7.11　たかが素因数分解，されど素因数分解

Wikipedia の数の頁には必ずその数の約数と約数の和が書いてあります。数を考えるうえで基本の情報です。しかし約数を求めるためにはその数の素因数[1]がわかる必要があります。今回は素因数を発見するための素因数分解[2]の話です。

昨年の 2024 年 10 月に新たな 2^p-1 の形のメルセンヌ素数 ($p=136279841$) が約 6 年ぶりに発見され最大素数が更新されたと話題になりました。しかし新しいメルセンヌ素数を発見したからといっても，それより小さな素数になれなかったメルセンヌ数が完全に素因数分解できたのではありません。素数の発見と素因数探しは全く別物です。「どうして？ 素因数がみつかったから，素数ではないとわかったんじゃないの？」と思うかもしれません。しかしそうではありません。素数と合成数[3]には明確な性質の違いがあるので，素数チェックプログラムを実行すれば素数なのか合成数なのかがわかるのです。チェックを通過することができた数は疑わしい素数というレベルで「擬素数[4]」といいます。完全な素数というためにはまだいくつかのチェックを受けなければなりません。というのはたくさんある合成数の中には素数のふるまいをする合成数[5]があるからです。このチェックを通過できなかった数は素数ではないと判定され，自動的に合成数ということになります。しかし合成数とわかっても素因数がわかったわけではありません。素数ではないということがわかっただけなのです。

7.11.1　素因数分解の現状

7.11.1.1　$p^n\pm1$ の最大素因数

整数列大辞典の 2^n-1 の最大素因数[6]の頁にある $A005420$ のコメントにこう書かれています。

> The factorization of the composite factor C_{337} of $2^{1207}-1$ with 337 decimal digits is considered by many to be the most desired open factorization problem.

英語の苦手な私が簡単に翻訳しますと

> $2^{1207}-1$ の因数 C_{337} の素因数分解は重要な未解決問題と多くの人に知られています。

$n=1207$ のときのメルセンヌ数 $2^{1207}-1$ は 363 桁の合成数です。その数の因数である 337 桁の合成数の素因数が発見できないためいまだ完全には素因数分解できていません。

$$\text{①}\quad 2^{1207}-1=C_{363}=131071\times228479\times48544121\times212885833\times C_{337}$$

最大素数が更新されたのはいいことなのですが，人類はこのたった 337 桁の合成数の素因数をまだ発見することができないのです。何千万桁の素数発見という誰もが飛びつくニュースはいいのですが，それは途中の山 (素数) を無視して高い山にヘリコプターで降りて「登頂したぞ！」と言っているようなものです。上記の 337 桁の合成数の素因数探しはどなたでも挑戦可能の状態です。

次の表は $p^n\pm1$ の最大素因数が載っている整数列大辞典の現状です。整数列大辞典には具体的な整数列が載せてあります。一般的な数列の数の範囲は，ほとんどが 1 から 100 とか 1 から

---

[1] Prime factor
[2] Prime factorization
[3] Composite number
[4] Pseudoprime
[5] 例としてフェルマーテストという素数チェックを通過する合成数をカーマイケル数といいます
[6] Largest prime factor：その数の素因数の中で最大の素因数

1000 までという 10^n を区切りとして載せてあるのですが，この $p^n \pm 1$ の最大素因数についてはそうではありません。これはその次の数の最大素因数がわからないためです。

式	整数列大辞典	範囲	次の数	式	整数列大辞典	範囲	次の数
2^n-1	$A005420$	-1206	①	2^n+1	$A002587$	-1128	②
3^n-1	$A074477$	$-\ 690$	③	3^n+1	$A074476$	$-\ 691$	④
5^n-1	$A074479$	$-\ 502$	⑤	5^n+1	$A074478$	$-\ 471$	⑥
7^n-1	$A074249$	$-\ 388$	⑦	7^n+1	$A227575$	$-\ 387$	⑧
11^n-1	$A274910$	$-\ 316$	⑨	11^n+1	$A062308$	$-\ 325$	⑩

