

1 scanf()関数による標準入力と基本演算子
例題 balance.c

<ソースコード>

```
/*
Program   : balance.c
Student-ID : 053772A
Author    : TANIGUCHI,Yuji
UpDate    : 2005/04/27(Wed)
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);

    return(0);
}
```

<出力結果>

```
nw0945:reibun e095745$ ./balance
Price?  => 3980
Payment? => 10000
----
```

```
price = 3980, payment = 10000, balance = 6020
```

```
----
```

```
5000-yen note = 1
```

```
nw0945:reibun e095745$
```

<解説>

まず始めに「int」を使って変数の整数系の値を格納する場所を指定しています。変数の名前は price、pay、balance、amount の四つです。あらかじめ price、pay には 1 2 3 4 と 1 0 0 0 0 の数が入っています。この数は入れなくてもプログラムには問題ないですが、もし数字以外の文字を入力した場合あらかじめ入った数字が出力されます。

次にソースコードの printf("Price? => "); scanf("%d",&price); の説明をします。出力結果からも分かるように printf で Price? => と表示させています。関数 scanf は数値を読み込むものです。scanf("%d",&price); の "%d" は 10 進数で数値を読み込むように指定し、&price で読み込んだ数値を price に格納するように指定しています。

次にソースコードの printf("price = %d, ",price); について説明します。一見関数 printf で price = %d と出力されそうですが、%d は先ほど格納した price の値が出力されます。

次に何行目の printf("payment = %d, balance = %d\n",pay,balance); を説明します。これは、すぐ前と同じようですが

```
printf("payment = %d, balance = %d\n",pay,balance);
```

という風に、表示先を指定しています。

次に balance = pay - price; を説明します。これは pay から price を引きその差の値を balance に収納せよという意味です。

次に amount = balance / 5000; を説明します。これは balance から 5000 を割り、その商を amount に収納せよという意味です。

次に balance = balance % 5000; を説明します。これは balance から 5000 を割り、その余りを balance に収納せよという意味です。

<考察>

あらかじめ、四則演算の演算子と関数や int を使って自分でいろいろプログラムを作ってみたので、例文のしたいことはすぐ理解できました。自分で作ったプログラムは無駄が多くて見づらいものだったので、例文の /* */ のコメントを利用して見やすくしているなど学ぶことが多い例文だと思いました。

ソースコード何行目の balance = balance % 5000; は計算しているものの、表示させてはいないので意味のない計算となっていると思います。次の問題のヒントにしたのだらうと思います。

2. 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

<ソースコード>

```
/*
Program   : Keisan.c
Student-ID : 095745D
Author    : Jun NISHISHIMAMOTO
UpDate    : 2009/05/20
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);

    /****** 2000-yen */
    amount = balance / 2000;
    balance = balance % 2000;
    printf("2000-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;
printf("100-yen note = %d\n",amount);

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

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

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

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

return(0);
}

```

<出力結果>

```

nw0945:kadai2 e095745$ ./keisan001
Price? => 1234
Payment? => 10000
----
price = 1234, payment = 10000, balance = 8766
----
5000-yen note = 1

```

```
2000-yen note = 1
1000-yen note = 1
500-yen note = 1
100-yen note = 2
50-yen note = 1
10-yen note = 1
5-yen note = 1
1-yen note = 1
nw0945:kadai2 e095745$
```

<解説>

何行目のコメント以外、何行目まで例文を使用している。例文の5000円の処理の方法を参考にして、おつりに2000円から1円玉まですべての紙幣と硬貨の枚数を表示できるようにした。

<考察>

例文が参考になったおかげで思ったより早くできた。私は変数が一度数値を設定したらもう書き換えられないと思っていた。しかし、全問であったヒントをもとに一気に書き上げることができた。もし例文を見ずにこのプログラムを作ったとしたら、紙幣と枚数の数だけ変数をつくって、ながながしく分かりづらくなったと思う。

a. scanf()関数を用いて、価格と支払い金額を入力せよ

ソースコードは2と同じなので省略する。

<出力結果①3980円のものに5000円払った場合>

```
nw0945:kadai2 e095745$ ./keisan001
Price?  => 3980
Payment? => 5000
----
price = 3980, payment = 5000, balance = 1020
----
5000-yen note = 0
2000-yen note = 0
1000-yen note = 1
500-yen note = 0
100-yen note = 0
50-yen note = 0
10-yen note = 2
5-yen note = 0
1-yen note = 0
nw0945:kadai2 e095745$
```

