

< 注意 >

本ファイルの著作権については，本ファイルの著作者である坂部俊樹，北英彦，山本晋一郎，酒井正彦，外山勝彦に帰属します．本ファイルを利用したことによる直接あるいは間接的な損害に関して，著者およびオーム社はいっさいの責任を負いかねますので，あらかじめご了承ください．

< コメント >

解答例は，あくまでひとつの例であって，妥当な解答は無数にあることに注意しよう．本章で取り上げた演習問題には，それしか解がないという意味の正解は存在しない．学習者ひとりひとりの「もの」の見方，分析の仕方，まとめ方が同じであることはありえない．学習者が異なれば，必ず異なる解答になるはずである．

「プログラムの設計」 3章 演習問題 解答例

問1 絵を言葉で説明するときには説明する人，質問する人が気をつけるべきことは，具体的には非常に多くあるであろうが，ここでは一例を示す．

考察1 絵を言葉で説明するときには気をつけるべきこととして，下記のようなものがある．

- (a) いきなり細かいところから伝えるのではなく，最初は全体のイメージをつかんでもらうようにするように説明の順番を考える．
- (b) 「まず最初に全体のイメージを伝えるので，まだ絵を描き始めないでほしい」と伝える．
- (c) 絵全体の大きさを縦?cm，横?cmのように伝える．
- (d) 絵の中に描かれている物体の数を伝える．
- (e) 絵の中の各物体が何であるかのおおよそイメージを，絵を再現する人に理解してもらえらるだろうと思われる言葉を頭の中で探して伝える．コラムの例題の絵では，糸巻き戦車，風車の部分が花になっている風車の塔．
- (f) 絵の中の各物体が何であるかを伝えるときに，理解してもらえらるだろうと思われるいろんな言葉で言い換えてみる．

- (g) 言葉で何か伝えたときに、絵を再現してもらう人の表情を観察する。理解したような表情をしているか。何をいつているんだろうという表情をしているか。
- (h) 言葉で何か伝えたときに、絵を再現してもらう人の手の動きを観察する。何か理解してメモをとっているかどうか。何か理解して絵を描きはじめているだろうか。
- (i) 各物体の位置を、紙の上のふちから?cm, 下のふちから?cm, 右のふちから?cm, 左のふちから?cm のように伝える。

考察 2 絵を再現する人が質問するときに気をつけるべきこととして、下記のようなものがある。

- (a) いきなり細かいところの説明を求めずに、最初は全体がどうなっているかのイメージをつかむための質問をする。
- (b) いきなり絵を描き始めず、全体のイメージがなんとなくつかめてから絵を描き始める。
- (c) 絵の説明を聞いて自分はこのように理解したが、それでよいかどうかの確認をするための質問をする。例えば、説明者が「風車には普通のドアがついています。」と説明したときに、「どのようなドアですか？縦長の長方形のドアでよいですか？」と質問をする。

問 2 「計算式の中で直前の計算結果を利用できるようにしたい」という要求は、例えば、次のような形で要求仕様書の中の項目としてまとめることができる。

- 直前の計算結果は、変数 A の中に保存される。
- 入力する計算式の中に変数 A を含めることができる。
(例)

$$\begin{aligned} com &> 1+2 \\ &= 3 \\ com &> A+3+A \\ &= 9 \end{aligned}$$
- 直前の計算結果がエラーだった場合には、変数 A には 0 が保存される。

$$\begin{aligned} com &> a+1 \\ ERROR &: illegal input \\ com &> A \\ &= 0 \end{aligned}$$

問3 「割り算のときに、商と余りという形で表示したい」という要求は、例えば、次のような形で要求仕様書の中の項目としてまとめることができる。

- 計算式の中で一番最後に計算される演算が割り算のときにのみ商と余りを表示する。割られる数(被除数)を M 、割る数(除数)を N としたとき、商 Q と余り R は次の式によって定義する。

$$M = Q * N + R$$

ここで、余り R は、 $0 \leq R < |N|$ とする。

(例)

$$com > 18/15$$

$$= 6/5 = 1 + 1/5 = 1.2 = 1...3$$

$$com > -18/15$$

$$= -6/5 = -(1 + 1/5) = -1.2 = -2...12$$

$$com > (3/2) / (2/5)$$

$$15/4 = 3 + 3/4 = 3.75 = 3...3/10$$

- その他の場合には、商と余りという形では表示しない。

問4 有理数電卓に追加する機能のアイデアのみを例としてあげる。要求仕様としては、より具体的に機能を決める必要がある。

- (a) 「過去の計算の計算式と計算結果のリストを表示できるようにする。」
- (b) 「過去の計算結果のどれでも計算式の中に記述できるようにする。」
- (c) 「演算として、剰余演算、累乗演算ができるようにする。」
- (d) 「有理数を分数として表現したときに、分子、分母の大きさに制限を設けず、計算機のメモリが許すかぎり、どのような有理数も扱えるようにする。」

