。

問題として 1234 円の品を 10000 円で買うということなので、その指示に従い、「Price? =>」に「1234」、「Payment? =>」に「10000」を入力し、実行。すると上記の実験結果になった。

結果の欄に表示された「5000-yen note = 1」。つまりこれが 5000 円札は一枚ということを示していると想定し、下記の問題を解いた。

◎問題 1234 円の買い物をして 1 万円札を出したときの、お釣りの札と硬貨の枚数を求めるプログラムを作成せよ。

a. `scanf()`関数を用いて、価格と支払い金額を入力せよ。

◎ソースコード(balance01.c)

```
/*
Program                               :          balance01.c
Student-ID                             :          065702G
Author                                 :          ARAKAKI, Tomonori
UpDate                                 :          2006/05/12(Fri)
Comments                               :          Payment    &    Balance
*/

#include <stdio.h>

int main()
{
    int price = 1234, pay = 10000;
    int balance, amount;

    /***** scanf */
    printf("Price?  => "); scanf("%d",&price);
    printf("Payment? => "); scanf("%d",&pay);
    printf("----\n");

    /***** balance */
    balance = pay - price;
    printf("price = %d, ",price);
    printf("payment = %d, balance = %d\n",pay,balance);
    printf("----\n");
}
```

```
/****** 5000-yen */
    amount = balance / 5000;
    balance = balance % 5000;
    printf("5000-yen note = %d\n",amount);

/****** 1000-yen */
    amount = balance / 1000;
    balance = balance % 1000;
    printf("1000-yen note = %d\n",amount);

/****** 500-yen */
    amount = balance / 500;
    balance = balance % 500;
    printf("500-yen note = %d\n",amount);

/****** 100-yen */
    amount = balance / 100;
    balance = balance % 100;

/****** 5-yen */
    amount = balance / 5;
    balance = balance % 5;
    printf("5-yen note = %d\n",amount);

/****** 1-yen */
    amount = balance / 1;
    balance = balance % 1;
    printf("1-yen note = %d\n",amount);

    return(0);
}
```

◎実行結果

```
[nw0602:~] j06002% cc -o coin balance01.c
[nw0602:~] j06002% ./coin
Price?  => 1234
Payment? => 10000
-----
price = 1234, payment = 10000, balance = 8766
-----
5000-yen note = 1
1000-yen note = 3
500-yen note = 1
100-yen note = 2
50-yen note = 1
10-yen note = 1
5-yen note = 1
1-yen note = 1
```

◎考察

例題の結果をもとに、1234 円の買い物をして 1 万円札を出したときの、お釣りの札と硬貨の枚数を表示するプログラム。

実行したところ「5000 円札が 1 枚、1000 円札 3 枚、500 円玉が 1 枚、100 円玉 2 枚、50 円玉 1 枚、10 円玉 1 枚、5 円玉 1 枚、1 円玉 1 枚」という結果が出た。初め、このプログラムを書いたときは「8766 円」を 5000、1000、500、100、50、10、5、1 で割っていくものだと思い、実行結果は「5000-yen note = 1 1000-yen note = 8 500-yen note = 17 100-yen note = 86 50-yen note = 175 10-yen note = 876 5-yen note = 1753 1-yen note = 8766」となると予想していたのだが、きちんとあまりのみを計算していて驚いた。

教科書を見てみると、P41 の表 4.3 に単純な演算式が載っており、単に割るだけだと「/」を使うとのことだった。

◎ 問題

1234円の買い物をして1万円札を出したときの、お釣りの札と硬貨の枚数を求めるプログラムを作成せよ。

◎a. 例題の変数名を変え、自分自身で考えた変数名にせよ

◎ソースコード(balance02.c)

```
/*
Program                               :          balance02.c
Student-ID                             :                065702G
Author                                 :          ARAKAKI, Tomonori
UpDate                                 :          2006/05/15(Mon)
Comments                               :          Payment      &      Balance
*/

#include <stdio.h>

int main()
{
int NEDAN = 1234, OSHIHARAI = 10000;
int AMARI, amount;

/***** scanf */
printf("NEDAN?      => "); scanf("%d",&NEDAN);
printf("OSHIHARAI? => "); scanf("%d",&OSHIHARAI);
printf("----\n");

/***** AMARI */
AMARI = OSHIHARAI - NEDAN;
printf("NEDAN = %d, ",NEDAN);
printf("OSHIHARAI = %d, AMARI = %d\n",OSHIHARAI,AMARI);
```

```
printf("----\n");
```

```
/****** 5000-yen */
```

```
amount = AMARI / 5000;
```

```
AMARI = AMARI % 5000;
```

```
printf("5000-yen note = %d\n",amount);
```

```
/****** 1000-yen */
```

```
amount = AMARI / 1000;
```

```
AMARI = AMARI % 1000;
```

```
printf("1000-yen note = %d\n",amount);
```

```
/****** 500-yen */
```

```
amount = AMARI / 500;
```

```
AMARI = AMARI % 500;
```

```
printf("500-yen note = %d\n",amount);
```

```
/****** 100-yen */
```

```
amount = AMARI / 100;
```

```
AMARI = AMARI % 100;
```

```
printf("100-yen note = %d\n",amount);
```

```
/****** 50-yem */
```

```
amount = AMARI / 50;
```

```
AMARI = AMARI % 50;
```

```
printf("50-yen note = %d\n",amount);
```

```
/****** 10-yen */
```

```
amount = AMARI / 10;
```

```
AMARI = AMARI % 10;
```

```
printf("10-yen note = %d\n",amount);
```



```

/***** 5-yen */
amount = AMARI / 5;
AMARI = AMARI % 5;
printf("5-yen note = %d\n",amount);

/***** 1-yen */
amount = AMARI / 1;
AMARI = AMARI % 1;
printf("1-yen note = %d\n",amount);

return(0);
}

