

# Practice 1

## All One Problem

Made by  
5 course student  
group II-10  
Artsiom Drapeza (Artyom Drapezo)

### Exercise 1

1. Create 100 random binary-chromosomes each with 1000 genes.
2. Fitness is the number of “1” in one chromosome – the more the better.
3. Select 2 chromosomes at random from the better half of the population.
4. Create a child chromosome by a one-point-crossover.
5. Give the child a mutation with a probability of  $1/1000 = 0.001$ .
6. Repeat from 2. to 5. 100 times and create the next generation.
7. Repeat 6. until the fitness value does not change any more.
8. Show the result:
  - (1) Display the best chromosome in the 1st, an intermediate & final generation.
  - (2) Display the best and average fitness vs. generation

### Code

Constants.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace lab1
{
    class Constants
    {
        public static int ginCountInChromosom = 1000;

        public static int chromosomCountInPopulation = 100;
    }
}
```

Chromosom.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace lab1
{
    class Chromosom
    {
        private static Random rng = new Random();

        private List<bool> gens;

        private int fitness;

        private double probabilityToBeSelected;
    }
}
```

```

private int selectedTimes = 0;

public Chromosom()
{
    gens = new List<bool>();

    for (int i = 0; i < Constants.ginCountInChromosom; i++)
    {
        gens.Add(rng.NextDouble() > 0.5 ? true : false);
    }

    calculateFitness();
}

public static List<Chromosom> crossover(Chromosom firstParent, Chromosom secondParent)
{
    int cut = rng.Next(Constants.ginCountInChromosom - 1) + 1;

    Chromosom firstChild = new Chromosom();

    List<bool> firstChildGins = new List<bool>();

    firstChildGins.AddRange(firstParent.getGens().GetRange(0, cut));
    firstChildGins.AddRange(secondParent.getGens().GetRange(cut, Constants.ginCountInChromosom -
cut));

    firstChild.setGens(firstChildGins);

    Chromosom secondChild = new Chromosom();

    List<bool> secondChildGins = new List<bool>();

    secondChildGins.AddRange(secondParent.getGens().GetRange(0, cut));
    secondChildGins.AddRange(firstParent.getGens().GetRange(cut, Constants.ginCountInChromosom -
cut));

    secondChild.setGens(secondChildGins);

    List<Chromosom> result = new List<Chromosom>();

    result.Add(firstChild);
    result.Add(secondChild);

    return result;
}

public int calculateFitness()
{
    int f = 0;

    for (int i = 0; i < Constants.ginCountInChromosom; i++)
    {
        f += (gens[i] == true ? 1 : 0);
    }

    fitness = f;

    return f;
}

public void setGens(List<bool> newGens)
{
    gens = newGens;
}

public List<bool> getGens()
{
    return gens;
}

public int getFitness()
{
    return fitness;
}

public void setProbabilityToBeSelected(double newProbability)
{
    probabilityToBeSelected = newProbability;
}

public double getProbabilityToBeSelected()

```

```

        {
            return probabilityToBeSelected;
        }

        public int getSelectedTimes()
        {
            return selectedTimes;
        }

        public void selectedTimesPlusOne()
        {
            selectedTimes++;
        }
    }
}

```

Population.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace lab1
{
    class Population
    {
        private List<Chromosom> chromosoms;

        private Random rng = new Random();

        public Population()
        {
            chromosoms = new List<Chromosom>();

            for (int i = 0; i < Constants.chromosomCountInPopulation; i++)
            {
                chromosoms.Add(new Chromosom());
            }
        }

        public List<int> getFitnessList() {
            List<int> fitnessList = new List<int>();

            for (int i = 0; i < Constants.chromosomCountInPopulation; i++)
            {
                fitnessList.Add(chromosoms[i].calculateFitness());
            }

            return fitnessList;
        }

        public void sortChromosoms()
        {
            chromosoms.Sort((ch1, ch2) => ch1.calculateFitness().CompareTo(ch2.calculateFitness()));
        }