問5 与えられた数の素因数分解を行うプログラムの要求仕様書の例を示す。

【要求仕様書】

システム名 素因数分解

作成者 石田 伸吾

作成日 2000年4月1日

最終更新日 2000 年 4 月 2 日

バージョン 1.2

概要 与えられた数の素因数分解を行うプログラム

動作環境 UNIX

実行方法 :

 コマンド名 prime

 コマンド引数 素因数分解する数 (10 進数)

 実行例 prime 12345

入力 標準入力

出力 標準出力

機能 :

1. 入力は、「0~9」の数字からなる 1 以上で 40 億以下の自然数 (10 進数) を表す文字列とする。(注: unsigned long(符号なし 4 バイト整数) で扱える数値)
2. 入力された数を素因数分解する。
3. 素因数分解した結果を出力する。出力形式は,
「(素因数**その素因数の数) * ... * (素因数**その素因数の数)」
とする。素因数は小さい順に左端から並べる。
(例)
> 4200
= (2 ** 3) * (3 ** 1) * (5 ** 2) * (7 ** 1)

例外処理 :

1. 範囲外の自然数, または, 自然数 (10 進数) 以外の文字列が入力された場合はエラーメッセージ「ERROR:illegal input」を表示する。

問 6 与えられた数の素因数分解を行うプログラムのシステム仕様書, すなわち, モジュール構成, 各モジュールのモジュール仕様書, および, データ辞書の例を示す。

【システム仕様書: モジュール構成】

システム名: 素因数分解

作成者 石田 伸吾

作成日 2000 年 4 月 3 日

最終更新日 2000 年 4 月 4 日

バージョン 1.2

モジュールリスト :

1. メインモジュール

機能 :

- プログラムの起動時に呼ばれる .
- 入力が条件を満たしているかどうか確認をする .
- 下記の 2 つのモジュールの制御を行う .
 - 素因数分解モジュール
 - 出力モジュール

呼び出し方法 `void main(int argc, char *argv[])`

入力 :

- 素因数分解する数 (文字列) : コマンド引数

出力 なし

2. 素因数分解モジュール

機能 :

- 素因数分解を行う .

呼び出し方法 `void prime(unsigned long number)`

入力 • 素因数分解する数 (関数引数)

出力 :

- 素因数分解の結果 `answer` (構造体配列) : グローバル変数
- 素因数の数 `count` (数値) : グローバル変数

3. 出力モジュール

機能 :

- 素因数分解の結果を次の形式で画面に表示する .
(素因数 ** その素因数の数) * ... * (素因数 ^ その素因数の数)

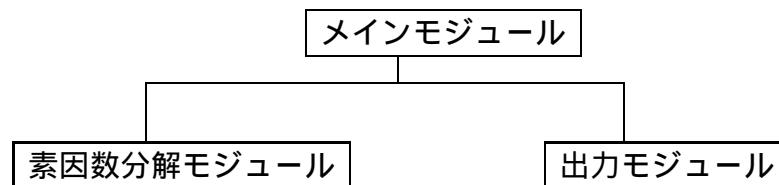
呼び出し方法 `void output(void)`

入力 :

- 素因数分解の結果 `answer` (構造体配列) : グローバル変数
- 素因数の数 `count` (数値) : グローバル変数

出力 素因数分解の結果の画面への表示 (文字列) : 標準出力

モジュール構成図 :



【システム仕様書 : モジュール仕様書】

システム名 素因数分解

モジュール名 メインモジュール

作成者 石田 伸吾

作成日 2000年4月6日

最終更新日 2000年4月7日

バージョン 1.2

概要 プログラムの起動時に呼ばれるモジュールである。

呼び出し方法 `void main(int argc, char *argv[])`

入力：

素因数分解する数(文字列)：コマンド引数

出力：なし

機能：

1. 入力が「0~9」の数字からなる1以上で40億以下の自然数(10進数)を表す文字列であるかどうか確認する。(注：unsigned long(符号なし4バイト整数)で扱える数値)
2. 素因数分解モジュールを呼び出し、与えられた数の素因数分解を行わせる。
3. 出力モジュールを呼び出し、素因数分解モジュールの出力である素因数分解の結果を画面に表示させる。

