cfep のインストール方法および暗号解読用プログラム集

野呂 正行 (理学研究科数学専攻)

November 21, 2013

1 cfep のインストール

1. 授業用ページのリンクから cfep のページに行く.
   ブラウザは Safari を使用すること.
   授業ページは “kobe noro” を Google 検索して, 多分先頭に表示される.
2. 「cfep のアーカイブ」(cfep-intel.zip) をダウンロードする.
3. 自動的に, あるいは cfep-intel.zip をダブルクリックすると, cfep-intel-20100827 というフォルダが現れる.
4. cfep-intel-20100827 をフォルダごとデスクトップに置く.
   デスクトップ直下に置かないと, cfep が正常に動作しない.
5. cfep のページから「Cfep/asir 超入門」もダウンロードする.

2 cfep の使用法

- 起動
  cfep フォルダ中の cfep アイコン (イノブタ) をダブルクリックする.
- 終了
  メニューバーの cfep から 「cfep を終了」を選択する.
- プログラムの保存
  ファイル → 保存
  ファイルを上書きしたくない場合には
  ファイル → 別名で保存

- 保存してあるプログラムの読み込み
  保存したプログラムファイルをダブルクリックする.
- 計算中のまま何も表示されない場合
  止ボタンを押し, 再起動 ボタンを押して復活させる. (超入門 20, 21 ページ参照)

3 入力上の注意

- × は *, 割算 (分数) は /, 余りは % カッコは () のみ
- 計算式の末尾に ; (セミコロン) を必ず書く.
- 文字は「英数」モードで入力すること. (いわゆる全角文字はダメ.)
- 変数は大文字に限る.

4 課題用プログラム集

- レポート問題用紙に印刷されている数字を手で入力すると間違えやすいので, データのみ授業用 web ページに掲載してある. レポートの右肩の番号をクリックするとデータが表示されるので, そこからコピーペーストして使うとよい.
- プログラムは, cfep の入力ウィンドウにどんどん書き足して実行を繰り返せばよい. ただし, 最初の素因数分解 **P1** は時間がかかるので, 一旦正しい P が計算できたら, この部分を実行しないよう, **P2** 以降を新規ファイルで実行すること.

P1：素因数分解

```
N=プリントに与えられた数;
B=isqrt(N);
for ( I=3; I <= B; I = I+2 ) {
  R = N%I;
  if ( R == 0 ) { P = I; print(P); }
}
```

N の因数のうち, $\sqrt{N}$ 以下のものが印字される. P には $\sqrt{N}$ 以下で最大の因数が入っている. もし何分か待っても何も印字されなかったり, たくさん印字される場合には N の入力ミスなので, 再起動ボタンを押して Risa/Asir を再起動すること.

素因数分解をコメント化

```
/*
B=isqrt(N);
for ( I=3; I <= B; I = I+2 ) {
  R = N%I;
  if ( R == 0 ) { P = I; print(P); }
}
*/
```

以下, 別ウィンドウで実行する. N および P1 で得られた P を先頭に書くのを忘れないこと.

P2, P3 : q, n', $kn' \bmod 65537 = 1$ を満たす k の計算 (簡易版)

```
N=プリントに与えられた数;
P=P1 で得た N の因数;
Q=N/P;
N1=(P-1)*(Q-1);
for ( I = 1; I <= 65536; I++ ) {
   R = (I*N1) % 65537;
   if ( R == 1 ) { K = I; print(K); }
}
```

K には $K \cdot N1 \equiv 1 \bmod 65537$ を満たす値が入っている.

P4, P5 : $d = (1 - kn')/65537$ から d を計算

```
D=(1-K*N1)/65537;
D=D%N1;
```

ここで得た D を使って, プリントの暗号ワードを一つずつ復号していく. 次のプログラムを, 各暗号ワードに対して繰り返し実行すれば, 平文を表す数が得られる.

P6: 復号 ($a^d \bmod n$ の計算)

```
A=プリントの暗号化ワード 1 つ;
R=1;
I = A;
for ( B = D; B > 0; ) {
   J = B%2;
   if ( J == 1 ) R = (R*I)%N;
   I = (I*I) % N;
   B = (B-J)/2;
}
print(R);
```

課題の平文は, 1 文字が 3 バイトで表現されている. この平文の 2 文字 (6 バイト) を一つの数として暗号化したものが, 課題の暗号文である. よって, 復号は次の手順による

1. 復号した数字を 6 バイトに分割する

2. 上 3 バイト を文字に変換, 表示

3. 下 3 バイト を文字に変換, 表示

P7: 数を 6 バイトに分割

```
/* R に二文字分入っている．これを，6バイトに分割する */
B6=...; B5=...; ... ; B1 = ...;
```

$B6, \ldots, B1$ は **P6** で得た数 R の下から 1 バイト目, ... 6 バイト目である (この順番に得られる.) これは

$$R = B1 \cdot 256^5 + B2 \cdot 256^4 + \cdots + B5 \cdot 256 + B6$$

と 256 進展開すれば得られる. 各 $B1, \ldots, B6$ は 0 以上 255 以下である.
256 進展開には 256 で割った余りと商を求めることが必要になる. 余りは %, 商は余りと割算 / を使うか, `idiv(A,B)` という関数を使う.

ヒント

```
B6=R%256; R=(R-B6)/256;
```

あとは $(B1, B2, B3)$, $(B4, B5, B6)$ を文字に変換すればよい. $(B1, B2, B3)$ という 3 バイトを 1 文字に変換するには, 組込み関数 `asciitostr` を使う.

3 バイトを 1 文字に変換

```
asciitostr([B1,B2,B3]);
```

まとめて 6 バイト分を 2 文字に変換してもよい.

6 バイトを 2 文字に変換

```
asciitostr([B1,B2,B3,B4,B5,B6]);
```

1 バイト目 ($B1$, $B4$) が 224 以上 239 以下でない場合エラーが起きる. この場合, バイトへの分割が間違っているか, そもそも復号結果が違っているかどちらかなので修正が必要である.