        public void calcProbabilitiesToBeSelected() {
            int fitnessSum = 0;

            for (int i = 0; i < Constants.chromosomCountInPopulation; i++)
            {
                fitnessSum += chromosoms[i].calculateFitness();
            }

            for (int i = 0; i < Constants.chromosomCountInPopulation; i++)
            {
                chromosoms[i].setProbabilityToBeSelected((double) chromosoms[i].getFitness() / fitnessSum);
            }
        }

        public Population getNextGeneration()
        {
            calcProbabilitiesToBeSelected();

            Population nextGeneration = new Population();

            List<Chromosom> nextPopulationChromosoms = new List<Chromosom>();

            for (int i = 0; i < Constants.chromosomCountInPopulation / 2; i++)
            {
                double r1 = rng.NextDouble();

```

```

        double r2 = rng.NextDouble();

        while (r1 == 0.0 || r1 == 1.0)
        {
            r1 = rng.NextDouble();
        }

        while (r2 == 0.0 || r2 == 1.0)
        {
            r2 = rng.NextDouble();
        }

        double probabilitySum1 = 0;
        double probabilitySum2 = 0;

        int indexOfTheFirstParent = 0;
        int indexOfTheSecondParent = 0;

        while (probabilitySum1 < r1)
        {
            probabilitySum1 += chromosomes[indexOfTheFirstParent].getProbabilityToBeSelected();
            indexOfTheFirstParent++;
        }

        while (probabilitySum2 < r2)
        {
            probabilitySum2 += chromosomes[indexOfTheSecondParent].getProbabilityToBeSelected();
            indexOfTheSecondParent++;
        }

        if (indexOfTheFirstParent >= Constants.chromosomCountInPopulation)
        {
            indexOfTheFirstParent = Constants.chromosomCountInPopulation - 1;
        }

        if (indexOfTheSecondParent >= Constants.chromosomCountInPopulation)
        {
            indexOfTheSecondParent = Constants.chromosomCountInPopulation - 1;
        }

        nextPopulationChromosoms.AddRange(Chromosom.crossover(chromosomes[indexOfTheFirstParent],
chromosomes[indexOfTheSecondParent]));
        chromosomes[indexOfTheFirstParent].selectedTimesPlusOne();
        chromosomes[indexOfTheSecondParent].selectedTimesPlusOne();

    }

    nextGeneration.setChromosoms(nextPopulationChromosoms);

    return nextGeneration;
}

public void setChromosoms(List<Chromosom> newChromosoms)
{
    chromosomes = newChromosoms;
}

public List<Chromosom> getChromosoms()
{
    return chromosomes;
}

public int getBestFitness()
{
    int bestFitness = 0;

    for (int i = 0; i < Constants.chromosomCountInPopulation; i++)
    {
        bestFitness = chromosomes[i].calculateFitness() > bestFitness ?
chromosomes[i].calculateFitness() : bestFitness;
    }

    return bestFitness;
}

public int getAvarageFitness()
{
    int fitnessSumm = 0;

    for (int i = 0; i < Constants.chromosomCountInPopulation; i++)
    {
        fitnessSumm += chromosomes[i].calculateFitness();
    }

    return fitnessSumm / Constants.chromosomCountInPopulation;
}

```

```

    }
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace lab1
{
    class Program
    {
        static void Main(string[] args)
        {
            List<int> maxFitnessList = new List<int>();
            List<int> averageFitnessList = new List<int>();

            Population currentGeneration = new Population();

            List<Population> populationsForTable = new List<Population>();
            List<int> populationsIndex = new List<int>();

            int generationCount = 0;

            do
            {
                maxFitnessList.Add(currentGeneration.getBestFitness());
                averageFitnessList.Add(currentGeneration.getAvarageFitness());

                currentGeneration = currentGeneration.getNextGeneration();
                currentGeneration.sortChromosoms();

                if (generationCount % 100 == 0)
                {
                    populationsForTable.Add(currentGeneration);
                    populationsIndex.Add(generationCount);
                }

                generationCount++;

