

1. scanf()関数による標準入力と基本演算子

ii.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。

●ソースコードの一部（冒頭のコメントは省略）

```
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  }
```

●出力結果 ※変数"Price","Payment"にはそれぞれ 1234,10000 を入力した。

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

- ・scanf()関数を用いて5000円以下のお釣りに対応するプログラムが出来た。
- ・仮に半角数字以外の文字を入力すると、"Price"は"1234"、"Pay"は10000として扱われる。
- ・"Price"に半角数字、"Pay"に半角数字以外を入力すると、"Pay"だけに10000が与えられる。
- ・"Price"、"Pay"はそれぞれ"214748364"（32ビットの上限値）までの整数なら計算できるが、それ以上の整数だと正常に計算されなかった。
- ・"Price"が"Pay"より大きい数字な場合、"Balance"はマイナス値が代入され、それぞれのお釣り、お札と硬貨の枚数もマイナスとして計算された。

b)例題の変数名を変え、自分自身で考えた変数名にせよ。

b-1:半角数字を含む変数

●ソースコードの一部

```
9  #include <stdio.h>
10
11  int main()
12  {
13      int nedan1 = 1234, 2nedan = 10000;
14      int 0987, ne1dan;
15
16      /***** scanf */
17      printf("Price? => "); scanf("%d",&nedan1);
18      printf("Payment? => "); scanf("%d",&2nedan);
19      printf("----\n");
20
21      /***** balance */
22      0987 = 2nedan - nedan1;
23      printf("price = %d, ",nedan1);
24      printf("payment = %d, balance = %d\n",2nedan,0987);
25      printf("----\n");
26
27      /***** 5000-yen */
28      ne1dan = 0987 / 5000;
29      0987 = 0987 % 5000;
30      printf("5000-yen note = %d\n",ne1dan);
31
32      return(0);
33  }
```

●コンパイル時のエラー

```
report@2b.c:13:22: error: expected identifier or '('
  int nedan1 = 1234, 2nedan = 10000;
                        ^
report@2b.c:13:21: error: expected ';' at end of declaration
  int nedan1 = 1234, 2nedan = 10000;
                        ^
                        ;
report@2b.c:14:7: error: expected identifier or '('
  int 0987, ballance;
      ^
report@2b.c:18:40: error: invalid suffix 'nedan' on integer constant
  printf("Payment? => "); scanf("%d",&2nedan);
                                   ^
report@2b.c:22:4: error: invalid digit '9' in octal constant
  0987 = 2nedan - nedan1;
      ^
report@2b.c:22:11: error: invalid suffix 'nedan' on integer constant
  0987 = 2nedan - nedan1;
              ^
report@2b.c:24:42: error: invalid suffix 'nedan' on integer constant
  printf("payment = %d, balance = %d\n",2nedan,0987);
                                                ^
report@2b.c:28:3: error: use of undeclared identifier 'ballance'
  ballance = 0987 / 5000;
  ^
report@2b.c:28:16: error: invalid digit '9' in octal constant
  ballance = 0987 / 5000;
                  ^
report@2b.c:29:4: error: invalid digit '9' in octal constant
  0987 = 0987 % 5000;
      ^
report@2b.c:29:11: error: invalid digit '9' in octal constant
  0987 = 0987 % 5000;
              ^
report@2b.c:30:33: error: use of undeclared identifier 'ballance'
  printf("5000-yen note = %d\n",ballance);
                                   ^
```

●エラーの意味

- ・ 13 行目:22 文字目:識別子もしくは '(' が無い
- ・ 13 行目:12 文字目: ';' が記述の終わりに無い
- ・ 14 行目:7 文字目:識別子もしくは '(' が無い
- ・ 18 行目:40 文字目:整数定数に無効な接尾辞 'nedan'
- ・ 22 行目:4 文字目:8 進数では '9' は無効な数字です
- ・ 22 行目:11 文字目:整数定数に無効な接尾辞 'nedan'
- ・ 24 行目:42 文字目:整数定数に無効な接尾辞 'nedan'
- ・ 28 行目:3 文字目: 'ballance' は宣言されていない識別子です
- ・ 28 行目:16 文字目:8 進数では '9' は無効な数字です
- ・ 29 行目:4 文字目:8 進数では '9' は無効な数字です
- ・ 29 行目:11 文字目:8 進数では '9' は無効な数字です
- ・ 30 行目:33 文字目: 'ballance' は宣言されていない識別子です

