

1. scanf()関数による標準入力と基本演算子

a. 例題 balance.c

```
1  /*
2   Program    : balance.c
3   Student-ID : 095739K
4   Author     : TOUME,Kouta
5   Date       : 09/05/10
6   Comment    : Payment & Balance
7  */
8
9  #include <stdio.h>
10
11 int main(){
12
13     int price = 1234, pay = 10000;
14     int balance, amount;
15
16     /***** scanf */
17     printf("Price? => "); scanf("%d",&price);
18     printf("Payment? => "); scanf("%d",&pay);
19     printf("----\n");
20
21     /***** balance */
22     balance = pay - price;
23     printf("price = %d, ",price);
24     printf("payment = %d, balance = %d\n",pay,balance);
25     printf("----\n");
26
27     /***** 5000-yen */
28     amount = balance / 5000;
29     balance = balance % 5000;
30     printf("5000-yen note = %d\n",amount);
31
32     return(0);
33 }
```

入 力

```
Price?  => 2500
Payment? => 10000
```

出力結果

```
----
price = 2500, payment = 10000, balance = 7500
----
5000-yen note = 1
```

考 察

例題プログラム27~30行目は、

- ・変数 balance の値を 5000 で割りその商を、変数 amount に代入。
- ・変数 balance の値を 5000 で割りその余りを、変数 balance に代入。
- ・変数 amount の値を出力。

という処理を行い、お釣りの 5000 円札の枚数を出力している。

また、5000 で割った余りを変数 balance に代入しているため、この値を 2000 で割ったときの商の値を出力すれば、お釣りの 2000 円札の枚数が出力できると考えられる。

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。

```
1  /*
2  Program   : balance_1.c
3  Student-ID : 095739K
4  Author    : TOUME,Kouta
5  Date      : 09/05/10
6  Comment   : Payment & Balance
7  */
8
9  #include <stdio.h>
10
11 int main(){
12
13     int price = 1234, pay = 10000;
14     int balance, amount;
15
16     /****** scanf */
17     printf("Price? => "); scanf("%d",&price);
18     printf("Payment? => "); scanf("%d",&pay);
19     printf("----\n");
20
21     /****** balance */
22     balance = pay - price;
23     printf("price = %d, ",price);
24     printf("payment = %d, balance = %d\n",pay,balance);
25     printf("----\n");
26
27     /****** 5000-yen */
28     amount = balance / 5000;
29     balance = balance % 5000;
30     printf("5000-yen note = %d\n",amount);
31
32     /****** 2000-yen */
33     amount = balance / 2000;
34     balance = balance % 2000;
35     printf("2000-yen note = %d\n",amount);
36
37     /****** 1000-yen */
38     amount = balance / 1000;
39     balance = balance % 1000;
40     printf("1000-yen note = %d\n",amount);
41
42     /****** 500-yen */
43     amount = balance / 500;
44     balance = balance % 500;
45     printf("500-yen coin = %d\n",amount);
46
47     /****** 100-yen */
48     amount = balance / 100;
49     balance = balance % 100;
50     printf("100-yen coin = %d\n",amount);
51
52     /****** 50-yen */
53     amount = balance / 50;
54     balance = balance % 50;
55     printf("50-yen coin = %d\n",amount);
56
```

```

57  /***** 10-yen */
58  amount = balance / 10;
59  balance = balance % 10;
60  printf("10-yen coin = %d\n",amount);
61
62  /***** 5-yen */
63  amount = balance / 5;
64  balance = balance % 5;
65  printf("5-yen coin = %d\n",amount);
66
67  /***** 1-yen */
68  printf("1-yen coin = %d\n",balance);
69
70  return(0);
71  }

```

a. scanf()関数を用いて、価格と支払い金額を入力せよ。

入 力

```

Price?  => 1234
Payment? => 10000

```

出力結果

```

----
price = 1234, payment = 10000,
balance = 8766
----
5000-yen note = 1
2000-yen note = 1
1000-yen note = 1
500-yen coin = 1
100-yen coin = 2
50-yen coin = 1
10-yen coin = 1
5-yen coin = 1
1-yen coin = 1

```

考 察

例題プログラムの27~30行目の5000という値を、2000、1000、500、…、5に変えると、それぞれ、お釣りの2000円札の枚数、1000円札の枚数、500円硬貨の枚数、…、5円硬貨の枚数となり、変数balanceを5で割ったときの余りの値を出力することで、1円硬貨の枚数となる。

b. 例題の変数名を変え、自分自身で考えた変数名にせよ。

```

13  int price = 1234, money = 10000;
14  int balance, total;

18  printf("Payment? => "); scanf("%d",&money);

27  /***** 5000-yen */
28  total  = balance / 5000;
29  balance = balance % 5000;
30  printf("5000-yen note = %2d\n",total);