            } while (!shallStop(maxFitnessList));

            populationsForTable.Add(currentGeneration);
            populationsIndex.Add(generationCount);

            using (System.IO.StreamWriter file = new
System.IO.StreamWriter(@"E:\A_WORK\5course\siit\lab1\table.txt"))
            {
                foreach (int i in populationsIndex)
                {
                    file.Write("{0}\t", i);

                }

                file.WriteLine();

                for (int i = 0; i < Constants.chromosomCountInPopulation; i++)
                {
                    foreach (Population p in populationsForTable)
                    {
                        file.Write(string.Format("{0}\t{1:0.0000}\t{2}\t",
p.getChromosoms()[i].getFitness(), p.getChromosoms()[i].getProbabilityToBeSelected(),
p.getChromosoms()[i].getSelectedTimes()));
                    }

                    file.WriteLine();
                }
            }

            for(int i = 0; i < maxFitnessList.Count; i++){
                Console.WriteLine("Iteration: \t" + i +"\t Max fitness: \t" + maxFitnessList[i] + "\t
Average fitness \t" + averageFitnessList[i]);
            }

            using (System.IO.StreamWriter file = new
System.IO.StreamWriter(@"E:\A_WORK\5course\siit\lab1\MaxFitness.txt"))
            {
                foreach (int fitness in maxFitnessList)
                {
                    file.WriteLine(fitness);
                }
            }
        }
    }
}

```

```
        }
    }

    using (System.IO.StreamWriter file = new
System.IO.StreamWriter(@"E:\A_WORK\5course\siit\lab1\AverageFitness.txt"))
    {
        foreach (int fitness in averageFitnessList)
        {
            file.WriteLine(fitness);
        }
    }

    Console.ReadKey();
}

public static bool shallStop(List<int> maxFitnessList)
{
    if (maxFitnessList.Last() == Constants.ginCountInChromosom)
        return true;

    int checkLast = 10;

