HappyCalc の手引き

2012-11-15

和田 英一 (IIJ 技術研究所)
eiiti.wada@nifty.com

Happy Hacking Calculator

Happy Hacking Calculator (HappyCalc) は, 逆ポーランド記法 (Reverse Polish Notation (RPN)) を採用した整数計算用の電卓で, iPod touch と iPhone 用に実装した. 式の処理に必要なスタックの要素は 2 の補数の 64 ビット. スタックの深さは 16 段で, 16 回を超えてプッシュすると, スタックの底は壊れる. 一方, スタックの底を超えてポップすると, 下から 0 が無限に現れる.

HappyCalc には '入力中 (Input)' と '演算後 (Ready)' の 2 つの状態をとる基本状態 (basic state) と, '十六進 (Hex)', '十進 (Dec)', '八進 (Oct)' の 3 つの状態をとる基数状態 (radix state) とがある.

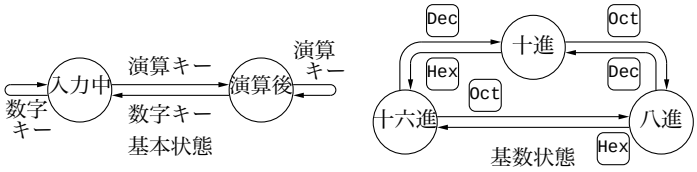

図 0 HappyCalc の状態遷移

HappyCalc のインターフェースは, 左半分に 0, 1, ···, F の数字キーと, 右半分に /, *, ···, JD の演算キーがそれぞれ 16 個あり, 数字キーの上に演算結果を示す窓がある.

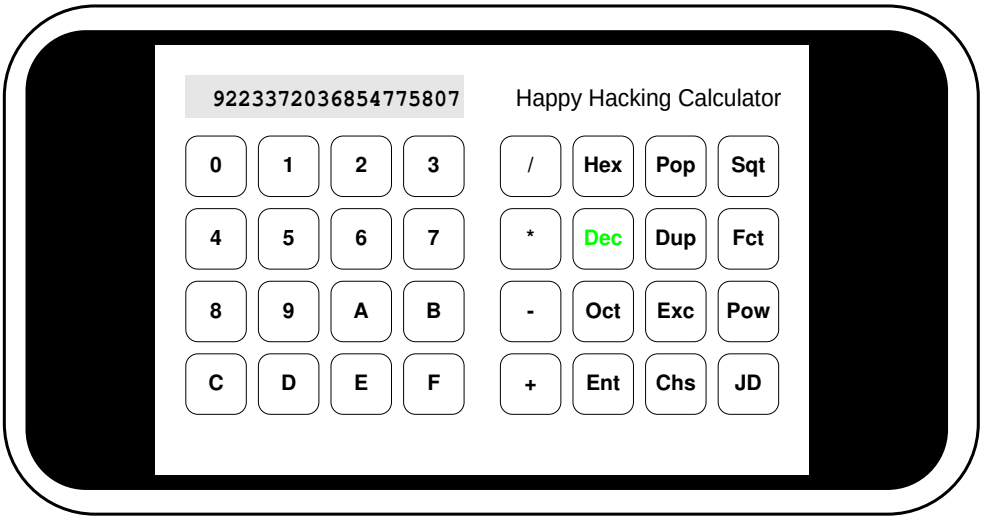

図 1 HappyCalc のインターフェース

基数状態は, 外部の基数進法と内部の二進法との変換に使われる. 基数切替えの演算キーは Hex, Dec, Oct で, それぞれ基数 (radix) を 16, 10, 8 に設定する. 基数状態は *ir* に記憶する.

基数の現在の状態は, 基数切替えのいずれかのキーの文字の緑で示す. 図 1 では Dec が緑になっている. 起動時の基数状態は '十進' に設定する.

‘演算後’ 状態で, どれかの数字キーが押されると, 基本状態は ‘入力中’ になり, スタックを 1 段押し下げ, 次の演算キーが押されるまでの数字キーの列が新しいスタックトップに入る. 基数状態に関係なく, 数字キー 0 から 9 は整数 0 から 9 を入力し, A から F は 10 から 15 を入力する.

演算キーは, 基本状態がいずれにあっても, その時点のスタックトップの被演算子について演算を行い, 演算結果をスタックトップに置き, 窓に表示する. 基本状態を ‘演算後’ に設定する.

