```
> restart;
> interface(warnlevel = 0) :   #  Maple 12
> with(RandomTools) :
> with(ListTools) :
```

This procedure illustrates Bell's Inequality by setting up a collection of items possessing a set of attributes: For example, light bulbs

Wattage: 60W or 100W. B≡60W, not B≡100W
Appearance: clear or frosted. A≡clear, not A≡frosted
Color: green or red. C≡green, not C≡red

No. of 100W clear bulbs + No. of 60W red bulbs ≥ No. of clear red bulbs

In general, nAnotB + nBnotC ≥ nAnotC

This procedure uses the Maple procedure Generate() to generate pseudo-random set of items having the above attributes.

BellIn(*pA,pnA,pB,pnB,pC,pnC,numb***)**
where *pA = A, pnA = not A*
pB = B, pnB = not B
pC = C, pnC = not C
numb = No. of items

Ex: BellIn(*clear,frosted,60W,100W,green,red,100***) - generates collection of** *100* **light bulbs**

Other examples: BellIn(*walking,running,laces,velcro,black, white, 100***) -- athletic shoes**
BellIn(*sedan,coupe,auto,stick,blue, red,100***) - cars**

BellIn Procedure

```
> BellIn :=proc(pA, pnA, pB, pnB, pC, pnC, numb)
    local i, A, B, C, LB, t, nANotB, nBNotC, nANotC,
          t1, t2, t3, t4, t5, t6, t7, t8;

    LB := [ ];            # the list
    nANotB := 0;          # an attribute set i.e 100 W clear bulb
    nBNotC := 0;          # an attribute set
    nANotC := 0;          # an attribute set
    t1 := 0; t2 := 0; t3 := 0; t4 := 0;     # counters
    t5 := 0; t6 := 0;  t7 := 0; t8 := 0;
    if numb > 0 then
       randomize( );
      for i from 1 to numb do
        A := Generate(choose({pA, pnA}));
        B := Generate(choose({pB, pnB}));
        C := Generate(choose({pC, pnC}));

        LB := [A, B, C]; # let's see what we have
        if LB = [pA, pB, pC] then t1 := t1 + 1;
         elif LB = [pA, pB, pnC] then t2 := t2 + 1;
         elif LB = [pA, pnB, pC] then t3 := t3 + 1;
         elif LB = [pA, pnB, pnC] then t4 := t4 + 1;
         elif LB = [pnA, pB, pC] then t5 := t5 + 1;
```

```
            elif LB = [ pnA, pB, pnC ] then t6 := t6 + 1;
            elif LB = [ pnA, pnB, pC ] then t7 := t7 + 1;
            elif LB = [ pnA, pnB, pnC ] then t8 := t8 + 1;
            end if ;

          if A = pA and B = pnB then   # Check terms of Inequality
                nANotB := nANotB + 1;
          end if;
          if B = pB and C = pnC then
                nBNotC := nBNotC + 1;
          end if;
          if A = pA and C = pnC then
                nANotC := nANotC + 1;
          end if;
        end do;

        print( pA, pB, pC, t1);
        print( pA, pB, pnC, t2) ;
        print( pA, pnB, pC, t3);
        print( pA, pnB, pnC, t4);
        print( pnA, pB, pC, t5);
        print( pnA, pB, pnC, t6);
        print( pnA, pnB, pC, t7);
        print( pnA, pnB, pnC, t8);
        print( );
        print(          Inequality          );
        print( pA and pnB,' nANotB ', nANotB);
        print( pB and pnC,' nBNotC ', nBNotC);
        print( pA and pnC,' nANotC ', nANotC);
        print( nANotC ≤ nANotB + nBNotC );
        print( nANotC ≤ nANotB + nBNotC);
    end if;
end proc:
```

```
> BellIn(clear, frosted, 60W, 100W, green, red, 100);
```

clear, 60 W, green, 7
clear, 60 W, red, 23
clear, 100 W, green, 6
clear, 100 W, red, 15
frosted, 60 W, green, 13
frosted, 60 W, red, 12
frosted, 100 W, green, 13
frosted, 100 W, red, 11

Inequality
clear **and** 100 W, nANotB, 21
60 W **and** red, nBNotC, 35
clear **and** red, nANotC, 38
nANotC ≤ nANotB + nBNotC
38 ≤ 56

(1)

```
> BellIn(walking, running, laces, velcro, black, white, 100);
```

walking, laces, black, 10
walking, laces, white, 7
walking, velcro, black, 14
walking, velcro, white, 17
running, laces, black, 18
running, laces, white, 9
running, velcro, black, 10
running, velcro, white, 15

Inequality
walking **and** *velcro, nANotB,* 31
laces **and** *white, nBNotC,* 16
walking **and** *white, nANotC,* 24
$nANotC \leq nANotB + nBNotC$
$24 \leq 47$ **(2)**