“More Digits” friendly Competition

July 10-12, 2006 — LORIA, Nancy, France

Laurent Fousse, laurent@komite.net
Guillaume Hanrot, Guillaume.Hanrot@loria.fr
Vincent Lefèvre, vincent@vinc17.org
Norbert Müller, mueller@uni-trier.de
Emmanuel Thomé, Emmanuel.Thome@normalesup.org

1 Introduction

In a loosely connected set of events, software for arbitrary precision arithmetic has been tested several times during related conferences:

- CCA'2000, Swansea
- CCA'2002, Malaga *(only presentation)*
- “Many Digit” Friendly Competition, Small TYPES workshop, Nijmegen 2005

In 2006, a new event will happen, the “More Digits” friendly competition, running in parallel to the RNC7 conference in Nancy. It is jointly organized by Laurent Fousse, Vincent Lefèvre, Guillaume Hanrot and Emmanuel Thomé (the winners of the Nijmegen event) and Norbert Müller (whose `iRRAM` won the first competition in Swansea).

The goal is to measure the performance of various numerical computing packages on a set of challenging problems, with a strong emphasis on accuracy.

All packages with arbitrary precision capabilities are welcome to compete, with no restriction on the programming language of implementation, as long as you can run it on our competition machine (a 32-bit Pentium-4 3GHz processor running GNU/Linux). If you develop, maintain, or just use such a software, don't hesitate to take part in the friendly competition, telling us which software you intend to use.

The competition will consist of 16 problems. The main emphasis will be on the correctness of the results, not the timings. Nevertheless, during the competition there will be just one hour of time for each participant to solve all the problems together.

The 16 problems are grouped as follows:

- 6 ‘simple’ formulae
- 8 ‘intermediate’ problems
- 2 ‘difficult’ problems

Of course, the rating as simple, intermediate, and evolved problems is quite subjective...

2 Remarks concerning the competition problems

1. All the problems have (up to five) parameters N, S, A, B, C, where the final sets of parameters will be given to the participants during the competition.

2. For each problem, there will be two parameter sets: The *simple* sets just try to figure out whether the problem can be solved by the participants at all, and the *harder* sets intend to stress the implementation a bit more.

3. During the competition, the parameter sets will be given in a text file, containing one single line consisting of all the five parameters for each parameter set, separated by spaces. Even if a certain parameter is not necessary for the specific problem, it will be contained in the parameter set, e.g. a line with content

   ```
   P07␣harder␣:␣10␣123456␣900000000␣654␣321
   ```

   would define $N = 10$, $S = 123456$, $A = 9 \cdot 10^8$, $B = 654$, and $C = 321$ as harder parameter set for problem P07.

4. The parameters are always natural numbers n with $0 \leq n \leq 10^9$, so they can safely be stored in a 32-bit-number. Some of the parameters will be restricted in the following to mark the difference between *simple* and *harder*.

5. Some problems depend on a sequence of pseudorandom numbers, here given via the simple Marsaglia generator: For a seed given by S, this sequence is defined via $\mathcal{X}_0 := S$ and $\mathcal{X}_{i+1} = (69069 * \mathcal{X}_i + 3) \bmod 2^{31}$, so each value $\mathcal{X}_i$ can be stored as a (signed) 32-bit-integer. The value $\mathcal{X}_0 = S$ will always be discarded. So e.g. using the seed $S = 10$, the first 3 values to be used are $\mathcal{X}_1 = 690693$, $\mathcal{X}_2 = 460834564$ and $\mathcal{X}_3 = 1527353911$.

6. In almost all problems, one of the parameters will denote a precision N. In this case, we expect that the output contains (at least) the first N significand decimal digits of the exact mathematical result.

   The parameter sets will always be chosen so that all correct solutions with more than the necessary number of significand decimal digits will coincide on those first N digits; this will also be true for any solution having the precise number of digits and an error of at most 0.65ulp. So usually you don't need to remove any trailing exponents or care for the exact number of decimals (if it's large enough)!

   To simplify the verification of the results, first any nondecimal characters will be discarded, then any leading zeros. From the resulting string, exactly the first N characters will be used. If you are familiar with gnu utilities:

   ```
   tr -d -c "[0-9]" < $1 | sed -e "s/^0*//" | dd count=1 bs=$2
   ```

3 Competition problems

- **Problem P01**, parameters: N, A, B, C

  Print N leading decimals from $z = \sqrt[C]{A/B}$

Restrictions	N	S	A	B	C
simple	$10^4..10^5$	–	–	>0	>0
harder	$10^5..10^6$	–	–	>0	>0

- **Problem P02**, parameters: N, A, B

  Print N leading decimals from $z = e^{\cos(A/B)}$

Restrictions	N	S	A	B	C
simple	$10^3..10^4$	–	–	>0	–
harder	$10^4..10^5$	–	–	>0	–

- **Problem P03**, parameters: N, A, B, C

  Consider the inverse trigonometric functions arccos and arcsin.