以下では, $s[ix]$ はスタックトップの要素を示す; $s[ix-1]$ はトップのすぐ下の要素を示す. ix はスタックトップの添字, ir は基数のレジスタである. (ix は 16 を法として増減される.)

注意: HappyCalc にはバックスペースキーがないので, 数値入力を間違えたとき, [Pop] キーを押し, スタックトップの数値を捨てる.

注意: 殆んどの演算は瞬時に終るが, 2^{45} を超える素数の素因数分解にはかなりの時間がかかる. 素数 $2^{47}-115$ で 7 秒程度かかる. 演算実行中は [Fct] が青になっているので分かる.

演算

[+] 加算. $s[ix-1] \leftarrow s[ix-1]+s[ix]$, $s[ix] \leftarrow 0$, $ix \leftarrow ix-1$.

[−] 減算. $s[ix-1] \leftarrow s[ix-1]-s[ix]$, $s[ix] \leftarrow 0$, $ix \leftarrow ix-1$.

[*] 乗算. $s[ix-1] \leftarrow s[ix-1] \times s[ix]$, $s[ix] \leftarrow 0$, $ix \leftarrow ix-1$.

[/] 除算. $n \leftarrow s[ix-1]$, $d \leftarrow s[ix]$. $d \neq 0$ なら, $r \leftarrow n\%d$, $q \leftarrow (n-r)/d$, ($0 < d$ の時, $0 \leq r < d$; $d < 0$ の時, $d \leq r < 0$.) $s[ix] \leftarrow q$, $s[ix-1] \leftarrow r$. $d = 0$ なら, エラー *.

[Hex] 十六進. $ir \leftarrow 16$.

[Dec] 十進. $ir \leftarrow 10$.

[Oct] 八進. $ir \leftarrow 8$.

[Ent] 何もしない.

[Pop] 削除. $s[ix] \leftarrow 0$, $ix \leftarrow ix-1$.

[Dup] 複製. $s[ix+1] \leftarrow s[ix]$, $ix \leftarrow ix+1$.

[Exc] 交換. $t \leftarrow s[ix]$, $s[ix] \leftarrow s[ix-1]$, $s[ix-1] \leftarrow t$.

[Chs] 符号反転. $s[ix] \leftarrow -s[ix]$.

[Sqt] 開平. $n \leftarrow s[ix]$. $n \geq 0$ なら, $q \leftarrow \lfloor\sqrt{n}\rfloor$, $r \leftarrow n-q \times q$, $s[ix+1] \leftarrow q$, $s[ix] \leftarrow r$, $ix \leftarrow ix+1$. $n < 0$ なら, エラー *.

[Fct] 素因数分解. $n \leftarrow s[ix]$. $n \geq 2$ なら, $r \leftarrow n/p^k$, $k \geq 1$ (ただし $p \geq 2$ は n の最小素因数; r は p を素因数に持たない.) $s[ix+2] \leftarrow p$, $s[ix+1] \leftarrow k$, $s[ix] \leftarrow r$, $ix \leftarrow ix+2$. $n < 2$ なら, エラー *.

[Pow] 冪乗. $b \leftarrow s[ix-1]$, $p \leftarrow s[ix]$. $p > 0$ または ($p = 0$ かつ $b \neq 0$) なら, $s[ix-1] \leftarrow b^p$, $s[ix] \leftarrow 0$, $ix \leftarrow ix-1$. $p < 0$ または ($p = 0$ かつ $b = 0$) なら, エラー *.

[JD] ユリウス日. $s[ix] \leftarrow y$ 年 m 月 d 日世界時正午のユリウス日数. y, m, d は $s[ix]$ に $y \times 10000 + m \times 100 + d$ で表わす.

* エラーの場合, 演算キーの名前が赤で示され, スタックは変らない.

使い方の例

スタック

入力とスタックの関係を次の図 2 に示す. (a) 111 と入力. スタックには 111 が積まれる. (b) 数値に続けて次の数値を入れるために [Ent] を押す. (c) 次の数値が入れられるようになったので, 222 を入れる. 111 の上に

