roman_trans ver2.0

【作成者】67140390 平木場 風太
【制作日】2016 年 6 月 29 日 (水曜日)
【連絡先】hm14039@student.miyazaki-u.ac.jp
【動作環境】OS X El Capitan(10.11.5)
【開発環境】gcc 4.2.1
【ダウンロード】`https://github.com/korosuke613/Roman_num_trans`
【インストール】上の URL で zip 形式でファイルをダウンロードし解凍する。Roman_num_trans という実行ファイルがあるので、カレントディレクトリにて./Roman_num_trans をコマンドラインで実行することで使用できる。

1 概要

このソフトウェアはアラビア数字をローマ数字に変換するソフトウェアです。

2 使い方

コマンドラインから実行してください。
パイプまたはリダイレクトによる入力がある場合挙動が異なります。
プログラム名の後に、引数としてオプションを与えると変換の法則が変わります。

2.1 パイプまたはリダイレクトによる入力を与えない場合

プログラム実行時に引数を与えない場合、キーボードから入力された数字を変換して出力します。終了コード'q' が入力されるまで繰り返します。

動作：

1. コマンドライン上で Roman_num_trans を実行する。
2. 画面の支持に従い、変換したいアラビア数字をキーボードから入力する。
3. 変換されたローマ数字が標準出力される。(異常な値が入力された場合、エラーを出力する。)
4. 2,3 を繰り返す。('q' を入力したら終了)

(例)：入力値 17

```
$ ./Roman_num_trans
---------------------------------------
アラビア数字を入力してください (q で終了)
17
XVII
---------------------------------------
アラビア数字を入力してください (q で終了)
q
```

2.2 パイプまたはリダイレクトによる入力を与えない場合

プログラム実行時にパイプまたはリダイレクトによりプログラムに入力を与える場合、与えた数字を 1 つずつ変換し出力します。全ての数字を処理するまで繰り返します。

動作：

1. コマンドライン上で Roman_num_trans を実行する。
2. 外部から与えたアラビア数字を 1 つずつローマ数字に変換し、それぞれ標準出力される。(異常な値が入力された場合、エラーメッセージを出力する。)
3. 終了

(例)：入力値 17,350,1298

```
$ echo 17 350 1298 | ./Roman_num_trans
XVII
CCCL
MCCXCVIII
```

2.3 オプション

-f : ローマ数字の特定の文字を 4 回異常繰り返しては行けないルールを破棄する。
最小値：1、最大値：49999(50000 に当たる文字がないため)
(例)：入力値 4,9,4444

```
$ echo 4 9 4444 | ./Roman_num_trans -f
IIII
VIIII
MMMMCCCCXXXXIIII
```

-g : ローマ数字の特定の文字を 4 回異常繰り返しては行けないルールを破棄する。
また、ローマ数字の $5 * 10^n$ に当たる文字を消去する。
最小値：1、最大値：99999(100000 に当たる文字がないため)
(例)：入力値 5,90,9999

```
$ echo 5 90 9999 | ./Roman_num_trans -g
IIIII
XXXXXXXXXX
MMMMMMMMMCCCCCCCCCXXXXXXXXXIIIIIIIII
```

-v：プログラムのバージョンを表示する。

3 注意事項

- ローマ数字をアラビア数字に変換することはできません。(これから搭載していきたい)
- ローマ数字は一般的な表記法として、1 から 3999 の範囲の数字にしか対応してない [1] のですが、5000 と 10000 に当たる文字 [2] を導入したため、1 から 39999 の範囲となります。
  それ以外の数字を入力した場合、エラーを出力します。
- 数字以外の文字を入力した場合、エラーを出力します。
- C 言語で作られているため、コマンドラインからしか実行できません。

4 更新履歴

ver.2.0

- パイプ・リダイレクトでの入力に対応しました。そのかわり、引数からの数字の入力ができなくなりました。
- ローマ数字の 5000,10000 に当たる文字 [2] を導入しました。
- オプションとして同じ文字を 4 回以上繰り返せるモードと、$5 * 10^n$ に当たる文字を使わないモードを追加しました。

参考文献

[1] ローマ数字は奥が深い・・・3999 までは！
http://plaza.rakuten.co.jp/higashiindo/diary/201207200000/
[2] ローマ数字 - CyberLibrarian
http://www.asahi-net.or.jp/~ax2s-kmtn/ref/roman_num.html

付録 [ソースコード]