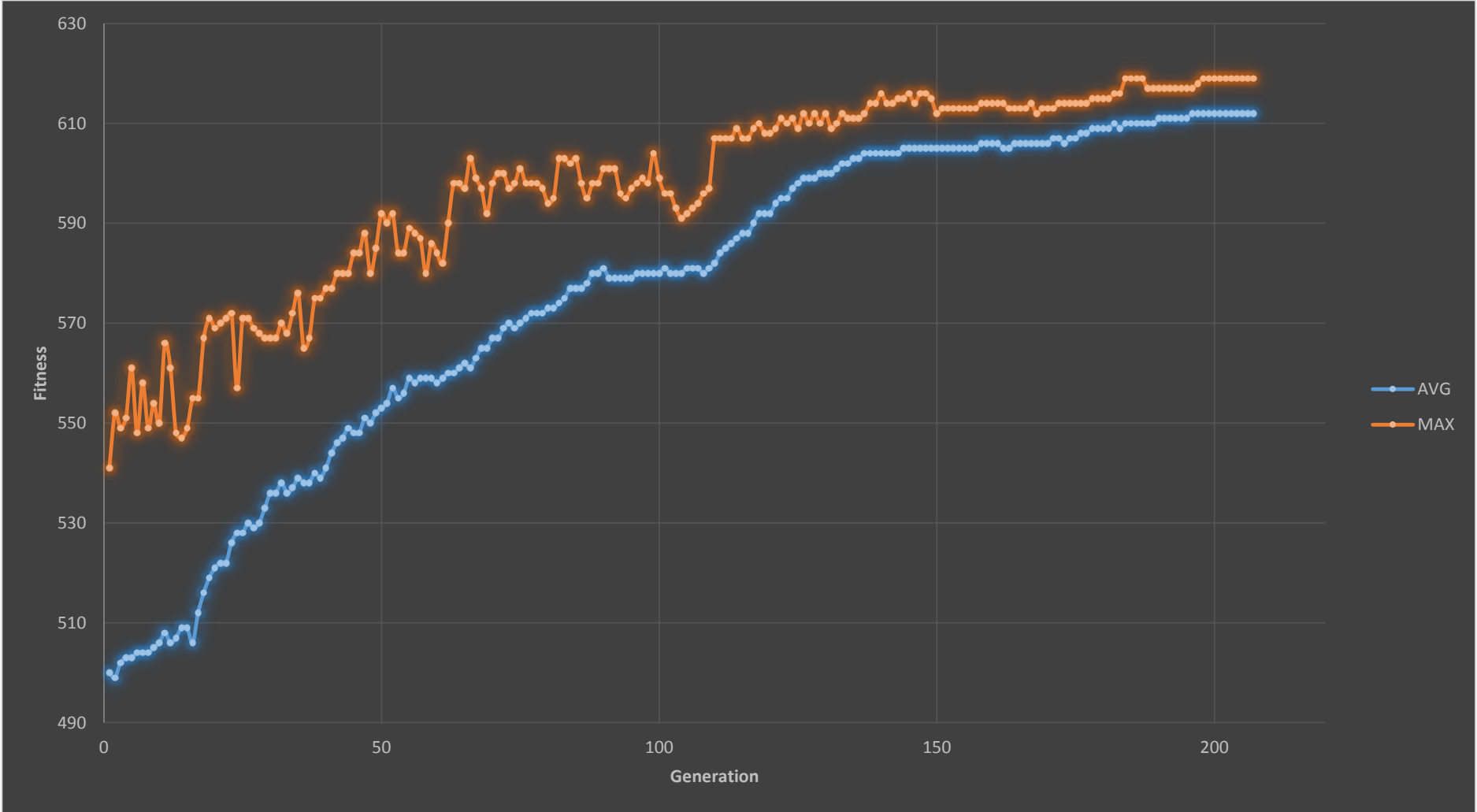
    if (maxFitnessList.Count > checkLast)
    {
        int c = 0;

        for (int i = 1; i < checkLast + 1; i++)
        {
            if (maxFitnessList[maxFitnessList.Count - i] > maxFitnessList.Last() -
(Constants.chromosomCountInPopulation * 0.01) && maxFitnessList[maxFitnessList.Count - i] <
maxFitnessList.Last() + (Constants.chromosomCountInPopulation * 0.01))
            {
                c++;
            }
        }