② $2^{1129}+1=3\times P_{10}\times C_{330}$

③ $3^{691}-1=2\times P_6\times P_7\times P_{22}\times P_{31}\times C_{265}$

④ $3^{692}+1=2\times 41\times P_{10}\times P_{22}\times P_{22}\times C_{277}$

⑤ $5^{503}-1=2^2\times 30181\times P_{12}\times P_{25}\times P_{52}\times C_{260}$

⑥ $5^{472}+1=2\times 17\times 11489\times P_8\times C_{317}$

⑦ $7^{389}-1=2\times 3\times P_{13}\times P_{18}\times C_{299}$

⑧ $7^{388}+1=2\times 1201\times 3881\times P_8\times P_{117}\times P_{198}$

⑨ $11^{317}-1=2\times 5\times P_{11}\times C_{319}$

⑩ $11^{326}+1=2\times 61\times 16301\times C_{334}$

次で紹介する素因数分解サイト `factordb.com` での結果は以上のようになりました。⑧は素因数分解済みでその他はまだ更新できていないようです。

7.11.1.2　他の数の最大素因数

(1) 1 からの連続整数を並べた数

1 からの連続整数を並べた数，例えば 1234567 は 127×9721 になる合成数です。この形の数は現在すべて合成数です。しかしこの数は合成数になるという証明もされていないようです。整数列大辞典の最大素因数の頁[7]には $n=56$ まで登録してあります。`factor.db` で挑戦したところ，$n=124$ まで素因数分解できました。$n=125$ で C_{224} が出現しました。整数列大辞典の登録年が 2003 年なので約 20 年の間にかなり進んだということが分かります。

(2) 各位の和が 2 の数

各位の和が 2 になる素数は現在 2, 11, 101 の 3 個で，これ以降の $10^n+1\,(n\geqq 3)$ の数はすべて合成数です。しかしこの数も $n\geqq 3$ のとき合成数になるという証明はされていないようです。整数列大辞典の最大素因数の頁[8]には $n=331$ まで登録してあります。`factor.db` で $n=332$ を挑戦したところ C_{295} が出現しました。最終更新日は 2022 年でした。

(3) レピュニット (`repunit`)

同じ数字が続く数をレピュニットといいます。式で表すと $\dfrac{10^n-1}{9}$ の形です。この数は素数[9]のときもあり，合成数[10]のときもあります。整数列大辞典の最大素因数の頁[11]には $n=352$ まで登録してあります。`factor.db` で $n=353$ を挑戦したところ C_{328} が出現しました。最終更新日は 2001 年でした。人類の成長は止まったままでした。2 進法でのレピュニットはメルセンヌ数です。

---

[7] https://oeis.org/A075022
[8] https://oeis.org/A003021
[9] https://oeis.org/A004022
[10] https://oeis.org/A199979
[11] https://oeis.org/A003020

7.11.1.3 RSA Factoring Challenge

素因数分解といえばやや昔になりますが，タイトルにあげた話題に触れないわけにはいきません。1991年から2007年まである指定された数の素因数分解に挑戦するイベント「RSA Factoring Challenge」がありました。賞金金額は数によって異なり C_{100} の1000ドルから C_{617} の20万ドルまでありました。賞金が有効な2007年までで，賞金がかけられていた数で素因数分解できた最大数は C_{193}(2万ドル) でした。賞金が出ないのにイベントが終了した後も挑戦している人がいて，2020年には C_{250} まで素因数分解が完了しています。当時出題された問題で C_{260}〜C_{617} までの31個の数がまだ素因数分解できていません。またどのようにしてそれらの数が作られたのかという情報は闇に葬られたままで，わかっているのは2つの素数の積でできているということだけです。その数が作られたPCとハードディスクは壊してしまったようです。[12]

素因数分解の技術はインターネットにおいてクレジットカード情報の暗号化に使われています。また機密文書の送付などにも使われています。昔の話になりますが，米国の数学科の優秀な生徒をその暗号化技術を元にビジネスを行っている会社が引き抜いているという噂を聞いたことがあります。なぜかというと速い素因数分解のソフトまたは理論を考え出されたら困るからです。自分の会社で囲ってしまえば後はなんとかなると考えたのでしょう。