```

◎実行結果

```

[nw0602:~] j06002% cc -o hensuu balance02.c
[nw0602:~] j06002% ./hensuu
NEDAN?      => 1234
OSHIHARAI? => 10000
-----
NEDAN = 1234, OSHIHARAI = 10000, AMARI = 8766
-----
5000-yen note = 1
1000-yen note = 3
500-yen note = 1
100-yen note = 2
50-yen note = 1
10-yen note = 1
5-yen note = 1
1-yen note = 1

```

◎ 考察

このプログラムは、例題のプログラムの変数を変えたものである。
例題の変数である「price」、「pay」「balance」をそれぞれ「NEDAN」、
「OSHIHARAI」、「AMARI」と置き換えて実行した。
手書きだったので馴染みの誤字エラーメッセージが出てきて、英文を
読んでエラー部分をデバックするのに1時間ほどかかっていた自分です。
根底のプログラムは先の課題で設計していたので、基本的に代入して
結果を出したのみだった。

◎問題 工夫

```
/*
Program                :          balance02.c
Student-ID             :                065702G
Author                 :          ARAKAKI, Tomonori
UpDate                :          2006/05/15(Mon)
Comments              :          Payment    &    Balance
*/

#include <stdio.h>

int main()
{
int NEDAN = 1234, OSHIHARAI = 10000;
int AMARI, amount;
```

```

/***** scanf */
printf("NEDAN?   => "); scanf("%x",&NEDAN);
printf("OSHIHARAI? => "); scanf("%x",&OSHIHARAI);
printf("----\n");

/***** AMARI */
AMARI = OSHIHARAI - NEDAN;
printf("NEDAN = %d, ",NEDAN);
printf("OSHIHARAI = %d, AMARI = %d\n",OSHIHARAI,AMARI);
printf("----\n");

/***** 5000-yen */
amount = AMARI / 5000;
AMARI = AMARI % 5000;
printf("5000-yen note = %d\n",amount);

/***** 1000-yen */
amount = AMARI / 1000;
AMARI = AMARI % 1000;
printf("1000-yen note = %d\n",amount);

/***** 500-yen */
amount = AMARI / 500;
AMARI = AMARI % 500;
printf("500-yen note = %d\n",amount);

/***** 100-yen */
amount = AMARI / 100;
AMARI = AMARI % 100;
printf("100-yen note = %d\n",amount);

/***** 50-yen */

```

```

amount = AMARI / 50;
AMARI = AMARI % 50;
printf("50-yen note = %d\n",amount);

/***** 10-yen */
amount = AMARI / 10;
AMARI = AMARI % 10;
printf("10-yen note = %d\n",amount);

/***** 5-yen */
amount = AMARI / 5;
AMARI = AMARI % 5;
printf("5-yen note = %d\n",amount);

/***** 1-yen */
amount = AMARI / 1;
AMARI = AMARI % 1;
printf("1-yen note = %d\n",amount);

return(0);
}

```

◎実行結果

```

NEDAN?      => 1234
OSHIHARAI? => 10000
-----
NEDAN = 4660, OSHIHARAI = 65536, AMARI = 60876
-----
5000-yen note = 12
1000-yen note = 0
500-yen note = 1
100-yen note = 3

```

50-yen note = 1

10-yen note = 2

5-yen note = 1

1-yen note = 1

◎考察

例題の作品を 10 進数ではなく、16 進数で作成。

それでも問題なく動いた。入れた数字によってやはりというか 10 進数とは異なった実行結果が出て、(%d) と (%x) の違いをはっきりと理解できた。

◎問題 int型整数の下限・上限の値について、簡単なプログラムと実行結果を示し考察せよ。

テキスト PP. 68 基数 16 の表記法を用いたプログラムを考えること。

◎ソースコード(jougen.c)

```
/*  
program           :           jougen.c  
Student-ID       :           065702G  
Author           :           ARAKAKI, Tomonori  
UpDate          :           2006/05/16(Tue)  
Comments        :           jougen           sirabe  
*/
```

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

```
int main()
```

```
{
```

```
    int jougen = 0;
```

```
/****** scanf */
printf("nyuuryoku => "); scanf("%x",&jougen);

/****** jougen */
printf("syuturyoku = %d\n",jougen);

return(0);
}
```