        return c == checkLast;
    }

    return false;
}
}
```

## Graphic Fitness(Generation)



## Table



| Generation | 0       |                            |                | 100     |                            |                | 200     |                            |                | 207 last |                            |                |
|------------|---------|----------------------------|----------------|---------|----------------------------|----------------|---------|----------------------------|----------------|----------|----------------------------|----------------|
| Chromosom  | Fitness | Probability to be selected | Selected count | Fitness | Probability to be selected | Selected count | Fitness | Probability to be selected | Selected count | Fitness  | Probability to be selected | Selected count |
| 1          | 552     | 0,011                      | 2              | 596     | 0,0103                     | 0              | 619     | 0,0101                     | 1              | 619      | 0                          | 0              |
| 2          | 540     | 0,0108                     | 1              | 594     | 0,0102                     | 1              | 617     | 0,0101                     | 1              | 619      | 0                          | 0              |
| 3          | 538     | 0,0108                     | 1              | 593     | 0,0102                     | 0              | 617     | 0,0101                     | 1              | 619      | 0                          | 0              |
| 4          | 537     | 0,0107                     | 1              | 592     | 0,0102                     | 1              | 617     | 0,0101                     | 3              | 618      | 0                          | 0              |
| 5          | 528     | 0,0106                     | 1              | 592     | 0,0102                     | 0              | 617     | 0,0101                     | 0              | 617      | 0                          | 0              |
| 6          | 527     | 0,0105                     | 0              | 592     | 0,0102                     | 1              | 617     | 0,0101                     | 1              | 617      | 0                          | 0              |
| 7          | 526     | 0,0105                     | 1              | 590     | 0,0102                     | 1              | 617     | 0,0101                     | 0              | 617      | 0                          | 0              |
| 8          | 525     | 0,0105                     | 1              | 590     | 0,0102                     | 0              | 616     | 0,0101                     | 0              | 617      | 0                          | 0              |
| 9          | 524     | 0,0105                     | 1              | 590     | 0,0102                     | 0              | 616     | 0,0101                     | 0              | 617      | 0                          | 0              |
| 10         | 522     | 0,0104                     | 3              | 588     | 0,0101                     | 1              | 616     | 0,0101                     | 0              | 616      | 0                          | 0              |
| 11         | 522     | 0,0104                     | 2              | 588     | 0,0101                     | 1              | 616     | 0,0101                     | 2              | 616      | 0                          | 0              |
| 12         | 521     | 0,0104                     | 1              | 588     | 0,0101                     | 0              | 616     | 0,0101                     | 1              | 616      | 0                          | 0              |
| 13         | 520     | 0,0104                     | 1              | 588     | 0,0101                     | 4              | 616     | 0,0101                     | 1              | 616      | 0                          | 0              |
| 14         | 520     | 0,0104                     | 1              | 588     | 0,0101                     | 0              | 616     | 0,0101                     | 1              | 615      | 0                          | 0              |
| 15         | 519     | 0,0104                     | 1              | 587     | 0,0101                     | 0              | 615     | 0,01                       | 0              | 615      | 0                          | 0              |
| 16         | 518     | 0,0104                     | 2              | 586     | 0,0101                     | 1              | 615     | 0,01                       | 0              | 615      | 0                          | 0              |
| 17         | 517     | 0,0103                     | 0              | 586     | 0,0101                     | 2              | 615     | 0,01                       | 0              | 615      | 0                          | 0              |
