

maXbox6

all about code blocks

About

EKON 30

maXbox

RockChain

TEE

X-Mas 2025

recent posts

[Rosetta Code](#)

[Free Google Translator](#)

[API II](#)

[Free Google](#)

[Translator API](#)

[TEE Kraftwerk Concert](#)

[Starter165 Addons](#)

[Machinelearning Steps](#)

[Statistics with](#)

[Statsmodels](#)

about



Rosetta Code

February 23, 2026

maXbox Tutorial 169

Rosetta Code is a programming chrestomathy site. The idea is to present solutions to the same task in as many different languages as possible, to demonstrate how languages are similar and different, and to aid a person with a grounding in one approach to a problem in learning another. Rosetta Code currently has 1,342 tasks, 400 draft tasks, and is aware of 988 languages, though we do not (and cannot) have solutions to every task in every language.

We will show that in 5 Examples.

1. Cyclops numbers – Rosetta Code

A cyclops number is a number with an odd number of digits that has a zero in the center, but nowhere else. They are named so in tribute to the one eyed giants Cyclops from Greek mythology.

Cyclops numbers can be found in any base. This task strictly is looking for those cyclops numbers in base 10.

```

166 [Do full suite of Cyclops tasks]
167 begin
168   GetCyclopsNumbers(Memo);
169   GetPrimeCyclops(Memo);
170   GetBlindPrimeCyclops(Memo);
171   GetPalindromicPrimeCyclops(Memo);
172 end;
173
174 Begin //@main
175   writeln(#$d8d#$de42);
176   maxForm1.setconsole;
177   memo2.font.size:= 10;
178   memo2.font.name:= 'Courier';
179   DisplayCyclopsNumbers(memo2);
180 End.

```

```

XCompiler Message Count: 6
maxBox6 1467_cyclop_numbers.txt Compiled done: 19/02/2026 10:33:39
debug: 59- 4294967295 err:0
First 50 Cyclops Numbers
0 101 102 103 104 105 106 107 108 109
201 302 303 304 305 306 307 308 309 301
302 303 304 305 306 307 308 309 401 402
403 404 405 406 407 408 409 501 502 503
504 505 506 507 508 509 601 602 603 604
First 50 Prime Cyclops Numbers
101 103 107 109 307 401 409 503 509 601
407 701 709 809 907 11027 11047 11057 11059 11069
11071 11083 11087 11093 12011 12037 12041 12045 12049 12071
12073 12097 13033 13037 13043 13049 13063 13093 13099 14011

```

https://sourceforge.net/projects/maxbox5/files/examples/1467_cyclop_numbers.txt/download

2. Damm algorithm

The Damm algorithm is a checksum algorithm which detects all single digit errors and adjacent transposition errors. Verify the checksum, stored as last digit of an input.

https://rosettacode.org/wiki/Damm_algorithm#Pascal

https://sourceforge.net/projects/maxbox5/files/examples/1468_damm_algol.txt/download

```

1 function Damm(s : string) : BOOLEAN;
2 VAR interim,i : Byte; //UInt8;
3 BEGIN
4   interim := 0;
5   i := 1;
6   WHILE i <= length(s) DO Begin
7     interim := b[interim][ORD(s[i])-ORD('0')];
8     INC(i);
9   END;
10  result:= interim=0;
11 END;

```

debug: 4- 4294967295 err:0



5724 is valid

5727 is invalid

112946 is valid

112949 is invalid

mX5 executed: 23/02/2026 10:11:12 Runtime: 0:0:2.375 Memload: 65% use

3. Chessgrains

Calculate the number of grains of wheat on a chessboard given that the number on each square doubles.

There once was a wise servant who saved the life of a prince. The king promised to pay whatever the servant could dream up. Knowing that the king loved chess, the servant told the king he would like to have grains of wheat. One grain on the first square of a chess board, with the number of grains doubling on each successive square.

There are 64 squares on a chessboard (where square 1 has one grain, square 2 has two grains, and so on).