◎実行結果

```
[tomonori-arakaki-no-ibook-g4:~] j06002% ./a.out
nyuuryoku => f
syuturyoku = 15
[tomonori-arakaki-no-ibook-g4:~] j06002% ./a.out
nyuuryoku => ff
syuturyoku = 255
[tomonori-arakaki-no-ibook-g4:~] j06002% ./a.out
nyuuryoku => fff
syuturyoku = 4095
[tomonori-arakaki-no-ibook-g4:~] j06002% ./a.out
nyuuryoku => ffff
syuturyoku = 65535
[tomonori-arakaki-no-ibook-g4:~] j06002% ./a.out
nyuuryoku => fffff
syuturyoku = 1048575
[tomonori-arakaki-no-ibook-g4:~] j06002% ./a.out
nyuuryoku => fffffff
syuturyoku = 16777215
[tomonori-arakaki-no-ibook-g4:~] j06002% ./a.out
```

nyuuryoku => ffffffff
syuturyoku = 268435455
[tomonori-arakaki-no-ibook-g4:~] j06002% ./a.out
nyuuryoku => ffffffff
syuturyoku = -1
[tomonori-arakaki-no-ibook-g4:~] j06002% ./a.out
nyuuryoku => 1fffffff
syuturyoku = 536870911
[tomonori-arakaki-no-ibook-g4:~] j06002% ./a.out
nyuuryoku => 2fffffff
syuturyoku = 805306367
[tomonori-arakaki-no-ibook-g4:~] j06002% ./a.out
nyuuryoku => 3fffffff
syuturyoku = 1073741823
[tomonori-arakaki-no-ibook-g4:~] j06002% ./a.out
nyuuryoku => 4fffffff
syuturyoku = 1342177279
[tomonori-arakaki-no-ibook-g4:~] j06002% ./a.out
nyuuryoku => 5fffffff
syuturyoku = 1610612735
[tomonori-arakaki-no-ibook-g4:~] j06002% ./a.out
nyuuryoku => 6fffffff
syuturyoku = 1879048191
[tomonori-arakaki-no-ibook-g4:~] j06002% ./a.out
nyuuryoku => 7fffffff
syuturyoku = 2147483647
[tomonori-arakaki-no-ibook-g4:~] j06002% ./a.out
nyuuryoku => 8fffffff
syuturyoku = -1879048193
[tomonori-arakaki-no-ibook-g4:~] j06002% ./a.out
nyuuryoku => 80000000
syuturyoku = -2147483648

◎考察

int 型整数の上限下限を調べるために書いたプログラム。

最初、何をしたらいいのかが全く分からなかったが、授業内容の復習から始めた。すると、2進数の上限下限の例が載っていたのでそれを活用。16進数を10進数表示するというプログラムを、試行錯誤しながら必死に書き、その後「f」を重ね、正と負の表示の入れ替わりを探すため数度試行した。すると、「ffffffff」で符号が正から負へと入れ替わった。すなわち、正の上限と負の下限は「1ffffffff」～「ffffffff」の間にあるということになる。

よって、今度は「1ffffffff」から順に試行していき、「8ffffffff」で符号が入れ替わった。

そのことから符号の入れ替わりは「7ffffffff」～「8ffffffff」の間にあると推測でき、「7ffffffff」の+1である「80000000」をで実行すると「-2147483648」という負の結果が出た。

ゆえに、正負は「7ffffffff」と「80000000」の間で行われていると推測でき、int 型の上限は「2147483647」、下限は「-2147483648」という結果になった。

◎問題 エラーについて考察せよ。

◎エラーソースコード(jougen.c)

```
  ]/*  
  program                :                jougen.c  
  Student-ID            :                065702G  
  Author                :                ARAKAKI, Tomonori  
  UpDate                :                2006/05/16(Tue)  
  Comments              :                jougen          sirabe  
  */  
  
#include <stdio.h>  
  
int main()  
{  
    int jougen = 0  
  
    /****** scanf */  
    printf("nyuuryoku => "); scanf("%x",&jougen);  
  
    /****** jougen */  
    printf("syuturyoku = %d\n",jougen);  
  
    return(0);  
}
```

◎実行結果

```
[nw0602:~] j06002% cc joucen.c
powerpc-apple-darwin8-gcc-4.0.1: joucen.c: No such file or
directory
powerpc-apple-darwin8-gcc-4.0.1: no input files
[nw0602:~] j06002% cc jougen.c
jougen.c:1: error: parse error before ']' token
jougen.c: In function 'main':
jougen.c:16: error: parse error before 'printf'
```

◎考察

今回のレポートで起こったエラーは上記のものだけでした。原因は前回と同じく余分な語があったり誤字なのですが、まず最初のエラーは「cc joucen.c」。コンパイルする対象がありませんとでました。正しいのは「cc jougen.c」で、「c」と「g」の入力ミスでした。2つ目のミスは、上記のソースコードの一番始めの部分で「/*」の前に『[]が余分に挿入されていたことと、「int jougen = 0」の直後の「;」が抜けていたことでした。

初めて間もないですが、このタイプミスが結構響いてきます。見つけることに時間をとられて、仕上がるまでにかかなりの時間がかかっていて練習の必要性を多いに感じました。