| 18         | 515     | 0,0103                     | 0              | 586     | 0,0101                     | 2              | 615     | 0,01                       | 1              | 615      | 0                          | 0              |
| 19         | 514     | 0,0103                     | 2              | 586     | 0,0101                     | 2              | 615     | 0,01                       | 0              | 615      | 0                          | 0              |
| 20         | 514     | 0,0103                     | 1              | 585     | 0,0101                     | 4              | 615     | 0,01                       | 2              | 615      | 0                          | 0              |
| 21         | 513     | 0,0103                     | 4              | 585     | 0,0101                     | 0              | 615     | 0,01                       | 1              | 615      | 0                          | 0              |
| 22         | 513     | 0,0103                     | 1              | 585     | 0,0101                     | 1              | 615     | 0,01                       | 0              | 615      | 0                          | 0              |
| 23         | 513     | 0,0103                     | 0              | 585     | 0,0101                     | 1              | 615     | 0,01                       | 0              | 615      | 0                          | 0              |
| 24         | 512     | 0,0102                     | 2              | 584     | 0,0101                     | 2              | 615     | 0,01                       | 1              | 614      | 0                          | 0              |
| 25         | 512     | 0,0102                     | 2              | 584     | 0,0101                     | 1              | 615     | 0,01                       | 2              | 614      | 0                          | 0              |
| 26         | 512     | 0,0102                     | 1              | 583     | 0,01                       | 1              | 615     | 0,01                       | 0              | 614      | 0                          | 0              |
| 27         | 512     | 0,0102                     | 0              | 583     | 0,01                       | 1              | 614     | 0,01                       | 2              | 614      | 0                          | 0              |
| 28         | 512     | 0,0102                     | 0              | 583     | 0,01                       | 1              | 614     | 0,01                       | 2              | 614      | 0                          | 0              |
| 29         | 511     | 0,0102                     | 2              | 583     | 0,01                       | 0              | 614     | 0,01                       | 0              | 614      | 0                          | 0              |
| 30         | 510     | 0,0102                     | 2              | 583     | 0,01                       | 1              | 614     | 0,01                       | 0              | 614      | 0                          | 0              |
| 31         | 510     | 0,0102                     | 1              | 583     | 0,01                       | 1              | 614     | 0,01                       | 1              | 614      | 0                          | 0              |
| 32         | 510     | 0,0102                     | 1              | 582     | 0,01                       | 1              | 614     | 0,01                       | 2              | 614      | 0                          | 0              |
| 33         | 510     | 0,0102                     | 1              | 582     | 0,01                       | 0              | 614     | 0,01                       | 0              | 614      | 0                          | 0              |
| 34         | 509     | 0,0102                     | 0              | 582     | 0,01                       | 2              | 614     | 0,01                       | 1              | 614      | 0                          | 0              |
| 35         | 508     | 0,0102                     | 2              | 582     | 0,01                       | 1              | 614     | 0,01                       | 0              | 614      | 0                          | 0              |
| 36         | 507     | 0,0101                     | 0              | 582     | 0,01                       | 3              | 614     | 0,01                       | 1              | 614      | 0                          | 0              |
| 37         | 506     | 0,0101                     | 2              | 582     | 0,01                       | 1              | 614     | 0,01                       | 1              | 613      | 0                          | 0              |