例外処理：

1. 範囲外の自然数、または、自然数(10進数)以外の文字列が入力された場合はエラーメッセージ「ERROR:illegal input」を表示する。

備考 なし

【システム仕様書：モジュール仕様書】

システム名 素因数分解

モジュール名 素因数分解モジュール

作成者 石田 伸吾

作成日 2000年4月6日

最終更新日 2000年4月7日

バージョン 1.2

概要 与えられた数の素因数分解を行う。

呼び出し方法 `int prime(unsigned long number)`

入力：

素因数分解する数(数値)：関数引数

出力 :

素因数分解の結果 answer(構造体配列) : グローバル変数

素因数の数 count(数値) : グローバル変数

機能 :

1. 与えられた数の素因数分解を行う .
2. 素因数分解の結果の中の異なる素因数の数をグローバル変数 count(数値) へ設定する .
3. 素因数分解の結果をグローバル変数 answer(構造体配列) に設定する .
 - i. 配列の第 n 番目の要素の構造体のメンバー prime には , 第 n 番目に大きい素因数を設定する .
 - ii. 配列の第 n 番目の要素の構造体のメンバー count には , 第 n 番目に大きい素因数の数を設定する .

例外処理 なし

備考 なし

【システム仕様書 : モジュール仕様書】

システム名 素因数分解

モジュール名 出力モジュール

作成者 石田 伸吾

作成日 2000 年 4 月 7 日

最終更新日 2000 年 4 月 8 日

バージョン 1.2

概要 素因数分解の結果を画面に出力する .

呼び出し方法 void output(void)

入力 :

素因数分解の結果 answer(構造体配列) : グローバル変数

素因数の数 count(数値) : グローバル変数

出力 :

素因数分解の結果 (文字列) : 標準出力

機能 :

素因数分解の結果を次の形式で画面に表示する .

(素因数 ** その素因数の数) * ... * (素因数 ^ その素因数の数)

例外処理 : なし

備考 : なし

【システム仕様書：データ辞書】

システム名 素因数分解

作成者 石田 伸吾

作成日 2000年4月7日

最終更新日 2000年4月11日

バージョン 1.4

型 :

素因数分解の結果 素因数分解の結果の各素因数を格納するための型を定義する .

```
typedef struct {
    unsigned long prime ; /* 素因数 */
    int count ; /* その素因数の数 */
} Prime ;
```

グローバル変数 :

素因数分解の結果 answer :

```
Prime answer [ MAX ] ;
```

素因数の数 count :

```
int count ;
```

マクロ :

配列の大きさ :

```
#define MAX 10000
```

コマンド引数 :

素因数分解する数 「0~9」の数字からなる1以上で40億以下の自然数(10進数)を表す文字列 . (注 : (unsigned long(符号なし4バイト整数))で扱える数値)

標準入力 なし

標準出力 :

素因数分解の結果を次の形式で画面に表示する .

(素因数 ** その素因数の数) * ... * (素因数 ^ その素因数の数)

問7 有理数電卓のメインモジュール，入力モジュール，出力モジュールのモジュール仕様書の例を示す．

【システム仕様書：モジュール仕様書】

システム名 有理数電卓

モジュール名 メインモジュール

作成者 山部 俊彦

作成日 2000年2月20日

最終更新日 2000年3月3日

バージョン 1.2

概要 プログラムの起動時に呼ばれるモジュールである．入力モジュール，構文解析/演算実行モジュール，出力モジュールの制御を行う．

呼び出し方法 void main(void)

入力 :なし

出力 :なし

機能：

1. 文字「q」を入力するとプログラムが終了することを画面に表示する．
2. 入力モジュールを呼び出す．
3. 入力モジュールの戻り値が「有理数電卓の終了」(マクロ INPUT_QUIT) ならば，プログラムを終了する．
4. 構文解析/演算実行モジュールを呼び出す．
5. 構文解析/演算実行モジュールから計算結果を戻り値として受け取る．
6. 構文解析/演算実行モジュールの戻り値を調べて構文解析/演算実行中にエラーがあった場合(マクロ CALC_ERROR) には，2へ戻る．
7. 出力モジュールを呼び出す．出力モジュールへ計算結果を引数 answer として引き渡す．
8. 2から繰り返す．