もし現在，素因数分解できなくて困っている合成数に対して簡単で高速な素因数分解ソフトを作ることができたのなら注意が必要です。注意というか命を狙われる可能性があります。それは現在の社会がこの素因数分解を用いた暗号化技術によって成り立っているからです。企業防衛のために必死でその公開をやめさせようとするでしょう。発見した理論やソフトがお金と交換ならまだいいのですが，下手をすると殺し屋が派遣されてくるかもしれません。気をつけてくださいね。このRSA Factoring ChallengeはRSA研究所が提起したものですが，現代の社会がどれくらいまで素因数分解できるかを確認したのではと感じました。まっ純粋な気持ちでの素因数分解挑戦ならいいのですが，このあたりからクレジットカードの有効期限が伸びたような気がしています。終わったことだし気にしないようにします。

ここで書いた情報は残念ながら日本語のWikipediaの項目にはありません。米国の頁[13]にあります。出題された具体的な数も載っています。最小の C_{260} と最大の C_{617} を書いて素因数分解の現状の話題を終わりにしましょう。素因数分解ができたら米国のWikipediaの頁を更新してください。C_{260} は大丈夫だと思いますが最大の C_{617} を更新してしまうと書き込んだIDを元にあなたを探し出して，米国の殺し屋がやってくるかもしれません。(^o^)

$$\begin{aligned}
\text{RSA-260} = C_{260} = {} & 2211282552952966643528108525502623092761208950247001539441374831912882294140200198651272972656974659908590033003140005117074220456085927635795375718595429883895870922923849100670303412462054578456641366454068421436129301769402084639106587591479425143514445819900\\
\text{RSA-2048} = C_{617} = {} & 2519590847565789349402718324004839857142928212620403202777713783604366202070759555626401852588078440691829064124951508218929855914917618450280848912007284499268739280728777673597141834727026189637501497182469116507761337985909570009733045974880842840179742910064245869181719511874612151517265463228221686998754918242243363725908514186546204357679842338718477444792073993423658482382428119816381501067481045166037730605620161967625613384414360383390441495263443219011465754445417842402092461651572335077870774981712577246796292638635637328991215483143816789988504044536402352738195137863656439121201039712282212072035 7
\end{aligned}$$

[12] 表向きかもしれませんが……

[13] https://en.wikipedia.org/wiki/RSA_numbers

7.11.2　たかが素因数分解

(1) 素因数分解サイト

現在では Web 上で素因数分解できるサイトがあります。私が使用しているサイトを紹介します。

順	サイト名	アドレス	特徴
①	factordb.com	https://factordb.com/	素数のデーターベースを自前で保有しているサイトです。調べた数の素因数が登録されていれば瞬時に素因数分解してくれます。素因数が 1 つでも発見できればその数とペアの数が素数か合成数かも判断してくれます。
②	Integer factorization calculator	http://www.alpertron.com.ar/ECM.HTM	自分が探した中では一番の速さの素因数分解サイトです。表示が英語ですが数を貼り付けて Factor をクリックすると素因数分解を始めます。

①のサイトは合成数と判断された数に対して，内部計算する機能があるらしく，しばらくたつと C と表示された合成数が F と表示され素因数がみつかった時がありました。またデータベースに登録されていない新発見の素数は赤字で表示されます。

(2) 素因数分解プログラム

①では合成数という結果で, ②ではいくら待っても計算が終わらない数の場合には御自身の PC で素因数分解することをおすすめします。私が使用している素因数分解のソフトは YAFU[14]というソフトでダウンロード先は https://sourceforge.net/projects/yafu/ です。[15] 一般的な市販ソフトと異なりインストーラはなく，zip 形式で圧縮された形ですのでそのまますべてのファイルを新しいフォルダーに展開します。展開した後に

```
yafu-x64.exe "factor(123456789101112131415161 7)"
                                  素因数分解したい数
pause
```