| 38         | 506     | 0,0101                     | 2              | 582     | 0,01                       | 0              | 614     | 0,01                       | 2              | 613      | 0                          | 0              |
| 39         | 505     | 0,0101                     | 3              | 582     | 0,01                       | 1              | 613     | 0,01                       | 0              | 613      | 0                          | 0              |
| 40         | 505     | 0,0101                     | 2              | 581     | 0,01                       | 0              | 613     | 0,01                       | 0              | 613      | 0                          | 0              |
| 41         | 505     | 0,0101                     | 1              | 581     | 0,01                       | 2              | 613     | 0,01                       | 2              | 613      | 0                          | 0              |
| 42         | 505     | 0,0101                     | 0              | 581     | 0,01                       | 3              | 613     | 0,01                       | 0              | 613      | 0                          | 0              |
| 43         | 504     | 0,0101                     | 2              | 581     | 0,01                       | 2              | 613     | 0,01                       | 3              | 613      | 0                          | 0              |
| 44         | 503     | 0,0101                     | 1              | 581     | 0,01                       | 1              | 613     | 0,01                       | 1              | 613      | 0                          | 0              |
| 45         | 501     | 0,01                       | 2              | 581     | 0,01                       | 1              | 613     | 0,01                       | 2              | 613      | 0                          | 0              |
| 46         | 501     | 0,01                       | 1              | 581     | 0,01                       | 1              | 613     | 0,01                       | 1              | 613      | 0                          | 0              |
| 47         | 501     | 0,01                       | 1              | 581     | 0,01                       | 1              | 613     | 0,01                       | 0              | 612      | 0                          | 0              |
| 48         | 500     | 0,01                       | 2              | 581     | 0,01                       | 3              | 613     | 0,01                       | 1              | 612      | 0                          | 0              |
| 49         | 500     | 0,01                       | 0              | 580     | 0,01                       | 0              | 613     | 0,01                       | 0              | 612      | 0                          | 0              |
| 50         | 499     | 0,01                       | 2              | 580     | 0,01                       | 0              | 613     | 0,01                       | 2              | 612      | 0                          | 0              |
| 51         | 499     | 0,01                       | 1              | 580     | 0,01                       | 0              | 613     | 0,01                       | 1              | 612      | 0                          | 0              |
| 52         | 499     | 0,01                       | 0              | 580     | 0,01                       | 0              | 613     | 0,01                       | 0              | 612      | 0                          | 0              |
| 53         | 499     | 0,01                       | 0              | 579     | 0,01                       | 3              | 613     | 0,01                       | 2              | 612      | 0                          | 0              |
| 54         | 498     | 0,01                       | 2              | 579     | 0,01                       | 1              | 613     | 0,01                       | 0              | 612      | 0                          | 0              |
| 55         | 498     | 0,01                       | 1              | 579     | 0,01                       | 2              | 613     | 0,01                       | 1              | 612      | 0                          | 0              |
| 56         | 498     | 0,01                       | 1              | 579     | 0,01                       | 0              | 612     | 0,01                       | 1              | 612      | 0                          | 0              |
| 57         | 498     | 0,01                       | 0              | 579     | 0,01                       | 0              | 612     | 0,01                       | 1              | 612      | 0                          | 0              |