```

※32~68行目も同様に対応する変数を変更する。

入 力

Price? => 2768
Payment? => 10000

出力結果

```
-----  
price = 2768, payment = 10000, balance = 7232  
-----
```

```
5000-yen note = 1  
2000-yen note = 1  
1000-yen note = 0  
500-yen coin = 0  
100-yen coin = 2  
50-yen coin = 0  
10-yen coin = 3  
5-yen coin = 0  
1-yen coin = 2
```

考 察

13,14行目の変数 pay を money に、変数 amount を total という変数に変更し、それぞれ対応する変数を変更したが、変更してもプログラムは、変わらず動作した。
これより、プログラム内で使われる変数はどの様な変数であっても構わないが、分かりやすい変数を設定することが重要だと思われる。

c. 工夫…!

```
7  /***** scanf */  
8  printf("-----\n");  
9  printf("Price? => "); scanf("%d",&price);  
10 printf("Payment? => "); scanf("%d",&money);  
11 printf("-----\n");  
12  
13 /***** balance */  
14 balance = money - price;  
15 printf("price = %6d\n",price);  
16 printf("payment = %6d\nbalance = %6d\n",money,balance);  
17 printf("-----\n");  
18  
19 /***** 5000-yen */  
20 total = balance / 5000;  
21 balance = balance % 5000;  
22 printf("5000-yen note = %2d\n",total);  
23  
24 /***** 2000-yen */  
25 total = balance / 2000;  
26 balance = balance % 2000;  
27 printf("2000-yen note = %2d\n",total);  
28  
29 /***** 1000-yen */  
30 total = balance / 1000;  
31 balance = balance % 1000;  
32 printf("1000-yen note = %2d\n",total);  
33  
34 /***** 500-yen */  
35 total = balance / 500;  
36 balance = balance % 500;  
37 printf(" 500-yen coin = %2d\n",total);  
38
```

```

39  /***** 100-yen */
40  total  = balance / 100;
41  balance = balance % 100;
42  printf(" 100-yen coin = %2d\n",total);
43
44  /***** 50-yen */
45  total  = balance / 50;
46  balance = balance % 50;
47  printf(" 50-yen coin = %2d\n",total);
48
49  /***** 10-yen */
50  total  = balance / 10;
51  balance = balance % 10;
52  printf(" 10-yen coin = %2d\n",total);
53
54  /***** 5-yen */
55  total  = balance / 5;
56  balance = balance % 5;
57  printf(" 5-yen coin = %2d\n",total);
58
59  /***** 1-yen */
60  printf(" 1-yen coin = %2d\n",balance);
61  printf("-----\n");

```

入 力

```

-----
Price?  => 1234
Payment? => 10000

```

出力結果

```

-----
price   = 1234
payment = 10000
balance = 8766
-----
5000-yen note = 1
2000-yen note = 1
1000-yen note = 1
 500-yen coin = 1
 100-yen coin = 2
  50-yen coin = 1
  10-yen coin = 1
   5-yen coin = 1
   1-yen coin = 1
-----

```

工 夫

- ・変数 price、 payment、 balance の値を、「%6d」6桁右詰めで出力。
- ・「-」で入力、出力結果の区切りを分かり易くした。
- ・出力される文字の最後を揃える。

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。

a. テキスト PP.68 基数 16 の表記法を用いたプログラミング

```
1  /*
2   Program    : int.c
3   Student-ID : 095739K
4   Author     : TOUME,Kouta
5   Date      : 09/05/10
6   Comment   : int 上限・下限
7  */
8
9  #include<stdio.h>
10 int main(){
11
12     int hex;
13
14     hex = 0x01;
15     printf("hex : 0x01      dec : %d\n",hex);
16
17     hex = 0x0f;
18     printf("hex : 0x0f      dec : %d\n",hex);
19
20     hex = 0xff;
21     printf("hex : 0xff      dec : %d\n",hex);
22
23     hex = 0xffff;
24     printf("hex : 0xffff    dec : %d\n",hex);
25
26     hex = 0x7fffffff;
27     printf("hex : 0x7fffffff dec : %d\n",hex);
28
29     hex = 0xffffffff;
30     printf("hex : 0xffffffff dec : %d\n",hex);
31
32     hex = 0xffffffff0;
33     printf("hex : 0xffffffff0 dec : %d\n",hex);
34
35     hex = 0xffffffff00;
36     printf("hex : 0xffffffff00 dec : %d\n",hex);
37
38     hex = 0xffff0000;
39     printf("hex : 0xffff0000 dec : %d\n",hex);
40
41     hex = 0xff000000;
42     printf("hex : 0xff000000 dec : %d\n",hex);
43
44     hex = 0x80000000;
45     printf("hex : 0x80000000 dec : %d\n",hex);
46
47     return(0);
48 }
```