ソースコード 1　main.c

```
1  /**
2   * アラビア数字をローマ数字に変換するプログラムです。
3   * roman_trans.c
4   **/
5
6  #include "Roman_num_trans.h"
7
8  /**
9   * メイン関数
10  * main.c
11  **/
12 int main(int argc, char* argv[])
13 {
14     int num, argc_start, max, min, opt;
15     char s[BUFSIZE+1];
16     Roman_arabic* pRoman;
17
18     /* 初期化 */
19     Roman_arabic standard[17]={ //通常
20         {"^^e2^^86^^82␣", 10000},
21         {"M^^e2^^86^^82␣", 9000},
22         {"^^e2^^86^^81", 5000},
23         {"M^^e2^^86^^81", 4000},
24         {"M", 1000},
25         {"CM", 900},
26         {"D", 500},
27         {"CD", 400},
28         {"C", 100},
29         {"XC", 90},
30         {"L", 50},
31         {"XL", 40},
32         {"X", 10},
33         {"IX", 9},
34         {"V", 5},
35         {"IV", 4},
36         {"I", 1}
37     };
38
39     Roman_arabic not_94[9]={ //9,4を抜いた場合
40         {"^^e2^^86^^82␣", 10000},
41         {"^^e2^^86^^81", 5000},
42         {"M", 1000},
43         {"D", 500},
44         {"C", 100},
45         {"L", 50},
46         {"X", 10},
47         {"V", 5},
48         {"I", 1}
49     };
50
51     Roman_arabic not_5[5]={ //5を抜いた場合
52         {"^^e2^^86^^82␣", 10000},
53         {"M", 1000},
54         {"C", 100},
55         {"X", 10},
56         {"I", 1}
57     };
58
59     //オプション
60     opt=getopt(argc,argv,"fgv");
61     switch (opt) {
62         case 'f':
63             // [-f] 9,4を抜く場合の初期化
64             pRoman = not_94;
65             argc_start = 3;
66             max = MAX94;
```

```
67              min = MIN;
68              break;
69
70          case 'g':
71              // [-g] 5を抜く場合の初期化
72              pRoman = not_5;
73              argc_start = 3;
74              max = MAX5;
75              min = MIN;
76              break;
77
78          case 'v':
79              // [-v] プログラムのバージョン出力
80              VERMSG;
81              exit(0);
82              break;
83
84          case '?':
85              // オプションが謎の場合、usage を表示する
86              USAGE(argv[0]);
87              exit(1);
88              break;
89
90          default:
91              // コマンドがない場合の初期化
92              pRoman = standard;
93              argc_start = 2;
94              max = MAX;
95              min = MIN;
96              break;
97
98      }
99
100
101
102     /* メインの処理 */
103     if(!isatty(fileno(stdin))){
104         //標準入力がある場合
105         while(1){
106             fscanf(stdin, "%s", s); //標準入力を文字列 s に代入する
107             if(feof(stdin)) break; //標準入力が終了したらループを抜ける
108             if(rcheck(s, max, min))continue; // 入力値が異常だと、ループを初めからにする
109             num = atoi(s);
110             rtrans_print(num, pRoman);
111         }
112         return 0;
113
114     }else{
115         //標準入力がない場合
116         while(1){
117             printf("---------------\n" );
118             printf("アラビア数字を入力してください (q で終了)\n");
119             scanf("%" XSTR(MAX) "s%*[^\n]%*c", s); //バッファオーバーランを防ぐ
120             if(s[0]=='q')exit(0); //'q'で終了
121             if(rcheck(s, max, min))continue; //入力値が異常だと、ループを初めからにする
122             num = atoi(s);
123             rtrans_print(num, pRoman); //与えた整数をローマ数字に変換し、標準出力する
124         }
125
126         return 0;
127     }
128
129     return 0;
130 }
```

ソースコード 2　roman_common.c

```c
#include "Roman_num_trans.h"

/**
 * ローマ数字に変換できるかチェックして、警告する関数
 * rcheck
 * 返り値 正常:0，異常:1
 * 引数 評価する文字列，最大値，最小値
 **/
int rcheck(char* str, int max, int min){
    int i=0, frag = 0;

    while (*(str+i) != '\0') {
        if(*(str+i) < '0' || *(str+i) > '9')frag = 1;
        i++;
    }
    if(frag) {
        ER_NO_INT; // エラーを標準出力する。
        return 1;
    }
    if(atoi(str) > max || atoi(str) < min){
        ER_REMITES_OVER(min, max); // エラーを標準出力する。
        return 1;
    }
    return 0;
}

/**
 * アラビア数字をローマ数字に変換する手続き
 * rtrans_print
 * 返り値 なし
 * 引数 変換するアラビア数字、変換表
 *_print
 * 返り値 なし
 * 引数 変換するアラビア数字、変換表
 **/
void rtrans_print(int num, Roman_arabic* a){
    int i=0;
    char str[BUFSIZE]="0";

    while(num != 0){
        if(num / a[i].arabic >= 1){
            strcat(str, a[i].roman);
            num -= a[i].arabic;
        }else i++;
    }
    printf("%s\n",str+1);
}
```

ソースコード 3　Roman_num_trans.h

```
1  /**
2   * Roman_num_trans のヘッダファイルです。
3   * Roman_num_trans.h
4   **/
5
6  #ifndef _ROMAN_NUM_TRANS_H_
7  #define _ROMAN_NUM_TRANS_H_
8
9  /* プリプロセッサ */
10 #include <stdio.h>
11 #include <stdlib.h>
12 #include <string.h>
13 #include <unistd.h>
14
15 #define BUFSIZE 128 //文字列の最大値
16 #define MAX 39999 //ローマ数字の最大値
17 #define MAX94 49999
18 #define MAX5 99999
19 #define MIN 1 //ローマ数字の最小値
20
21 //メッセージ関連
22 #define ER_NO_INT printf("ERR:数字以外が入力されました\n");
23 #define ER_REMITES_OVER(min,max) printf("ERR:ローマ数字は %d~%d までの数字しか扱えません\n", min, max);
24 #define VERMSG printf("Roman_num_trans␣ver.2.0\n");
25 #define USAGE(program_name) printf("usage:␣%s␣[-f]␣[-g]\n", program_name);
26
27 //scanf の第一引数にプリプロセッサを適用するための設定
28 #define STR(s) #s
29 #define XSTR(s) STR(s)
30
31
32 /* 構造体宣言 */
33 typedef struct { //ローマ数字とアラビア数字の変換表の構造体
34     char roman[5]; /* ローマ数字 */
35     int arabic; /* アラビア数字 */
36 }Roman_arabic;
37
38
39 /* プロトタイプ宣言 */
40 int rcheck(char* str, int max, int min);
41 void rtrans_print(int num, Roman_arabic* a);
42
43 #endif //_ROMAN_NUM_TRANS_H_
```