| 58         | 497     | 0,0099                     | 1              | 579     | 0,01                       | 2              | 612     | 0,01                       | 2              | 612      | 0                          | 0              |
| 59         | 497     | 0,0099                     | 0              | 579     | 0,01                       | 2              | 612     | 0,01                       | 0              | 612      | 0                          | 0              |
| 60         | 496     | 0,0099                     | 1              | 579     | 0,01                       | 2              | 612     | 0,01                       | 3              | 612      | 0                          | 0              |
| 61         | 495     | 0,0099                     | 1              | 579     | 0,01                       | 0              | 612     | 0,01                       | 1              | 612      | 0                          | 0              |
| 62         | 495     | 0,0099                     | 0              | 579     | 0,01                       | 0              | 612     | 0,01                       | 2              | 612      | 0                          | 0              |
| 63         | 494     | 0,0099                     | 1              | 578     | 0,01                       | 1              | 612     | 0,01                       | 0              | 611      | 0                          | 0              |
| 64         | 494     | 0,0099                     | 0              | 578     | 0,01                       | 0              | 612     | 0,01                       | 1              | 611      | 0                          | 0              |
| 65         | 493     | 0,0099                     | 0              | 578     | 0,01                       | 1              | 612     | 0,01                       | 4              | 611      | 0                          | 0              |

|     |     |        |   |     |        |   |     |        |   |     |   |   |
|-----|-----|--------|---|-----|--------|---|-----|--------|---|-----|---|---|
| 66  | 493 | 0,0099 | 0 | 578 | 0,01   | 3 | 612 | 0,01   | 1 | 611 | 0 | 0 |
| 67  | 492 | 0,0098 | 2 | 578 | 0,01   | 0 | 611 | 0,01   | 0 | 611 | 0 | 0 |
| 68  | 492 | 0,0098 | 2 | 578 | 0,01   | 2 | 611 | 0,01   | 1 | 611 | 0 | 0 |
| 69  | 492 | 0,0098 | 1 | 578 | 0,01   | 1 | 611 | 0,01   | 0 | 611 | 0 | 0 |
| 70  | 491 | 0,0098 | 2 | 577 | 0,0099 | 1 | 611 | 0,01   | 1 | 611 | 0 | 0 |
| 71  | 491 | 0,0098 | 1 | 577 | 0,0099 | 0 | 611 | 0,01   | 2 | 611 | 0 | 0 |
| 72  | 491 | 0,0098 | 0 | 577 | 0,0099 | 1 | 611 | 0,01   | 3 | 611 | 0 | 0 |
| 73  | 491 | 0,0098 | 0 | 577 | 0,0099 | 0 | 611 | 0,01   | 2 | 611 | 0 | 0 |
| 74  | 490 | 0,0098 | 0 | 577 | 0,0099 | 1 | 611 | 0,01   | 6 | 611 | 0 | 0 |
| 75  | 489 | 0,0098 | 0 | 577 | 0,0099 | 2 | 611 | 0,01   | 0 | 611 | 0 | 0 |
| 76  | 488 | 0,0098 | 1 | 577 | 0,0099 | 1 | 611 | 0,01   | 1 | 611 | 0 | 0 |
| 77  | 488 | 0,0098 | 0 | 577 | 0,0099 | 1 | 611 | 0,01   | 0 | 611 | 0 | 0 |
| 78  | 488 | 0,0098 | 0 | 577 | 0,0099 | 1 | 611 | 0,01   | 0 | 611 | 0 | 0 |
| 79  | 487 | 0,0097 | 0 | 576 | 0,0099 | 0 | 611 | 0,01   | 2 | 611 | 0 | 0 |
| 80  | 486 | 0,0097 | 2 | 576 | 0,0099 | 1 | 611 | 0,01   | 0 | 611 | 0 | 0 |
| 81  | 485 | 0,0097 | 1 | 575 | 0,0099 | 0 | 610 | 0,01   | 2 | 611 | 0 | 0 |
| 82  | 485 | 0,0097 | 1 | 575 | 0,0099 | 2 | 610 | 0,01   | 1 | 611 | 0 | 0 |
| 83  | 485 | 0,0097 | 1 | 575 | 0,0099 | 3 | 610 | 0,01   | 1 | 611 | 0 | 0 |
| 84  | 483 | 0,0097 | 1 | 575 | 0,0099 | 0 | 610 | 0,01   | 0 | 611 | 0 | 0 |
| 85  | 482 | 0,0096 | 1 | 575 | 0,0099 | 1 | 610 | 0,01   | 0 | 611 | 0 | 0 |
| 86  | 481 | 0,0096 | 0 | 575 | 0,0099 | 0 | 610 | 0,01   | 3 | 611 | 0 | 0 |
| 87  | 478 | 0,0096 | 1 | 575 | 0,0099 | 0 | 610 | 0,01   | 0 | 610 | 0 | 0 |
| 88  | 475 | 0,0095 | 1 | 575 | 0,0099 | 1 | 610 | 0,01   | 2 | 610 | 0 | 0 |
| 89  | 475 | 0,0095 | 0 | 575 | 0,0099 | 1 | 610 | 0,01   | 1 | 610 | 0 | 0 |
| 90  | 474 | 0,0095 | 1 | 574 | 0,0099 | 1 | 609 | 0,0099 | 2 | 610 | 0 | 0 |
| 91  | 474 | 0,0095 | 1 | 574 | 0,0099 | 1 | 609 | 0,0099 | 0 | 610 | 0 | 0 |
| 92  | 472 | 0,0094 | 1 | 574 | 0,0099 | 0 | 609 | 0,0099 | 2 | 610 | 0 | 0 |
| 93  | 471 | 0,0094 | 1 | 574 | 0,0099 | 3 | 609 | 0,0099 | 0 | 610 | 0 | 0 |
| 94  | 469 | 0,0094 | 1 | 573 | 0,0099 | 0 | 609 | 0,0099 | 0 | 610 | 0 | 0 |
| 95  | 469 | 0,0094 | 0 | 573 | 0,0099 | 1 | 608 | 0,0099 | 1 | 609 | 0 | 0 |
| 96  | 468 | 0,0094 | 1 | 572 | 0,0099 | 0 | 607 | 0,0099 | 0 | 609 | 0 | 0 |
| 97  | 466 | 0,0093 | 0 | 570 | 0,0098 | 1 | 607 | 0,0099 | 1 | 609 | 0 | 0 |
| 98  | 465 | 0,0093 | 0 | 570 | 0,0098 | 2 | 607 | 0,0099 | 1 | 609 | 0 | 0 |
| 99  | 458 | 0,0092 | 4 | 570 | 0,0098 | 0 | 607 | 0,0099 | 2 | 609 | 0 | 0 |
| 100 | 451 | 0,009  | 0 | 564 | 0,0097 | 0 | 607 | 0,0099 | 0 | 608 | 0 | 0 |