例外処理 :なし

備考 なし

【システム仕様書：モジュール仕様書】

システム名 有理数電卓

モジュール名 入力モジュール

作成者 山部 俊彦

作成日 2000年2月20日

最終更新日 2000年3月3日

バージョン 1.8

概要 ユーザからの入力された文字列を読み込む。

呼び出し方法 `int input(void)`

入力：

ユーザからの入力(文字列)：標準入力

出力：

入力バッファ(文字列)：グローバル変数

画面へのエラーメッセージの表示(文字列)：標準出力

機能：

1. 画面へプロンプト(入力促進記号)「com>」を出力する。
2. ユーザから入力された1行の文字列を入力バッファ(グローバル変数)を読み込む。ただし、空白文字(スペース、タブ)は読みとばす。
3. 81文字以上入力された場合には、エラーメッセージ「ERROR: too long input」を表示して、1へ戻る。
4. 入力された文字列が、演算記号「+」「-」「*」「/」、括弧記号「(」「)」, 数字「0」～「9」のみからなる文字列であることを確認する。
5. 入力が正しくない場合には、エラーメッセージ「ERROR: illegal input」を表示して、1へ戻る。
6. 入力された文字が「q」の場合には、終了コード「マクロ INPUT_QUIT」を設定してモジュールを終了する。
7. 入力された文字列がその他の場合には、終了コード「マクロ INPUT_OK」を設定してモジュールを終了する。

例外処理：

入力が正しくない場合にはエラーメッセージ「ERROR: illegal input」を表示し、さらに、プロンプトを表示して再入力を促す。

1行に81文字以上入力された場合には、エラーメッセージ「ERROR: too long input」を表示する。プロンプトを表示して、再入力を促す。

備考 なし

【システム仕様書：モジュール仕様書】

システム名 有理数電卓

モジュール名 出力モジュール

作成者 山部 俊彦

作成日 2000年2月20日

最終更新日 2000年3月3日

バージョン 1.9

概要 計算結果を画面に出力する。

呼び出し方法 void output(Rational answer)

入力 :

計算結果 answer(有理数) : 関数引数

出力 :

画面への計算結果の表示 (文字列) : 標準出力

画面へのエラーメッセージの表示 (文字列) : 標準出力

機能 :

1. 計算結果の有理数は、仮分数、帯分数、小数点表示の3通りで表示する。
2. 整数になる場合には、整数でのみ表示する。

(例)

```
com > 6/2
= 3
```

3. 小数点表示は、四捨五入して小数点以下2桁までとする。
4. 正の仮分数は、「自然数/自然数」の形式で表す。
5. 正の帯分数は、「自然数+自然数/自然数」の形式で表す。

(例)

```
com > 18/15
= 6/5 = 1 + 1/5 = 1.2
```

6. 負の仮分数は、「-自然数/自然数」の形式で表す。
7. 負の帯分数は、「-(自然数+自然数/自然数)」の形式で表す。

(例)

```
com > 18/15*-1
= -6/5 = -(1 + 1/5) = -1.2
```

8. 帯分数にならない場合には、帯分数としては表示しない。

(例)

```
com > 2/6
= 1/3 = 0.33
```

例外処理 :なし

備考 なし

問8 仕様書のレビューを行ってみて、例えば、次のようなことに気が付くかもしれない。

- (a) メンバーの間のプログラミングに関する能力の差が大きいと、能力の低いメンバーが議論に加わりにくくなる。
- (b) ある程度のプログラミングに関する能力がないと仕様書の中の不明な点、問題点を指摘することができない。

問9 著者が日頃ほしいと思っているソフトウェアのアイデアのみを記す。要求仕様としてはより具体的に機能を決める必要がある。

- (a) 「プログラミングのレポートの課題を学生に出して多数のプログラムが提出されたときに、それらの類似度を計算するプログラム」
- (b) 「プログラミングのレポートの課題を学生に出して多数のプログラムが提出されたときに、それらがコンパイルできること、および、決められたテストデータの集合に対して、プログラムが正しく動作することを確認するプログラム」
- (c) 「プログラミングに関する講義の試験の際に、受験者ひとりひとりに対して異なる試験問題を配付したい。試験問題の中の各内容に対して、同程度の難しさの類似問題を複数用意しておき、それらの中から適当に問題を選んで、受験者ひとりひとりに対して異なる試験問題を作成するプログラム」