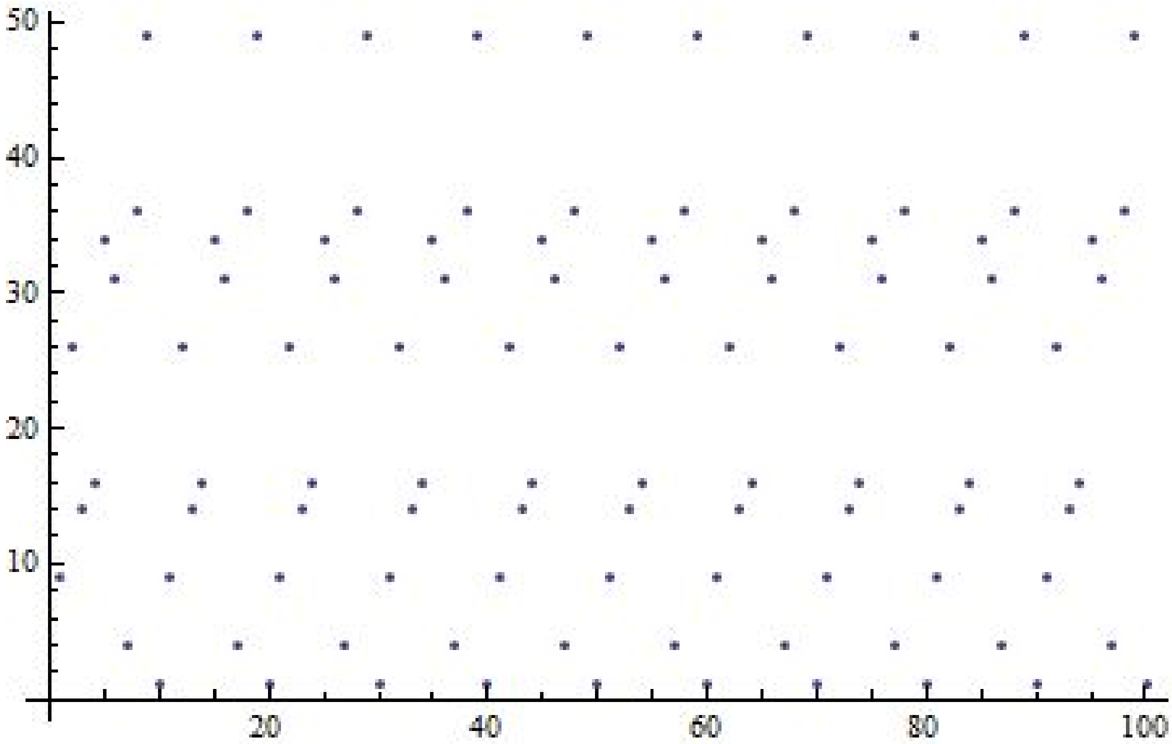
Lab8 Shor's algorithm - general idea on the simple example

Exercise T1 THEORY

Solve assigment II from the file http://www.lassp.cornell.edu/mermin/qcomp/prob4.pdf

Exercise Q1 QUIDE

Implement (in arbitrary programming language on classical computer) function a^x mod N for x=1..100, N=55, a=9 (as in exercise 1) . Observe the period. Optional: produce a plot (in any tool) .



On Wolframalpha you can use

- Table[Mod[9^x, 55], {x, 1, 100, 1}]
- ListPlot[Table[Mod[9^x, 55], {x, 1, 100, 1}]]

Exercise Q2 QUIDE

Quantum gate for calculating a^x mod N function.

Use the draft of the code below to calculate values of function a^x mod N for x=1..100 by using provided expModulo gate in the simulator .

```
//initialize x, N, a
      //int N= ...
      //int a= ...
      //ulong x
     // calculate how many bits we need for N
    ulong ulongN = (ulong)N;
    int width = (int)Math.Ceiling(Math.Log(N, 2));
```

```
// quantum computer initialization
QuantumComputer comp = QuantumComputer.GetInstance();

//input register
Register regX = comp.NewRegister(0, 2 * width);

// output register
Register regY = comp.NewRegister(1, width + 1);

// setting input register with x value
regX.Reset(x);

//setting output register with 1
   // needed  if we execute in a loop
regY.Reset(1);

  // calculate a^x mod N
comp.ExpModulo(regX, regY, a, N);

  //measure
int valueMeasured = (int)regY.Measure();

  Console.WriteLine ("Dla {0} reszta to {1}",x,
valueMeasured);
```

Exercise Q3 QUIDE

Run the whole period funding function in the simulator using the file.