  Print N leading decimals from

$$\arccos(A/C) + \arcsin(B/C)$$

Restrictions	N	S	A	B	C
simple	$10^3..10^4$	–	$(*)$	$(*)$	$(*)$
harder	$10^4..10^5$	–	$(*)$	$(*)$	$(*)$

  (*): $A < C$ and $B < C$.

- **Problem P04**, parameters: N, A, B

  Consider the Γ function, cf. `en.wikipedia.org/wiki/Gamma_function`.

  Print N leading decimals from $z = \Gamma(A/B)$

Restrictions	N	S	A	B	C
simple	$10^2..10^3$	–	>0	>0	–
harder	$10^3..10^4$	–	>0	>0	–

- **Problem P05**, parameters: N, A, B

  Consider the Riemann ζ function, cf. `en.wikipedia.org/wiki/Riemann_zeta_function`.

  Print N leading decimals from $\zeta(2 + \sin(A/B))$

Restrictions	N	S	A	B	C
simple	100..300	–	–	>0	–
harder	500..2000	–	–	>0	–

- **Problem P06**, parameters: N, A, B

  Consider the error function erf, cf. `en.wikipedia.org/wiki/Error_function`.

  Print N leading decimals from $\mathrm{erf}(\sin(A/B))$

Restrictions	N	S	A	B	C
simple	1000..5000	–	–	> 0	–
harder	10000..50000	–	–	> 0	–

- **Problem P07**, parameters: N, S, A

  Using seed S, determine the first N significand decimals of

$$\sum_{i=1}^{A} \frac{1}{\mathcal{X}_i}$$

Restrictions	N	S	A	B	C
simple	10	$(*)$	$10^4..10^5$	–	–
harder	5	$(*)$	$10^8..2 \cdot 10^8$	–	–

(*) The seed will be chosen such that $\mathcal{X}_i \neq 0$ for all used values.

- **Problem P08**, parameters: N, S, A, B

  Using seed S, determine the first N significand decimals of

$$\sum_{i=1}^{A} |\sin(\mathcal{X}_i/B)|$$

Restrictions	N	S	A	B	C
simple	10..20	–	$10^3..10^4$	> 0	–
harder	100..200	–	$10^4..10^5$	> 0	–

- **Problem P09**, parameters: N, A, B

  Consider the 'logistic sequence' defined by $x_0 = 0.5$, $x_{n+1} = c \cdot x_n \cdot (1 - x_n)$, where $c = A/B$ such that $3.5 < c < 4$. For each of the values x_n, let

$$a_n := \begin{cases} 1, & \texttt{if } x_n < 0.5 \\ 0, & \texttt{if } x_n \geq 0.5 \end{cases}$$

  and let

$$z := \sum_{i=1}^{\infty} a_i \cdot 2^{-i}$$

  Print N leading decimals from z.

Restrictions	N	S	A	B	C
simple	50..200	–	$(*)$	$(*)$	–
harder	5000..10000	–	$(*)$	$(*)$	–

$(*)$: $3.5 < A/B < 4$

- **Problem P10**, parameters: N, S, A

  Consider an $(A\times A)$-Matrix M with random entries:

$$M = \begin{pmatrix} \mathcal{X}_1 & \mathcal{X}_2 & \cdots & \mathcal{X}_A \\ \mathcal{X}_{A+1} & \mathcal{X}_{A+2} & \cdots & \mathcal{X}_{2A} \\ \vdots & \vdots & & \vdots \\ \mathcal{X}_{A^2-A+1} & \mathcal{X}_{A^2-A+2} & \cdots & \mathcal{X}_{A^2} \end{pmatrix}$$

  Let $\overline{M} = ((\overline{m}_{i,j}))$ be the inverse of the matrix M and let z be the sum of all the absolute values in $\overline{M}$:

$$z := \sum_{i,j} |\overline{m}_{i,j}|$$

  Print N leading decimals from z.

Restrictions	N	S	A	B	C
simple	10	$(*)$	50..100	–	–
harder	10	$(*)$	200..300	–	–

  $(*)$: The seed S will be given such that M is not singular

- **Problem P11**, parameters: A, B, C

  Consider the algebraic number $\sqrt[4]{A/B}$ as a regular continued fraction

$$\sqrt[4]{A/B} = \alpha_0 + \cfrac{1}{\alpha_1 + \frac{1}{\alpha_2 + \frac{1}{\cdots}}}$$

  with integers α_i (where $\alpha_i > 0$ for $i > 0$).