●考察

- ・ 先頭が数字の変数だけがエラーとして出力された。
- ・ 数字の変数は 8 進数で表されると考えられる。

●修正したソースコードの一部

```
9 #include <stdio.h>
10
11 int main()
12 {
13     int nedan1 = 1234, 2nedan = 10000;
14     int 0987, ne1dan;
15
16     /***** scanf */
17     printf("Price? => "); scanf("%d",&nedan1);
18     printf("Payment? => "); scanf("%d",&2nedan);
19     printf("----\n");
20
21     /***** balance */
22     0987 = 2nedan - nedan1;
23     printf("price = %d, ",nedan1);
24     printf("payment = %d, balance = %d\n",2nedan,0987);
25     printf("----\n");
26
27     /***** 5000-yen */
28     ne1dan = 0987 / 5000;
29     0987 = 0987 % 5000;
30     printf("5000-yen note = %d\n",ne1dan);
31
32     return(0);
33 }
```

●出力結果 ※変数"nedan1"、"nedan2"にはそれぞれ 1234、10000 を入力した

```
Price? => 1234
Payment? => 10000
----
price = 1234, payment = 10000, balance = 8766
----
5000-yen note = 1
```

●考察

- ・適切な変数にすることで、狙い通りの出力ができた。

b-2:全角日本語を含む変数

●ソースコードの一部

```
9 #include <stdio.h>
10
11 int main()
12 {
13     int 価格 = 1234, 支払い = 10000;
14     int 差額, 余り;
15
16     /***** scanf */
17     printf("Price? => "); scanf("%d",&価格);
18     printf("Payment? => "); scanf("%d",&支払い);
19     printf("----\n");
20
21     /***** balance */
22     差額 = 支払い - 価格;
23     printf("price = %d, ",価格);
24     printf("payment = %d, balance = %d\n",支払い,差額);
25     printf("----\n");
26
27     /***** 5000-yen */
28     余り = 差額 / 5000;
29     差額 = 差額 % 5000;
30     printf("5000-yen note = %d\n",余り);
31
32     return(0);
33 }
```

●コンパイル時のエラー

```
report@2b2.c:13:7: error: expected identifier or '('
  int 価格 = 1234, 支払い = 10000;
      ^
report@2b2.c:14:7: error: expected identifier or '('
  int 差額, 余り;
      ^
report@2b2.c:17:39: error: expected expression
  printf("Price? => "); scanf("%d",&価格);
                                  ^
report@2b2.c:18:39: error: expected expression
  printf("Payment? => "); scanf("%d",&支払い);
                                  ^
report@2b2.c:22:3: error: expected expression
  差額 = 支払い - 価格;
      ^
report@2b2.c:23:25: error: expected expression
  printf("price = %d, ", 価格);
                        ^
report@2b2.c:24:41: error: expected expression
  printf("payment = %d, balance = %d\n", 支払い, 差額);
                                          ^
report@2b2.c:28:3: error: expected expression
  余り = 差額 / 5000;
      ^
report@2b2.c:29:3: error: expected expression
  差額 = 差額 % 5000;
      ^
report@2b2.c:30:33: error: expected expression
  printf("5000-yen note = %d\n", 余り);
                                  ^
```

●エラーの意味

- ・ 13 行目:7 文字目:識別子もしくは'('が無い
- ・ 14 行目:7 文字目:識別子もしくは'('が無い
- ・ 17 行目:39 文字目:予想される表現
- ・ 18 行目:39 文字目:予想される表現
- ・ 22 行目:3 文字目:予想される表現
- ・ 23 行目:25 文字目:予想される表現
- ・ 24 行目:41 文字目:予想される表現
- ・ 28 行目:3 文字目:予想される表現
- ・ 29 行目:3 文字目:予想される表現
- ・ 30 行目:33 文字目:予想される表現

●考察

- ・ 全角日本語を変数に指定した場合、うまく出力できなかった。
- ・ 全角文字は変数として扱えないと考えられる。

c)工夫…!

