

```

1: program All_LottoCombinations_bit_Power;
2: //*****
3: // To provide a fast algo with large numbers (45bit ore more) can turn into a
4: // complicated debate. Here's a solution for compiler or interpreter.
5: // ZWEIHOCHFUNDVIERZIG = 35184372088832; by Max & Beat Straehl
6: // time consuming is about 45 sec.(minutes maXbox) without optimisation days!!
7: // number of lotto spheres are 45 with 6 crosses on it(change)
8:
9: const NUMBER_OF_SPHERES = 45;
10:     NUMBER_OF_BITS_ARRAY_UPPER_END = 255;
11:
12: var counter, counterCopy, counterOffset, range: int64;
13:     bitCounter, bitCount, bitPosition, combCounter: integer;
14:     byteFragmentIndexofCounter: integer;
15:     positionOfLowestBitOutofSix: integer;
16:     numberOfBits: array[0..NUMBER_OF_BITS_ARRAY_UPPER_END] of byte;
17:     before, after: TDateTime;
18:     combstr: string;
19:     lottoList: TStringList;
20:
21: function powerOf2(exponent: integer): int64;
22: var shifter: int64;
23:     counter: integer;
24: begin
25:     shifter:= 1;
26:     for counter:= 1 to exponent do
27:         shifter:= shifter + shifter;
28:     result:= shifter;
29: end;
30:
31: procedure presetBitCountLookupTable(highestIndex: integer);
32: var arrayIndex: integer;
33:     numberOfBitsSet: integer;
34:     byteValue: integer;
35: begin
36:     //preset bit count lookup table
37:     for arrayIndex:= 0 to highestIndex do begin
38:         byteValue:= arrayIndex;
39:         numberOfBitsSet:= 0;
40:         for bitCounter:= 1 to 8 do begin
41:             if ((byteValue and 1) = 1) then
42:                 inc(numberOfBitsSet);
43:             byteValue:= byteValue shr 1;
44:         end;
45:         numberOfBits[arrayIndex]:= numberOfBitsSet;
46:     end;
47: end;
48:
49: begin //main
50:     presetBitCountLookupTable(NUMBER_OF_BITS_ARRAY_UPPER_END);
51:     range:= powerOf2(NUMBER_OF_SPHERES);
52:     lottoList:= TStringList.create;
53:     before:= Time;
54:     combCounter:= 0;
55:     counter:= 0;
56:     lottoList.add('All lotto combinations 6 from ' +IntToStr(NUMBER_OF_SPHERES));
57:     lottoList.add('*****');
58:     while counter < range do begin //check cases
59:         bitCount:= 0;
60:         counterCopy:= counter;
61:         for byteFragmentIndexofCounter:= 0 to 5 do begin
62:             bitCount:= bitCount+numberOfBits[counterCopy mod 256];
63:             counterCopy:= counterCopy shr 8;
64:         end;
65:         if (bitCount=6) then begin
66:             counterCopy:= counter;
67:             positionOfLowestBitOutofSix:= -1;
68:             inc(combCounter);
69:             //write('combination #' + inttostr(combinationCounter) +': ');
70:             for bitCounter:=1 to NUMBER_OF_SPHERES do begin
71:                 if ((counterCopy and 1) = 1) then begin
72:                     //write(inttostr(bitCounter) + ' ');
73:                     combstr:= combstr+ ' '+IntToStr(bitCounter);
74:                     if (positionOfLowestBitOutofSix= -1) then
75:                         positionOfLowestBitOutofSix:= bitCounter;
76:                 end;

```

```

77:         counterCopy:= counterCopy shr 1;
78:     end;
79:     //writeln(' ');
80:     lottoList.add('# '+IntToStr(combCounter)+' ': combstr)
81:     combstr:= '';
82:     counterOffset:= 1;
83:     bitPosition:= 1;
84:     while (bitPosition<positionOfLowestBitOutOfSix) do begin
85:         counterOffset:= counterOffset + counterOffset;
86:         inc(bitPosition);
87:     end;
88:     counter:= counter + counterOffset;
89: end else
90:     counter:= counter + 1;
91: end; //while
92: after:= Time;
93: Writeln('Save lotto file at: ' +TimeToStr(after) + ' '+DateToStr(date));
94: lottoList.SaveToFile('All_lotto_combinations2.txt');
95: lottoList.Free;
96: Writeln('Lotto report started at: ' + FormatDateTime('hh:nn:ss.zzz',before));
97: Writeln('Lotto report ended at: ' + FormatDateTime('hh:nn:ss.zzz',after));
98: Writeln('Lotto report lasted: ' +FormatDateTime('hh:nn:ss.zzz',after-before));
99: Writeln('6 of ' +IntToStr(NUMBER_OF_SPHERES)+' are: ' +IntToStr(combCounter));
100: end.
101:
102:
103: //files at:
104: www.softwareschule.ch/download/97_pas_lottocombinations_beat_plus.pdf
105: for compiler file at:
106: www.softwareschule.ch/download/maxbox97_lottoproject_allcombinations_dpr.txt
107:
108: {
109: maXbox 97_pas_lottocombinations_beat_plus Compiled done: 28.04.2010 14:41:56
110: -----
111: Save lotto file at: 15:27:42 28.04.2010
112: Lotto report started at: 14:41:56.970
113: Lotto report ended at: 15:27:42.867
114: Lotto report lasted: 00:45:45.897
115: 6 of 45 are: 8145060
116: runNMax executed on: 28.04.2010 15:27:55 }
117:
118:
119: You may know this:
120: Lotto combination 1-2-3-4-5-6 has the same chance of being drawn as any other
121: lottery combination. But the combination 1,2,3,4,5,6 should not be played. It
122: represents a weird arrangement of lottery numbers. Why?
123: If one hits the jackpot, you should be advised to never play the statistically
124: dreadful 1,2,3,4,5,6!
125:
126: 1. If the lotto 6/45 has a total of 8,145,060 comb., and only one is drawn as
127: the winner, the probability is undeniably 1 / 8,145,060.
128:
129: All shapes have, however, the same probability of appearance. Reality only has
130: a different idea! There been thousands of lotto drawings conducted worldwide.
131: We should have seen by now a lottery combination of the type 1,2,3,4,5,6. There
132: has been none so far or you know anyone?.
133:
134: 2. Furthermore, you can use the code to generate all lotto combinations in
135: lexicographic order for any experiment or game. Beware! The output files can be
136: extremely large, for 6/45 is about 240 MB!
137:
138: One interesting point at all is the binominal coefficient. It means that you get
139: the same amount of combinations with 6 or 39 tips (crosses) of 45!
140:
141: Cause the mathematics shows
142: 3. Combination (binominal coefficient)= nCr = n!/(k! * (n-k)!)
143:                                     45! /6! * 39!
144: so you can see 6 or 39 are interchangeable...
145:
146: The probabilities are equal, but the combinations are not equal.
147: -----
148:
149:
150:

```