出力結果

```
hex : 0x01      dec : 1
hex : 0x0f      dec : 15
hex : 0xff      dec : 255
hex : 0xffff    dec : 65535
hex : 0x7fffffff dec : 2147483647
hex : 0xffffffff dec : -1
hex : 0xffffffff0 dec : -16
hex : 0xffffffff00 dec : -256
hex : 0xffff0000 dec : -65536
hex : 0xff000000 dec : -16777216
hex : 0x80000000 dec : -2147483648
```

考 察

16進数の1から-1まで順番に増やしていき、int型整数の上限を求め、16進数の-1から順番に減らしていき、int型整数の下限を求めた。
出力結果より、int型整数の上限は、「2,147,483,647」となり、下限は「-2,147,483,648」となった。
また、「2,147,483,647」は、2の31乗から1を引いた値(0を含むから)と同じことから、int型整数は、32bitで表され最上位ビットの32bit目は符号ビットとなっていることがわかった。

4. エラーについて考察せよ。

エラー 1

```
balance_2.c: In function 'main' :
balance_2.c:57: error: 'amount' undeclared (first use in this function)
balance_2.c:57: error: (Each undeclared identifier is reported only once
balance_2.c:57: error: for each function it appears in.)
```

考 察

このエラーは、問2.bの変数名を変更する際に出たエラーで、「'amount'が、宣言されていない変数である」というエラーであった。
プログラムを見直してみると57行目の

```
printf("5-yen coin = %d\n",amount);
```

printf()文のオブジェクト部分のamountを変数totalに変更し忘れていたことがわかった。
オブジェクト部分を変数totalに変更するとプログラムは正しく動作した。

エラー 2

```
int.c: In function 'main' :
int.c:21: warning: integer constant is too large for 'long' type
int.c:21: warning: overflow in implicit constant conversion
```

考 察

このエラーは、int型整数の上限・下限を求める問いで、16進数(ffffffff)を10進数に変換し出力しようとした時に出たエラーである。
エラー文より、16進数(ffffffff)はint型整数の上限・下限を超えていて、int型ではこの値は扱えないこと、更に、この値を扱うには、long int型を使用する必要があるということがわかった。

5. 考 察

今回は、printf()関数の変換指定や演算子を使い、様々なプログラムを作った。

例題、問2では、演算子の「%」と「/」を組み合わせることで、お釣りのお札、硬貨の枚数を出力するプログラムを作成した。

演算子には、四則演算だけではなく、&&(アンド)や、|| (オア)、!(否定)などがあり、組み合わせて使うことで様々な処理ができることがわかった。

また、コンピュータ内の数値表現は左図のようになっていて、負の数は、2の補数を用いて表現され、4bitで、-8 ~ 7まで表現することができる。int型整数は、問3の結果から32bitで表され最上位ビットが符号ビットとなっていることがわかった。

さらに、printf()関数の変換指定に、フラグや、フィールド幅、精度などを付加することで、出力結果を分かり易くすることができ、「%x」や「%o」を指定することで、16進数や8進数を出力することができることがわかった。

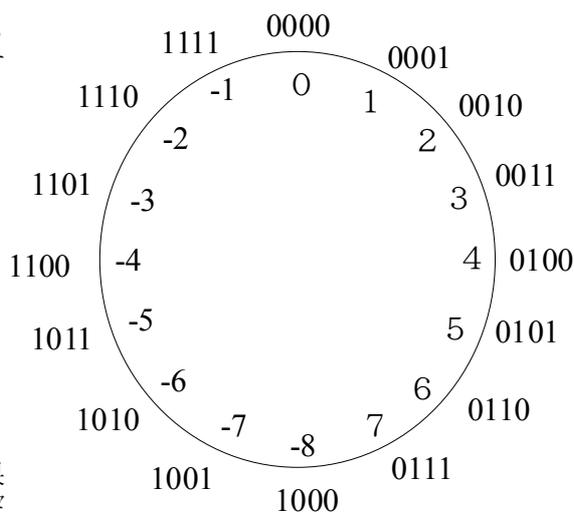


図. コンピュータ内部表現(数値)

6. 感 想

今回のプログラミングの範囲は高校のときに一度やっていたので理解し易かった。int型整数の上限・下限を考察する問いは、上限・下限のおおよその値は分かっていたけど、プログラムを作るのは結構難しかったです。また、そろそろ、学んでいない範囲になってくるので、楽しみです。

参考文献

Steve Oualline 著、谷口功 訳 『C実践プログラミング 第3版』 オーム社