222 が積まれる．(d) 再び[Ent]を押し, 次の数値を入れられるようにする．(e) 333 を入れる．222 の上に 333 が積まれる．(f) [+]を押す．スタックの上 2 段, 222 に 333 が足されて 555 になる．(g) [+]を押す．111 に 555 が足されて 666 になる．(h) 777 を入力．666 の上に 777 が積まれる．(i) [Exc]を押す．スタックの最上段の 2 段の内容が入れ替わり, 777 の上に 666 が積まれる．(j) [−]を押す．引き算の結果, 111 がスタックに残る．(k) [Chs]を押すと, 最上段の数値の符号が反転し, −111 になる．(l) [Dup]により最上段の内容がその上にコピーされる．(m) 次に[*]を押すと, スタックの上 2 段の乗算を実行し, 12321 がスタックに残る．(n) 110 を入力すると, 12321 の上に 110 が積まれる．(o) [/]を押す．12321/110 が実行され, 商 112, 剰余 1 がスタックに得られる．(p) 見えているのは商の方なので, [Pop]を押し, 剰余 1 を得る．この一連の演算は, 基数が十六進でも十進でも八進でも同じになる．

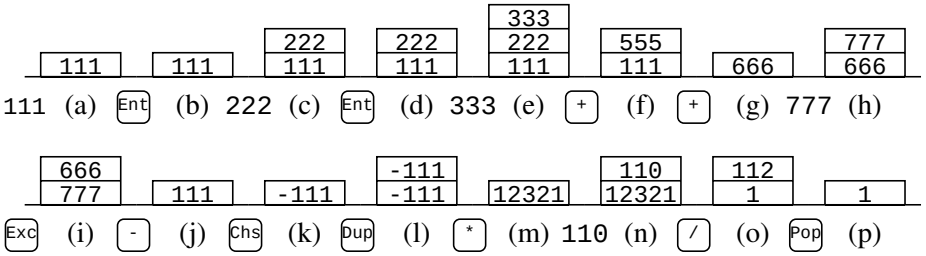

図 2 スタックの遷移 (演算)

基数切替え

2 [Ent] 10 [Pow] 1 [−] (→ 1023) [Hex] (→ 3FF) [Oct] (→ 1777). 数値入力の途中では基数は変えられない.

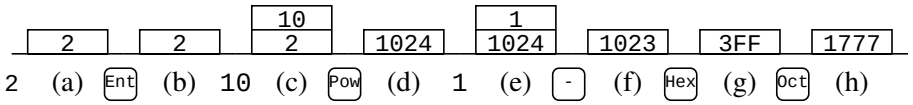

図 3 スタックの遷移 (基数)

除算の商と剰余

剰余の符号は除数の符号となるので, 特に負数で割るときに注意.

100 [Ent] 8 [/] (→ 商 12 剰余 4),
100 [Chs] 8 [/] (→ 商 −13 剰余 4),
100 [Ent] 8 [Chs] [/] (→ 商 −13 剰余 −4),
100 [Chs] 8 [Chs] [/] (→ 商 12 剰余 −4),

200 [Ent] 8 [/] (→ 商 25 剰余 0),
200 [Chs] 8 [/] (→ 商 −25 剰余 0),
200 [Ent] 8 [Chs] [/] (→ 商 −26 剰余 −8),
200 [Chs] 8 [Chs] [/] (→ 商 24 剰余 −8).

$2^{63}-1$ の平方根

[Hex] 7FFFFFFFFFFFFFFF (F を 15 個) [Sqt] (→ B504F333, 剰余 1615E23D6) [Dup] [*] (→ 7FFFFFFE9EA1DC29) [+] (→ 7FFFFFFFFFFFFFFF, 元の数).

2 の平方根

[Dec] 20000000000000000(0 を 16 個) [Sqt] (→ 141421356 $= \lfloor\sqrt{2}\times 10^8\rfloor$) [Dup] [*] (→ 19999999932878736) [+] (→ 20000000000000000).

$2^{63}-1$ の素因数分解

$2^{63}-1 = 7\times 7\times 73\times 127\times 337\times 92737\times 649657$.

[Dec] 2 [Ent] 63 [Pow] 1 [−] (→ 9223372036854775807) ([Fct] [Pop] [Pop])^6 (→ 7, 2, 188232082384791343, 73, 1, 2578521676503991, 127, 1, 20303320287433, 337, 1, 60247241209, 92737, 1, 649657, 649657, 1, 1).

20! の素因数分解

Dec 2 Ent 3 * 4 * ... * 9 * A * B * ... * F * 16 * 17 * 18 * 19 * 1A * (→ 2432902008176640000 = 20!).
$20! = 2^1\times3^1\times2^2\times5^1\times(2^1\times3^1)\times7^1\times2^3\times3^2\times(2^1\times5^1)\times11^1\times(2^2\times3^1)\times13^1\times(2^1\times7^1)\times(3^1\times5^1)\times2^4\times17^1\times(2^1\times3^2)\times19^1\times(2^2\times5^1) = 2^{18}\times3^8\times5^4\times7^2\times11^1\times13^1\times17^1\times19^1$ ここで, Fct (→ 2 = 最小の素因数) Pop (→ 18 = 指数) Pop (→ 9280784638125) Fct (→ 3 = 最小の素因数) Pop (→ 8 = 指数) Pop ... .