上のような内容で"test.bat"という実行ファイルを展開したフォルダーに作っておくと便利です。素因数分解が終わる前に OS の再起動や，誤ってプログラムを閉じてしまった場合には 1 行目の最後に"-R"と書いて実行すると途中から再実行してくれます。実行結果はディスプレイに表示されますが，同じフォルダー内の"factor.log"というファイルにどのメソッドで素因数がみつかったのか，実行時間，素因数および素因数の桁等の情報が記録されています。以下そのときの結果表示です。ASUSI7 は私の PC の機種名です。

```
05/11/25 07:30:12 v1.34.5 @ ASUSI7, ****************************
05/11/25 07:30:12 v1.34.5 @ ASUSI7, Starting factorization of 123456789101121314151617
05/11/25 07:30:12 v1.34.5 @ ASUSI7, using pretesting plan: normal
05/11/25 07:30:12 v1.34.5 @ ASUSI7, no tune info: using qs/gnfs crossover of 95 digits
05/11/25 07:30:12 v1.34.5 @ ASUSI7, ****************************
05/11/25 07:30:12 v1.34.5 @ ASUSI7, rho: x^2 + 3, starting 1000 iterations on C24
05/11/25 07:30:12 v1.34.5 @ ASUSI7, rho: x^2 + 2, starting 1000 iterations on C24
05/11/25 07:30:12 v1.34.5 @ ASUSI7, rho: x^2 + 1, starting 1000 iterations on C24
05/11/25 07:30:12 v1.34.5 @ ASUSI7, prp8 = 22999343
05/11/25 07:30:12 v1.34.5 @ ASUSI7, prp16 = 5367839816168719
05/11/25 07:30:12 v1.34.5 @ ASUSI7, Total factoring time = 0.0000 seconds
```

[14]Yet Another Factorization Utility
[15]2025年 5 月現在 Version は 1.34

$$123456789101121314151617 = C_{25} = P_8 \times P_{16}$$
$$= 22999343 \times 5367839816168719$$

【他の実行例】

$$C_{90} = 940833678094073784654106687953925186996784$$
$$6586303009001296529561643865447411259378240947 29$$
$$= P_{32} \times P_{59}$$
$$= 364505682067706087911780963857 83 \times$$
$$25811221179243952186920238827413131290368483933428434308863$$

上記のC_{90}は約51秒でできました。これが②のInteger factorization calculatorでは40分かかりました。10年ほど前，同サイトを32bit-PCで実行したときには2時間21分かかりました。当時とは異なる環境の64bit-PCに変わって約3.5倍速くなりました。このようにハードの進歩も素因数分解にとっては大切な要素です。昨年YAFUを使ってC_{164}を半年かかって素因数分解しました。素因数分解は基本有限時間内でできるのですがその時間が問題なのです。それとPCを動かす電気代も。桁数が多くなると指数関数的に実行時間が増加していくからです。

私が感じる現在の素因数分解の一般的な限界は150〜170桁位だと感じています。もちろんこれより大きな桁の数でもすぐに素因数分解ができる数はあります。しかし数によっては素因数がみつからない数があるということです。ようするに現在の人類は150〜170桁の数に含まれている素因数，おおよそ80桁位の素数が発見できないのです。

たまに話題になる国産スーパーコンピュータ「富岳」を使用できればもっと速く素因数分解できるのでしょうが，単体マシンではないため新たなソフトウェアが必要になってきます。何千万桁の素数発見と騒いでいる世の中ですが，現状はこんな状態です。中飛ばしの必要な素数をきちんと発見しなければ人類の進歩はありません。

7.11.3　されど素因数分解

私のサイトに多倍長計算用BASIC通称UBASIC86上で動作する素因数分解ソフトFACT2025.UBを公開しています。これは私が発見した素数をデータベースとして素因数を発見するソフトです。ソフト自力で素因数を発見できるルーティン[16]は入っていますが，UBASIC86は32bit上で動作するため64bitで動作するソフトと比較すると実行スピードで全く立ち向かうことができないため，発見できなかった合成数の素因数を自前で探していいかどうか確認のメッセージを出すようになっています。プログラムの特徴は32bitなのでレベルは劣っていますが考え方はfactordb.comのサイトと大体同じです。素因数分解によって新しい素数を発見すると自前のデータベースにその素数を登録するので，使えば使うほど賢くなっていきます。ある数または式を入力すると以下の順で素因数を探していきます。