●ソースコードの一部

```
9  #include <stdio.h>
10
11  int main()
12  {
13      int price = 1234, pay = 10000;
14      int balance, amount;
15      printf("変数宣言を終了しました\n");
16      /***** scanf */
17      printf("合計の値段はいくらですか? => "); scanf("%d",&price);
18      printf("いくら払いますか? => "); scanf("%d",&pay);
19      printf("----\n");
20      if(price > pay){
21          printf("ERROR!値段が支払いより多いです\n");
22      }
23      else{
24          /***** balance */
25          balance = pay - price;
26          printf("小計 = %d円, 支払金額 = %d円, お釣り = %d円\n",price,pay,balance);
27          printf("----\n");
28
29          /***** 5000円 */
30          amount = balance / 5000;
31          balance = balance % 5000;
32          printf("5000円札 = %d枚\n",amount);
33          price = amount;
34          /***** 2000円 */
35          amount = balance / 2000;
36          balance = balance % 2000;
37          printf("2000円札 = %d枚\n",amount);
38          price = price+amount;
39          /***** 1000円 */
40          amount = balance / 1000;
41          balance = balance % 1000;
42          printf("1000円札 = %d枚\n",amount);
43          price = price + amount;
44          /***** 500円 */
45          amount = balance / 500;
46          balance = balance % 500;
47          printf("500円硬貨 = %d個\n",amount);
48          pay = amount;
49          /***** 100円 */
50          amount = balance / 100;
51          balance = balance % 100;
52          printf("100円硬貨 = %d個\n",amount);
53          pay = pay + amount;
54          /***** 50円 */
55          amount = balance / 50;
56          balance = balance % 50;
57          printf("50円硬貨 = %d個\n",amount);
58          pay = pay + amount;
59          /***** 10円 */
60          amount = balance / 10;
61          balance = balance % 10;
62          printf("10円硬貨 = %d個\n",amount);
63          pay = pay + amount;
64          /***** 5円 */
65          amount = balance / 5;
66          balance = balance % 5;
67          printf("5円硬貨 = %d個\n",amount);
68          pay = pay + amount;
69          /***** 1円 */
70          printf("1円硬貨 = %d個\n",balance);
71          pay = pay + balance;
72          printf("----\n");
73          printf("お札の枚数 : %d枚,硬貨の個数 : %d個\n",price,pay);
74          price = price + pay;
75          printf("お札と硬貨の合計枚数 : %d枚\n",price);
76      }
77
78      return(0);
79  }
```

●出力結果① ※変数"price"、"pay"にはそれぞれ 1234、10000 を入力

```
変数宣言を終了しました
合計の値段はいくらですか? => 1234
いくら払いますか? => 10000
----
小計 = 1234 円, 支払金額 = 10000 円, お釣り = 8766 円
----
5000 円 札 = 1 枚
2000 円 札 = 1 枚
1000 円 札 = 1 枚
500 円 硬貨 = 1 個
100 円 硬貨 = 2 個
50 円 硬貨 = 1 個
10 円 硬貨 = 1 個
5 円 硬貨 = 1 個
1 円 硬貨 = 1 個
----
お札の枚数: 3 枚, 硬貨の個数: 7 個
お札と硬貨の合計枚数: 10 枚
```

●出力結果② ※変数"price"、"pay"にはそれぞれ 1234、999 を入力

```
変数宣言を終了しました
合計の値段はいくらですか? => 1234
いくら払いますか? => 999
----
error!値段が支払いより多いです
```

●解説

- ・ printf 関数によって日本語に対応.
- ・ 変数宣言が終わった後、printf 関数により宣言終了を表示するようにした.
- ・ お札と硬貨のそれぞれの合計枚数を最後に出力するようにした.
- ・ なるべく変数を多く使わないようにし、例題と同じく 4 つの変数を使用した.
- ・ レポートの課題範囲からは外れるが、計算結果にマイナスが現れるのを避けるために、if 文により "price" より "pay" が多い場合にエラーを返すようにした.