<出力結果②198円のものに10000円払った場合>

nw0945:kadai2 e095745\$./keisan001

Price? => 198

Payment? => 10008

price = 198, payment = 10008, balance = 9810

5000-yen note = 1

2000-yen note = 2

1000-yen note = 0

500-yen note = 1

100-yen note = 3

50-yen note = 0

10-yen note = 1

5-yen note = 0

1-yen note = 0

nw0945:kadai2 e095745\$

<出力結果③123456789円のものに987654321円払った場合>

nw0945:kadai2 e095745\$./keisan001

Price? => 123456789

Payment? => 987654321

price = 123456789, payment = 987654321, balance = 864197532

5000-yen note = 172839

2000-yen note = 1

1000-yen note = 0

500-yen note = 1

100-yen note = 0

50-yen note = 0

10-yen note = 3

5-yen note = 0

1-yen note = 2

nw0945:kadai2 e095745\$

<考察>

三つの結果でやってみて、自分でも計算してみたが全く問題なく終了した。どこも問題なかったのだろう。しかし③であり得ない値段の商品にあり得ない金額を支払った結果、計算結果は問題ないが5000円札が以上な枚数必要だと表示されている。

普通に考えると店の人に嫌がられる結果だが、コンピュータは忠実だ。

b. 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>

```
*  
Program   : Keisan002  
Student-ID : 095745D  
Author    : Jun NISHISHIMAMOTO  
UpDate    : 2009/05/20  
Comments  : Payment & Balance  
*/
```

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

```
int main()  
{  
  
    int nedan = 1234, oazukari = 10000;  
    int oturi, amari;
```

※以下のソースはaで用いたものの変数を変更しただけで、変わりがないので省略する。

<出力結果>

```
nw0945:kadai2 e095745$ ./keisan002  
prise? => 1234  
Payment? => 10000  
----  
price = 1234, payment = 10000, balance = 8766  
----  
5000-yen note = 1  
2000-yen note = 1  
1000-yen note = 1  
500-yen note = 1  
100-yen note = 2  
50-yen note = 1  
10-yen note = 1  
5-yen note = 1  
1-yen note = 1  
nw0945:kadai2 e095745$
```

<解説>

変数をpriseはnedanに、paymentはoazukariに、balanceはoturiに、amountはamariに変更した。

<考察>

出力された画面には何の変化も見られないが、ソースコードは変数がすべて変換されている。変数を一気に置き換える方法を探したが、なかったので一つ一つタイプし直した。結果何回か失敗しておつりがマイナスの値になったりして困った。タイプミスは気をつけようと思う。

c. 工夫…！

<ソースコード>]

```
/*
Program   : Keisan003
Student-ID : 095745D
Author    : Jun NISHISHIMAMOTO
UpDate    : 2009/05/20
Comments  : Payment & Balance
*/

#include <stdio.h>

int main()
{

    int nedan = 1234, oazukari = 10000;
    int oturi, amari;

    /****** scanf */
    printf("おいくらになりますか? => "); scanf("%d",&nedan);
    printf("お預かりはいくらですか? => "); scanf("%d",&oazukari);
    printf("----\n");

    /****** oturi */
    oturi = oazukari - nedan ;
    printf("お値段は%d円になります\n",nedan);
    printf("お客様から%d円預かりました\n",oazukari);
    printf("おつりは%d円になります\n",oturi);
    printf("ありがとうございました\n");
    printf("----\n");

    /****** 10000-yen */
    amari = oturi / 10000;
    oturi = oturi % 10000;
    printf("10000円札は%d枚\n",amari);

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



```
amari = oturi / 5000;
oturi = oturi % 5000;
printf("5000円札は%d枚\n",amari);

/***** 2000-yen */
amari = oturi / 2000;
oturi = oturi % 2000;
printf("2000円札は%d枚\n",amari);

/***** 1000-yen */
amari = oturi / 1000;
oturi = oturi % 1000;
printf("1000円札は%d枚\n",amari);

/***** 500-yen */
amari = oturi / 500;
oturi = oturi % 500;
printf("500円玉は%d枚\n",amari);

/***** 100-yen */
amari = oturi / 100;
oturi = oturi % 100;
printf("100円玉は%d枚\n",amari);

/***** 50-yen */
amari = oturi / 50;
oturi = oturi % 50;
printf("50円玉は%d枚\n",amari);

/***** 10-yen */
amari = oturi / 10;
oturi = oturi % 10;
printf("10円玉は%d枚\n",amari);

/***** 5-yen */
amari = oturi / 5;
oturi = oturi % 5;
printf("5円玉は%d枚\n",amari);