① 内部組み込み関数から素因数を探す。
② 自前の素数のデータベース (Prime1.TBL, Prime2.TBL) から素因数を探す。
③ 方法の選択 (素因数の入力，素因数の内部計算，再実行) という流れで素因数分解を行います。再実行は桁が大きい数の入力間違いを防ぐために，コマンド待ちの間にデータベースを自分で更新して実行できるようにしたものです。

素数のデータベースは2つに分かれていて，最後の③に到達するまでにかなりの時間がかかります。もっと高速にできるアイデアはありますが，プログラムとファイル形式が煩雑になるので，今後の課題とします。

[16]数によって異なりますが30桁位なら待ってられるレベル，作成者は木田祐司さん

(1) `Prime.TBL`

File size 494 K Byte，これは (m,k)-完全数[17]を求めるために私が発見した素数で，約 2 万個の素数のデータベースです。この素数はほとんどが連続約数の和を求める段階で出現した合成数の素因数です。現在の `factordb.com` に登録されていない素数も数多くあります。

(2) `Prime2.TBL`

このファイルは圧縮分割してあります。解凍時 File size 27.9 M Byte，約 52 万個の素数のデータベースです。整数列大辞典や今回紹介しなかったいろいろな数をテーマにした素因数分解サイトから集めてきた素数です。素数が載っている頁をみつけたら数を抜き出して素数チェックをして保存するプログラムを作った覚えがあります。

ここで，「いちいち素因数分解するよりもどんどん素数を発見していけばいいじゃないの？」と感じた方はいませんか？ 発想はいいのですが，それは無謀です。数は無限にあります。素数を発見するのは意外と簡単なのですが，発見できた素数を保存しておくハードディスクの容量はすぐに足りなくなります。素数のおよその個数は「素数定理」から求めることができますが，10^{29}(28 桁) までの実際の個数が整数列大辞典 $A006880$[18]にありました。次の表は素数の個数と素数の文字数を掛け合わせて，単純に 1 文字 1 byte で計算し，おおよその素数を保存する領域の大きさを示した表です。

① 桁数	② 素数の個数	③ その桁の素数の数	① × ③	累計 (byte)	キロ K byte	メガ M byte	ギガ G byte
1	4	4	4	4	–	–	–
2	25	21	42	46	–	–	–
3	168	143	429	471	–	–	–
4	1229	1061	4244	4673	4.6	–	–
5	9592	8363	41815	46059	45.0	–	–
6	78498	68906	413436	455251	444.6	–	–
7	664579	586081	4102567	4516003	4410.2	4.3	–
8	5761455	5096876	40775008	44877575	43825.8	42.8	–
9	50847534	45086079	405774711	446549719	–	425.9	–
10	455052511	404204977	4042049770	4447824481	–	4241.8	4.1
11	4118054813	3663002302	40293025322	44335075092	–	42281.2	41.3

実際には区切り記号も必要ですがそれは無視して, 8 桁までの素数を保存するのに約 43 M byte, 11 桁までの素数を保存するのに約 41 G byte 必要です。およそ 3 桁増えるごとに 10^3 倍になることがわかればいいと思い，表には載せませんでしたが 14 桁までの素数を保存するのに約 40 テラ T byte，17 桁までの素数を保存するのに約 39 ペタ P byte 必要です。ファイルの圧縮や 2 桁を 1 byte で表す等の手段を使ってもすぐに破綻します。無謀という意味がわかりましたか？

単純に素数といってもいろいろな性質をもつ素数があります。現在の研究対象の数に対して必要な素数はたくさんある素数の中のほんの一握りの素数なので，その素数を探し出す素因数分解プログラムはどうしても必要です。

