Small Fry – Primes A'counting

Valentín Albillo (HPCC #1075)

Welcome to a new series which will feature some of my *shortest* routines and programs (usually coded for the HP-71B but occasionally also for other models such as the HP-15C, 34C, 41C, etC.), where the accent is placed on *lightness*, both lightness of content *and* lightness of exposition (<= 1 page).

Let's begin with the topic of ***prime counting***, i.e., *finding out how many prime numbers there are up to a given limit N*. For large N, generating all primes up to N and returning the count is prohibitive. In fact, getting the *exact* count for N>10^{20} is a daunting task requiring utmost computational power. But if we content ourselves with an *asymptotic* approximation, where "asymptotic" means the *larger* is N the *smaller* is the *relative* error, then this 8-liner is a *fast*, *extremely accurate* one:

```
100  DEF FNZ(Z) @ IF Z=2 THEN FNZ=PI*PI/6 ELSE IF Z=3 THEN FNZ=1.20205690316
110  IF Z=4 THEN FNZ=1.08232323371 ELSE IF Z=5 THEN FNZ=1.03692775514
120  IF Z=6 THEN FNZ=1.01734306198 ELSE IF Z=7 THEN FNZ=1.00834927738
130  IF Z<8 THEN END ELSE S=1 @ T=0 @ N=2
140  S=S+N^(-Z) @ N=N+1 @ IF S<>T THEN T=S @ GOTO 140 ELSE FNZ=S
150  DEF FNR(N) @ J=LN(N) @ R=1 @ N=1 @ K=1
160  R=R+1/(K*FNZ(K+1))*J^K/FACT(K) @ IF R<>N THEN K=K+1 @ N=R @ GOTO 160
170  FNR=INT(R+.5) @ END DEF
```

This code implements *two multiline user-defined functions*, namely:

```
FNR(N)   gives a very close approximation to the number of primes up to N
FNZ(N)   auxiliary: returns Riemann's Z function for integer N>1, fast
```

Let's test our function **FNR(N)** for N = 10^3, 10^6, 10^9, 10^{12}, 4.10^{16} :

```
> FOR I=3 TO 12 STEP 3 @ DISP 10^I,FNR(10^I) @ NEXT I @ DISP 4E16,FNR(4E16)
   1000       168               { exact  = 168               , % err = 0             }
   1000000    78527             { exact = 78498              , % err = 0.0369%       }
   1000000000 50847455          { exact = 50847534           , % err =-0.000155%     }
   1.E12      37607910541       { exact = 37607912018        , % err =-0.000004%     }
   4.E16      1.07529277875E15  { exact = 1075292778793150   , % err =-0.000000004%  }
```

As you can see, this is ***very** close* to the exact values and the relative error (which never is that big anyway) decreases very quickly for large N. How well does it fare against other well-known approximations ? Let's check for N large and small:

Approximation	# Primes up to N=1000	# Primes up to N = 10^{12}
N/LN(N)	145 (-13.69%)	36191206825 (-3.767040%)
Log Integral Li(N)	177 (5.36%)	37607950280 (0.000102%)
This approximation	168 (0.00%)	37607910541 (-0.000004%)
Exact count	168 -	37607912018 -

Pretty good, isn't it ? And *lots* faster than generating and counting primes ! ...