iii.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ムを考えること。

●ソースコードの一部

```
9  #include<stdio.h>
10
11  int main(){
12
13      int i ;
14
15      printf("調べたい数を入力==>"); scanf("%d",&i);
16      printf("(Dec):%010d\n",i);
17      printf("(Hex):0x%08x\n",i);
18
19      return(0);
20 }
```

●出力結果 ※ "i"には 734 を入力

```
調べたい数を入力==>734
(Dec):000000734
(Hex):0x00002de
```

●方針

- ・ 10 進数の数字を入力し、10 進数と 16 進数に変換して表示するプログラムを作成した.
- ・ 16 進数を表す場合、「テキスト PP.68 基数 16 の表記法」を参考にし "0x"から始めるようにした.
- ・ 2 進数 32bit の数は、10 進数の場合 10 桁、16 進数の場合 8 桁で表されるので "%10d"、"%08x"により桁数を表し、何もない桁には 0 を詰めるために 0 を入力した.

- ・ 以降の考察では上記のプログラムを用いて、調べたい 10 進数の数字を入力し、10 進数と 16 進数で出力する.
- ・ テキスト P.409 によれば int 型整数の bit 数は 32bit であり、値域は 10 進法で "2147483647" ~ "-2147483648" であるという.
- ・ それぞれの上限値、下限値とその前後の数字を入力して観察する.

a-1:int 型整数の上限値について

●出力結果 ※ "i"には 2147483647 を入力

```
調べたい数を入力==>2147483647
(Dec):2147483647
(Hex):0x7fffffff
```

●考察

- ・ int 型整数の上限値である 2147483647 を入力した場合、10 進数と 16 進数共に問題なく表示された.
- ・ 次に、上限値の前後の数字を入力してみる.

a-2:上限値の前後について

●出力結果① ※ "i"には 2147483648 を入力

```
調べたい数を入力==>2147483648  
(Dec):-2147483648  
(Hex):0x80000000
```

●出力結果② ※ "i"には 2147483646 を入力

```
調べたい数を入力==>2147483646  
(Dec):2147483646  
(Hex):0x7fffffff
```

●考察

- ・ 上限値を超えた数を入力した場合、10進数の出力にはマイナスがついて出力された。(これは int 型整数の下限値である.)
- ・ 上限を超えない数を入力した場合、問題なく出力された。

a-3:int 型整数の下限値について

●出力結果 ※ "i"には-2147483648 を入力

```
調べたい数を入力==>-2147483648  
(Dec):-2147483648  
(Hex):0x80000000
```

●考察

- ・ int 型整数の下限値である-2147483648 を入力した場合、10進数は問題なく表示された。
- ・ 16進数には、2147483648 を入力した場合 (a-2 項②) と同じ結果が現れた。
- ・ 次に、下限値の前後の数字を入力してみる。

a-4:下限値の前後について

●出力結果① ※ "i"には-2147483649 を入力

```
調べたい数を入力==>-2147483649  
(Dec):2147483647  
(Hex):0x7fffffff
```

●出力結果② ※ "i"には-2147483647 を入力

```
調べたい数を入力==>-2147483647  
(Dec):-2147483647  
(Hex):0x80000001
```

●考察

- ・ 下限値を超えた数を入力した場合、a-1 項の 10進数、16進数の出力と同じ結果が現れた。
- ・ 下限を超えない数を入力した場合、問題なく出力された。

a-5:0の前後の値について

●出力結果① ※ "i"には1を入力

```
調べたい数を入力==>1
(Dec):000000001
(Hex):0x0000001
```

●出力結果② ※ "i"には-1を入力

```
調べたい数を入力==>-1
(Dec):-00000001
(Hex):0xffffffff
```

●考察

- ・1を入力した場合、10進数、16進数共に問題なく出力された。
- ・-1を入力した場合、10進数では問題なく出力されたが、16進数では"ffffffff"が出力された。