ユリウス日

2008 年 10 月 20 日のユリウス日:
20081020 JD (→ 2454760)1 + 7 / Pop (→ 1 = 月曜. (JD + 1)%7 で曜日が得られる).
−4712 年 1 月 1 日のユリウス日:
4712 Chs 10000 * 101 + JD (→ 0. 参考: −47119899 Ent 10000 / (→ 商 −4712 剰余 101)).

沙翁の生涯

William Shakespeare は 1564 年 4 月 23 日生まれ, 1616 年 4 月 23 日没. 何日生きていたか.
16160424 JD 15640423 JD − (→ 18984 日).

ゾロ目の計算

Dec 12345679012345679 (8 はとばす) Ent 81 * (→ 999999999999999999) Sqt (→ 999999999)
Hex 123456789ABCDF (E はとばす) Ent E1 * (→ FFFFFFFFFFFFFFF)
Oct 12345701234570123457 (6 はとばす) Ent 61 * (→ 777777777777777777777)
1...7 が 2 回だと Oct 1234570123457 Ent 61 * (→ 77777777777777 7 が偶数個) Sqt (→ 7777777).

2038 年問題

UNIX time の繰上がり時刻の計算 (UTC 1970 年 1 月 1 日 0 時 (正子) から 2^{31} 秒後の時刻)
2 Ent 31 Pow (→ 2147483648) 86400 / (→ 24855 日後) 365 / (→ 68 年後=2038 年) 4 / (→ 17 回閏年) Exc Pop (→ 17, 1 段下の 35 からこの閏年を引く) − (→ 18. (1 月) 18 日一杯で 24855 日になる. 以下 1 月 19 日の時刻を計算) Pop (→ 11648 秒) 3600 / (→ 3 時) Pop (→ 848) 60 / (→ 14 分) Pop (→ 8 秒. 従って 2038 年 1 月 19 日 3 時 14 分 8 秒.)

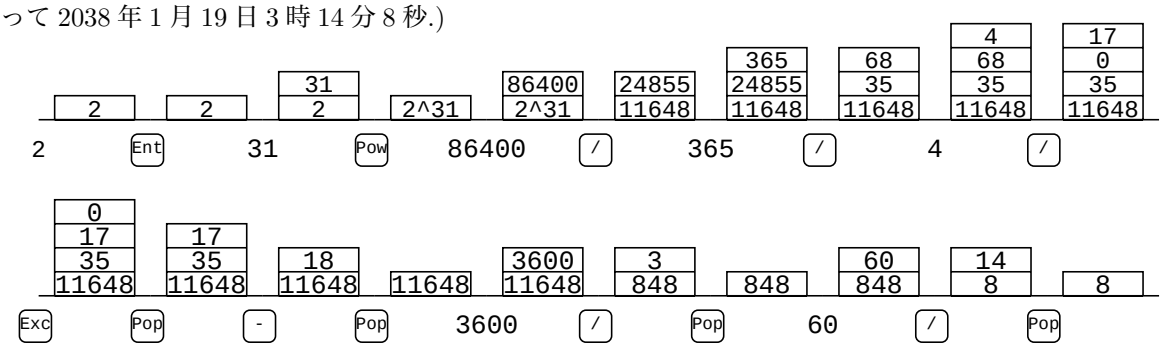

図 4 Unix time の計算

円周率の連分数展開

3141592654 Ent 10 Ent 9 Pow / (→ 3) Pop (→141592654) 10 Ent 18 Pow Exc / 10 Ent 9 Pow / (→ 7) Pop (→ 62513285) 10 Ent 18 Pow Exc / 10 Ent 9 Pow / (→ 15) Pop (→ 99699762) 10 Ent 18 Pow Exc / 10 Ent 9 Pow / (→ 1) Pop (→ 3411839) 10 Ent 18 Pow Exc / 10 Ent 9 Pow / (→ 293) Pop (→ 97065834).
従って, $\pi = 3 + /\!/7, 15, 1, 293, .../\!/ \approx 355/113$. 4 項目の正しい値は 292.

終