私が研究している (m,k)-完全数を求めるために発見した素数は約 2 万個 (494 Kbyte) と書きました。6 桁未満の素数はプログラムの内部関数が発見してくれるため登録してありません。しかし素数の総数に比べて明らかに少ないことがわかります。これは探している素数が約数の和からできる数の素因数という性質のためです。約数の和という性質からすべて素数の累乗和からできる数の素因数を求めているからです。また対象としている n (1〜10000) が数の世界から見ると小さいことにも起因しています。

[17]整数nの連続約数の和が何回 (m) で自身の整数倍 (k) になるかを表した数, Wikipedia の「超完全数」の頁を参照
[18]David Baugh という方が 2020年に 10^{28} まで，2022年に 10^{29} まで数えたようです。すごい！

7.11.4 UBASIC86

私が数値計算に使用している多倍長計算用 BASIC 通称「UBASIC86」を紹介しておきます。これは立教大学理学部教授木田祐司[19]さんが開発した BASIC インタプリタです。ダウンロードは https://www.vector.co.jp/soft/win95/prog/se089826.html で DOS/V 版が可能です。ただし 32bit-PC で動作します。現在主流の 64bit-PC 上で動かすには他にエミュレータといわれる 64bit-PC を仮想 32bit-PC にするソフトを別にダウンロードする必要があります。これは「DOSBox-X」を推薦します。ダウンロード先は https://github.com/joncampbell123/dosbox-x/releases です。

UBASIC86 の主な特徴は，整数は 2600 桁まで，有理数は分子，分母あわせて 2600 桁まで，実数は整数部，小数部あわせて 2600 桁まで，複素数も実部，虚部あわせて 2600 桁まで計算可能[20]という特徴があります。作者の木田祐司さんのコメントを載せます。

> BASIC をおぼえてしばらくして計算機の様子が少し分かって来ると正確に計算できる数の範囲が驚くほど狭いことに気がつきこれでは何もできないと頭を抱えたことが誰でも一度はあるでしょう。環境に恵まれた人はあるいは数式処理言語を用いるという解決法を見い出したかも知れません。しかしそれでも何故こんな大げさで面倒なことをしなければならないのかと疑問に思うこともあったでしょう。数値計算だけで良いからコンパクトで高速な言語が望まれるわけです。
>
> UBASIC86 はそういう要求から作られた言語です。形式は誰でも使えるように通常の BASIC に準じています。BASIC は専門家の指摘する通り悪い癖の付きやすい問題のある言語ですが覚え易くしかも一度覚えると忘れない (忘れるほどの内容がない？) ことは非専門家には何よりの利点です。また我々のように短いプログラムしか作らないものにとってはその欠点もあまり出てきません。もちろん言語を作り易いというこちらの都合もあります。そのため普通の BASIC を少しかじったことのある人はすぐに使うことが出来ます。スピードも (機能の少なさゆえ) 普通のミニコンで REDUCE[21]を使うよりも優れているようです。実用性も素因数分解関係のものを十分高速に作ることにより証明出来ました。

開発終了からすでに 30 年近く経ったソフト[22]ですが，今なおこれに変わるソフトがないくらいの信頼ある数値計算専用のソフトです。主言語に BASIC を用いていることもプログラミング初心者に対応しており，通常の BASIC 言語ではできない再帰的な定義も可能になっています。他言語のいいとこどりの BASIC です。

UBASIC86 で作成したプログラムは a + f・6 (ファイル名) で保存するといいと思います。Windows 上のエディターでプログラムの改変ができるからです。また実行中のプログラムの強制終了は CTRL + c です。

使い方はマニュアル本[23]が出版されていますが, 作者の木田祐司さんが多倍長計算用 UBASIC86 のための実用プログラムを複数公開[24]されています。ある程度プログラムに触れたことのある方ならプログラムのソースから勉強した方が上達は速いかもしれません。BASIC が対話型言語という性質からダイレクトモードで桁数にとらわれない計算が可能です。ご自身の PC に動作するようにセッティングしておいても損はないと思います。