/***** 1-yen */
amari = oturi / 1;
printf("1円玉は%d枚\n",amari);
return(0);
}
```

<出力結果①>

nw0945:kadai2 e095745\$./kesan003

おいくらになりますか? => 1234

お預かりはいくらですか? => 10000

お値段は1234円になります

お客様から10000円預かりました

おつりは8766円になります

ありがとうございました

10000円札は0枚

5000円札は1枚

2000円札は1枚

1000円札は1枚

500円玉は1枚

100円玉は2枚

50円玉は1枚

10円玉は1枚

5円玉は1枚

1円玉は1枚

<出力結果②>

nw0945:kadai2 e095745\$./kesan003

おいくらになりますか? => 123456789

お預かりはいくらですか? => 987654321

お値段は123456789円になります

お客様から987654321円預かりました

おつりは864197532円になります

ありがとうございました

10000円札は86419枚

5000円札は1枚

2000円札は1枚

1000円札は0枚

500円玉は1枚

100円玉は0枚

50円玉は0枚

10円玉は3枚

5円玉は0枚

1円玉は2枚

nw0945:kadai2 e095745\$

<解説>

ソース自体にそんなに変化はないが、一万円のおつりもできるようになっている。
英語で表示していた部分を日本語に変えている。

<考察>

おつりで一万円を使うことはないけれども、先ほどの問題で5000円札が大量に必要なことを解消しようと思って作ってみた。
工夫した点は出力結果に現れているように日本語を使って読みやすくしたことだ。

3. 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。

<ソースコード>

```
/* file jougen */
#include<stdio.h>

int main()
{
    int juroku;

    scanf("%x",&juroku);

    printf("%d\n",juroku);

    return(0);
}
```

<出力結果>

```
nw0945:kadai2 e095745$ ./jougen
ff
255
nw0945:kadai2 e095745$ ./jougen
fffff
16777215
nw0945:kadai2 e095745$ ./jougen
fffffff
-1
nw0945:kadai2 e095745$ ./jougen
fffffffe
-2
nw0945:kadai2 e095745$ ./jougen
1000000000
0
nw0945:kadai2 e095745$ ./jougen
```

```
ffffff
-1
nw0945:kadai2 e095745$ ./jougen
ffffff
268435455
nw0945:kadai2 e095745$ ./jougen
ffffff
-1
nw0945:kadai2 e095745$ ./jougen
1ffffff
536870911
nw0945:kadai2 e095745$ ./jougen
8ffffff
-1879048193
nw0945:kadai2 e095745$ ./jougen
7ffffff
2147483647
nw0945:kadai2 e095745$ ./jougen
8f000000
-1895825408
nw0945:kadai2 e095745$ ./jougen
8e000000
-1912602624
nw0945:kadai2 e095745$ ./jougen
89000000
-1996488704
nw0945:kadai2 e095745$ ./jougen
83000000
-2097152000
nw0945:kadai2 e095745$ ./jougen
81000000
-2130706432
nw0945:kadai2 e095745$ ./jougen
80000000
-2147483648
nw0945:kadai2 e095745$ ./jougen
7ffffff
2147483647
```

<解説>

上限値下限値を知るためにこのプログラムを作りました。16進数で入力し、それを10進数で表示するシンプルなものです。

<考察>

16進数で一番大きいfを何度も入力して、その値がどうなるか調べてみました。fを9回うった数で-1と出力できたので、すぐにその近くに上限値かなにかがあると思いました。しかし、fを八回の値も-1と出力されるので入力する値が大きすぎると思い、八桁でいろいろ試してみることにしました。その結果、80000000と入力すると-214783648と出力されました。また、その一つ下の7fffffffと入力すると2147483647と出力されました。

このことから、上限値は2147483647で下限値は-214783648であるという結果にたどり着きました。

4. エラーについて考察せよ

<発生したエラー>

```
nw0945:kadai2 e095745$ cc -o keisan0002 keisan002
ld warning: in keisan002, file is not of required architecture
Undefined symbols:
  "_main", referenced from:
      start in crt1.10.5.o
ld: symbol(s) not found
collect2: ld returned 1 exit status
nw0945:kadai2 e095745$
```

<考察>

コンパイル時に起きたエラーで、keisan002のファイルをkeisan0002という名前でコンパイルできないというものだった。コマンドが間違っているのだと思い調べ直したが当たっていたので困惑した。いろいろ考えたあげくlsコマンドでファイル名を見たとき、keisan002のファイルの拡張子に.cを付け忘れるという初歩的なものだった。変更するとすぐコンパイルできた。

このあとも、ファイルを開いたが中身が真っ白で消えたと焦って、友達に名前が1文字違っていると指摘されたりと初歩的なミスが何度もあった。失敗したと思っても、コンピュータに間違いはないのだし、自分をもっと疑っていこうと思った。

<感想>

まだまだ初歩的なプログラムだったと思うが、すごくパソコンを動かしているという楽しさがあった。これからさらに難しくなっていくが楽しんでやることは忘れずにいようと思った。