a-6:int型整数の表現の考察

●考察

- ・16進数の出力(Hex)を観察する
"i=1"の場合で"0x00000001"であり最小、"i=-1"の場合で"0xffffffff"であり最大になっている。
- ・上限値、下限値の場合では"0x7fffffff"、"0x80000000"となり、隣同士の数として出力された。
- ・しかし10進数の出力(Dec)を観察すると、上限値+1を入力した時と下限値、下限値+(-1)を入力した時と上限値の値が等しくなっている。
- ・コンピュータの内部数値は2進数を基本としているので、ある数字を10進数として入力した場合その数を2進数へ変換する。
- ・この時、2進数で符号を表現するために「2の補数」を用いると考えられる。
- ・よって、32bitで2進数の先頭(32桁目)が1になる場合は10進法でマイナスとして扱う。

●それぞれの基数の関係(32bit)

10進数	2進数	16進数
0	0000...0000	0x00000000
1	0000...0001	0x00000001
2	0000...0010	0x00000002
~		
2147483646	0111...1110	0x7fffffff
2147483647(上限)	0111...1111	0x7fffffff
-2147483648(下限)	1000...0000	0x80000000
-2147483647	1000...0001	0x80000001
~		
-2	1111...1110	0xffffffff
-1	1111...1111	0xffffffff

- ・前ページの表より、上限値、下限値を超えた値を扱う場合、2進数へ変換したときにその数は「2の補数表現」として扱うので、出力結果にマイナスが現れたりする。
- ・16進数の出力の変化を見ると、前ページの表は上下の値が繋がりに、1順しているという解釈が分かりやすい。
- ・確認のため、32bitの2進数で表すことのできる最大の数(4294967296)を入力してみる。

●出力結果 ※ "i"には4294967296を入力

```
調べたい数を入力==>4294967296
(Dec):0000000000
(Hex):0x00000000
```

- ・予想通り、32bitの2進数を超えた数を入力すると、オーバーフローを起こしてそれぞれに0が出力された。
- ・以上より、int型整数は32bitの2進数として扱い、それは「2の補数表現」で表すので上限値は2147483647(=2³¹-1)であり、下限値は-2147483648(=-2³¹)となる。

iv.エラーについて考察せよ。

●方針

- ・b-2項について、全角日本語を変数として扱うことができないことがわかった。
- ・ここでは、様々な記号を用いて変数に扱うことができる文字を模索してみる。

1: #, \$, %, & の記号について。

●ソースコード全体

```
1 #include<stdio.h>
2 int main(){
3
4     int #=1,$=2,%=3,&=4;
5     int A;
6
7     A= # + $ + % + &;
8     printf("A=%d\n",A);
9
10    return(0);
11 }
```

●コンパイル時のエラー

```
report@2-5.c:4:7: error: expected identifier or '('
    int #=1,$=2,%=3,&=4;
        ^
report@2-5.c:7:6: error: expected expression
    A= # + $ + % + &;
        ^
```

●考察

- ・ printf 関数内では記号はうまく表示できないので、直接の表現は避けるようにした。
- ・ "#”についての箇所にエラーが表示された。
- ・ ”#”の部分を消去し実行したが、今度は "\$”の箇所でエラーが出力された。
- ・ 以下同様に操作をしたが、すべての記号についてエラーが出力された。
- ・ よって、記号は変数として扱うことができないことがわかった。

—あとがき—

A:参考サイト・文献

- ・ 『ビット(bit)とバイト(Byte) 情報量の単位と 2 進数』
<http://www.biwako.shiga-u.ac.jp/sensei/mnaka/ut/binarydigit.html>
- ・ 『2 の補数を理解する (1)』
<http://d.hatena.ne.jp/simply-k/20100824/1282743815>
- ・ 『C 実践プログラミング 第三版』

B:感想

- ・ 前回のレポートとは違い、四則演算や scanf 関数を扱うことで有能なプログラムコードを書くことができた。ただ作業をするのではなく、考えながらコードを書きそれがうまく実行できたときはすごく嬉しかった。
- ・ int 型整数の扱いについても、様々な本、web ページを参考にしてより理解を深めることができた。