---

[19] 主な著作「コンピュータ整数論」日本評論社 1994年,「初等整数論」朝倉書店 2001年等
[20] 「コンピュータ整数論」木田祐司・牧野潔夫より
[21] オープンソースである数式処理システムのひとつ
[22] 最終更新日は 1998年 4 月
[23] 「UBASIC86 第 8.7 版ユーザーズマニュアル」日本評論社 1994年
[24] https://www.vector.co.jp/vpack/browse/person/an000710.html

7.11.5　DOSBox-X の設定

OS を Windows とする 64bit-PC 上で UBASIC86 を動かすために必要な，DOSBox-X[25]の設定について説明します。

C ドライブの最上位フォルダー C:\DOSBox-X に DOSBox-X がインストールされ，その下位フォルダー C:\DOSBox-X\UBASIC に UBASIC86 がインストール (展開) されているものとします。

(1) DOSBox-X (ファイル名：dosbox-x.exe) を実行する。(ショートカットをディスクトップに貼り付けておくといいでしょう。)

(2) メニューバーの「メイン」→「設定ツール」を選択する。

(3) 表示された DOSBox-X 設定ツールの中の"AUTOEXEC.BAT"を選択する。

(4) テキストウィンドウが開くので以下の文を入力する。(中身は昔の DOS コマンドです。)

```
mount C C:\DOSBox-X\UBASIC ↵
C: ↵
CD C:\DOSBox-X\UBASIC ↵
UBH.BAT ↵
```

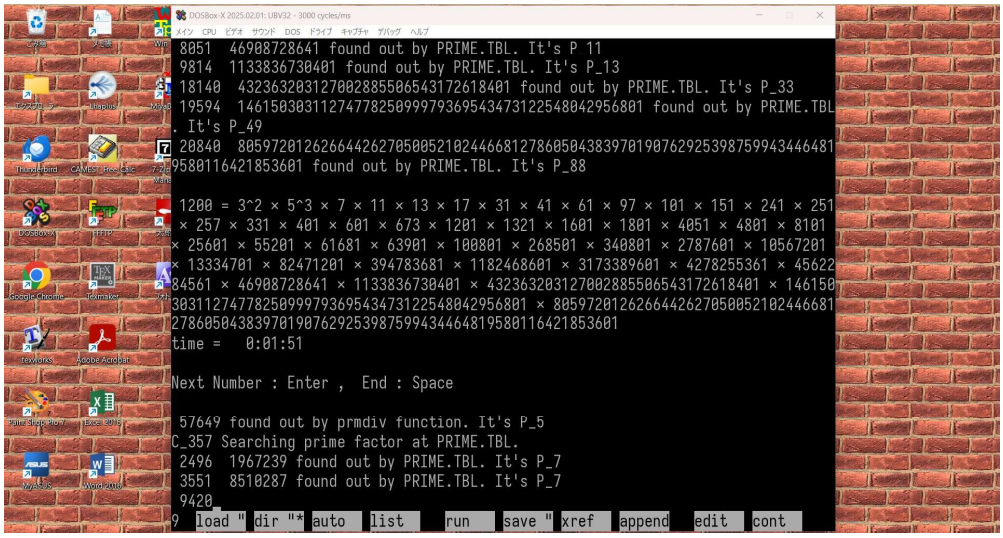
(5) [OK] をクリックし保存して再実行する。

(6) UBASIC86 が起動するので [f・1] ↵ から実行したいプログラムを選択する。

(7) UBASIC86 を終了したいときは"system↵ "と入力します。UBASIC86 が終了して，DOSBox-X 上で動作する 32bit-PC の DOS コマンド待ちになります。さらに"exit↵ "と入力すると DOSBox-X が終了します。

現在の段階では DOSBox-X についてはこれくらいしかまだ知りません。今後 UBASIC86 を動作するにあたってわかったことがあったら書いていきます。

最後に，今年の 3 月 factordb.com で素因数がみつからなかった C_{339} を 4 月に再度実行したところ $C_{339} = P_{55} \times C_{284}$ と異なる結果を出力しました。こんな具合にわずかですが素因数分解は進歩しています。



Windows11 上の DOSBox-X で UBASIC86 を実行している様子

[25]2025年 5 月，Version 2025.02.01