Write code that shows: how many grains were on a given square, and the total number of grains on the chessboard

```
1  for it:= 1 to 64 do writ(bigpow(2,it))
2
3  2
4  4
5  8
6  16
7  32
8  64
9  128
10 256
11 512
12 1024
13 2048
14 4096
15 8192
16 16384
17 32768
18 65536
19 131072
20 262144
21 524288
22 1048576
23 2097152
24 4194304
25 8388608
26 16777216
27 33554432
28 67108864
29 134217728
30 268435456
31 536870912
32 1073741824
33 2147483648
34 4294967296
35 8589934592
36 17179869184
37 34359738368
38 68719476736
39 137438953472
40 274877906944
41 549755813888
42 1099511627776
43 2199023255552
44 4398046511104
45 8796093022208
46 17592186044416
47 35184372088832
48 70368744177664
49 140737488355328
50 281474976710656
51 562949953421312
52 1125899906842624
```

```

53 2251799813685248
54 4503599627370496
55 9007199254740992
56 18014398509481984
57 36028797018963968
58 72057594037927936
59 144115188075855872
60 288230376151711744
61 576460752303423488
62 1152921504606846976
63 2305843009213693952
64 4611686018427387904
65 9223372036854775808
66 18446744073709551616

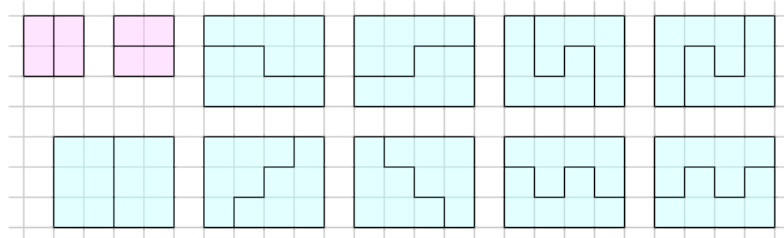
```

mX5  executed: 23/02/2026 10:18:35 Runtime: 0:0:5.55 Memload: 68% use

This problem is a classic illustration of exponential growth. While the starting number is small, the repeated doubling leads to a staggering final total.

4. Cut_a_rectangle

A given rectangle is made from $m \times n$ squares. If m and n are not both odd, then it is possible to cut a path through the rectangle along the square edges such that the rectangle splits into two connected pieces with the same shape (after rotating one of the pieces by 180°). All such paths for 2×2 and 4×3 rectangles are shown below.



Write a program that calculates the number of different ways to cut an $m \times n$ rectangle. Optionally, show each of the cuts.

```

1  initTableDir();
2
3      for y:= 1 to 8 do
4          for x:= 1 to y do
5              if not Odd(x) or not Odd(y) then begin
6                  writeln(format('%d x %d: %d', [y, x, msolve(
7                      //writ(inttostr64(msolve(y, x, true) ))
8                  end;

```

https://sourceforge.net/projects/maxbox5/files/examples/1469_cut_a_rectangle1.txt/download

Cut a rectangle – Rosetta Code**5. CubicSpecialPrimes**

n is smallest prime such that the difference of successive terms are the smallest cubics of positive integers, where $n < 15000$.

```

32   {Find first cube increment that is prime}
33   PP:=N -I*I*I;
34   if PP>limit then break;1
35   //if Sieve.Flags[PP] then
36   if Sieve.isprime(PP) then begin
37     Inc(Count);
38     S:=S+Format('%6D',[PP]);
39     if (Count mod 5) = 0 then S:=S+CRLF;
40     {Step to next cube position}
41     I:=I+1;
42     N:=PP;
43   end
44   else Inc(I);
45 end;
46 Memo.Lines.Add(Format('There are %d cubic special primes',[count]));
47 Memo.Lines.Add(S);
48 finally

```

```

Compiler Message Count: 1
maXbox5 1470_cCubicSpecialPrimes.txt Compiled done: 22/02/2026 20:37:48
debug: 21- 4294967295 err:0
S
There are 23 cubic special primes
 2   3   11  19   83
1011 2027 2243 2251 2467
2531 2539 2539 3547 4547
5059 10891 12619 13619 13627
13691 13907 14419
maXbox5 executed: 22/02/2026 20:37:48  Runtime: 0:0:2.615  Memload: 66% use
RemObjects Pascal Script. Copyright (c) 2004-2026 by RemObjects Software & maXbox5

```

https://sourceforge.net/projects/maxbox5/files/examples/1470_cCubicSpecialPrimes.txt/download

Cubic special primes – Rosetta Code

https://sourceforge.net/projects/maxbox5/files/examples/1470_CubicSpecialPrimes1.txt/download

Here's another program answering fairly meaningless questions but providing a good programming exercise with a little math mixed in.

CheatSheet:

<https://claude.ai/public/artifacts/11968d9f-d02c-4183-a7aa-68dcfefdf382e>

Mindmap:

<https://claude.ai/public/artifacts/8a58da49-1753-4522-b0a2-8207942b300e>



maXbox Topic Map

Advertisements

Occasionally, some of your visitors may see an advertisement here, as well as a [Privacy & Cookies banner](#) at the bottom of the page. You can hide ads completely by upgrading to one of our paid plans.

[UPGRADE NOW](#) [DISMISS MESSAGE](#)

Posted in Code, Data Science, maXbox, Tools

Leave a comment

← [Free Google Translator API II](#)

[Blog at WordPress.com.](#)