  Determine α_C and the number $\#_1^C$ of indices $i, 0 \leq i \leq C$, where $\alpha_i = 1$ holds.

  (Apart from these two numbers, the output should contain no further digits.)

Restrictions	N	S	A	B	C
simple	–	–	$(*)$	$(*)$	1000..10000
harder	–	–	$(*)$	$(*)$	100000..1000000

  $(*)$: A/B will not be a square of a rational number.

- **Problem P12**, parameters: N, S, A

  Consider the sequence given by $y_0 = 0$ and $y_i = \sin(y_{i-1} + \mathcal{X}_i)$.

  Print N leading decimals from y_A.

Restrictions	N	S	A	B	C
simple	10	–	1000..2000	–	–
harder	10	–	100000..500000	–	–

- **Problem P13**, parameters: N, S, A, B

  Consider the following function f_S defined via a random power series:

$$f_S(x) := \sum_{i=1}^{\infty} \frac{\mathcal{X}_i - 2^{30}}{i!} \cdot x^i$$

  Determine the first N significand decimals of $f_S(\sqrt{A/B})$

Restrictions	N	S	A	B	C
simple	500..2000	–	$1..10^4$	$10^3..10^9$	–
harder	10000..30000	–	$1..10^6$	$10^3..10^9$	–

- **Problem P14**, parameters: N, A, B

  Consider the functions $f(x) = e^{-x/A}$ and $g(x) = \tan(x/B)$.

  In the interval $(0, B \cdot \pi/2)$, there is a unique number z such that $f(z) = g(z)$.

  Determine the first N significand decimals of z.

Restrictions	N	S	A	B	C
simple	1000..5000	–	> 0	> 0	–
harder	10000..50000	–	> 0	> 0	–

- **Problem P15**, parameters: N, A, B

  Determine the first N significand decimals from the value of the following integral:

$$\int_{-1}^{1} \sin(A \cdot \cos(B \cdot x + C))dx$$

Restrictions	N	S	A	B	C
simple	20..30	–	$A + B < 50$		–
harder	100..500	–	$A + B < 50$		–

- **Problem P16**, parameters: N, A, B, C

  Determine the first N significand decimals of the C-th derivative $z = f^{(C)}(x_0)$ of f at $x_0 = 1/\sqrt{A}$ for the following function f:

$$f(x) = \sqrt{\sin(x) + \frac{1}{\sqrt{B}}}$$

Restrictions	N	S	A	B	C
simple	20	–	$1..10^4$	$1..10^6$	10..100
harder	20	–	$1..10^4$	$1..10^6$	100..1000

4 Rules for the competition

In advance, there will be access to the computer running the competition in order to install and test the software. During the competition, each participant will have access to the computer only for one hour. In this time all the problems have to be solved.

Each problem will be weighted with **5 points**, so that a total of **85 points** can be reached. These points will be given as follows:

- **5 points**, if correct solutions to both simple and harder parameter set are given.
- **4 points**, if the correct solution to one set (usually the simpler set) is given and there is no solution at all for the other set (usually the harder set).
- **2 points**, if both solutions are given, but exactly one of them is correct.
- **0 points**, if there are no or only wrong solutions

Unless there is a tie between two or more participants, the participant with the highest number of points wins the competition. In case of a tie, there will be a play-off between the best participants consisting of the same problems, but new parameter sets.

If there is again a tie, the timings $(\texttt{time})_{i,j}$ of the new harder parameter set for problem i and participant j will also be important: Then the winner will be the participant j with the highest value of

$$\sum_{i=1}^{20} \frac{\min\{(\texttt{time})_{i,k} \mid k \in \texttt{tie}\}}{(\texttt{time})_{i,j}}$$

5 Example parameter sets and partial results

If the following you find parameters sets to test your solutions. There are *trivial* sets, just to be sure that you and we understand the problems in the same way, and of course *simple* and *harder* sets that come close to the sets you have to deal with during the competition.

On the web page you will find a file `run.rc` (based on these sets) that you can take as a basis for a script to run all the problems. We recommend to use a similar script during the competition. We also recommend to use `ulimit` to restrict the execution time for each single program to two minutes (to be quite sure to meet the time limit of one hour for your time slot during the competition). At the start of the competition, you will get a similar file with the new parameter sets (but without the *trivials* sets).

If the results consist of more than 20 decimals, only the first and last 10 are printed here.

- Problem 1
  trivial: `20 0 5 2 4`, result: `12574334296829354083`
  simple: `15000 0 123456  10000 1000`, result: `1002516460...9374523580`
  harder: `200000 0 234567 10000 987654321`, result: `1000000003...5027065031`

- Problem 2
  trivial: `20 0 1 7 0`, result: `26907319593468288051`
  simple: `5000 0 6 7 0`, result: `1924372742...3629253638`
  harder: `50000 0 2348 11 0`, result: `2677658027...8777345736`

- Problem 3
  trivial: `10 0 21 11 30`, result: `1170822438`
  simple: `5000 0 2923 2813 3000`, result: `1442910413...5126981869`
  harder: `40000 0 3922 813 4000`, result: `4024825725...7336476259`

- Problem 4
  trivial: `10 0 233 100 0`, result: `1188192811`
  simple: `500 0 258 100 0`, result: `1408442894...6107969884`
  harder: `3000 0 3923 100 0`, result: `1211928509...5377987124`

- Problem 5
  trivial: `20 0 425 100 0`, result: `10107652671515329327`
  simple: `200 0 9999 10000 0`, result: `1236761124...8470451468`
  harder: `1000 0 47123886 10000000 0`, result: `1382241533...2100922049`

- Problem 6
  trivial: `20 0 9995 10000 0`, result: `76581109156264087697`
  simple: `2000 0 9995 10000 0`, result: `7658110915...7872601970`
  harder: `30000 0 1512 1000 0`, result: `8419822444...8647662558`

- Problem 7
  trivial: `9 12324 20 2 0`, result: `445506818`
  simple: `10 12324 20000 2 0`, result: `8940079109`
  harder: `5 12314 200000000 10 0`, result: `19378`

- Problem 8
  trivial: `16 12344 5 2 0`, result: `4002887262801917`
  simple: `16 12344 5000 2 0`, result: `3178893797767515`
  harder: `150 12384 100000 10 0`, result: `6383778312...3402802276`

- Problem 9
  trivial: `10 0 166 43 0`, result: `4107916672`
  simple: `100 0 161 43 0`, result: `2893154707...4014980657`
  harder: `7000 0 15 4 0`, result: `2893697060...6039082717`

- Problem 10
  trivial: `10 12157 2 0 0` , result: `6292680180`
  simple: `10 12165 80 0 0`, result: `7880687074`
  harder: `10 12385 250 0 0`, result: `5297164525`

- Problem 11
  trivial: `0 0 1 2 10 0 2`, result: `lastpq=25 ones=5`

simple: `0 0 82 13 5246`, result: `lastpq=10532 ones=2132`
harder: `0 0 1799 13 456123`, result: `lastpq=32042016 ones=189669`

- Problem 12
  trivial: `10 24898 2 0 0`, result: `7764437589`
  simple: `10 24905 1000 0 0`, result: `9525031372`
  harder: `10 14905 200000 0 0`, result: `8033815492`

- Problem 13
  trivial: `2 102312 1 1 0`, result: `23`
  simple: `2000 102348 9999 1001 0`, result: `1314897562...7305527079`
  harder: `20000 102317 999999 1001 0`, result: `8321021150...6332873222`

- Problem 14
  trivial: `10 0 5 4 0`, result: `2265057663`
  simple: `1000 0 10 10 0`, result: `5313908566...3281566724`
  harder: `20000 0 1234 4321 0`, result: `1372607407...5127333563`

- Problem 15
  trivial: `10 0 3 2 19`, result: `6542558170`
  simple: `20 0 9 11 36`, result: `10135644729604723625`
  harder: `253 0 20 20 20`, result: `7160868855...4539015485`

- Problem 16
  trivial: `20 0 3 5 1`, result: `42042593636979459912`
  simple: `20 0 3210 5432 30`, result: `11984464506645085515`
  harder: `20 0 4321 543210 200`, result: `13932674643255787917`

To get an idea how much time the actual solutions might take, the following table contains timings (in seconds) for the harder problem sets and for the `iRRAM` library, all taken on a Pentium-4 3GHz processor.

problem	1	2	3	4	5	6	7	8
time[s]	18.31	12.27	17.72	12.50	15.13	18.69	17.36	14.70
problem	9	10	11	12	13	14	15	16
time[s]	19.92	23.16	13.91	9.76	10.74	11.96	19.28	4.15

Please note that `MPFR` and `iRRAM` are out of competition: As organizers, we were of course able to influence the rules, the problems and the parameter sets; so a participation in the competition itself would be highly unfair.