Student Id: 9910821

S	d_1	d_2	$\prod(S,t)$
3000	0.275255	-2.8384	4434.82
3200	0.557819	-2.55584	5514.88
3400	0.826603	-2.28705	6520.34
3600	1.08228	-2.03138	7407.06
3800	1.32548	-1.78817	8154.1
4000	1.55683	-1.55683	8758.49
4200	1.77689	-1.33677	9229.19
4400	1.98622	-1.12744	9582.28
4600	2.18534	-0.928318	9837.71
4800	2.37475	-0.738909	10017.1
5000	2.55492	-0.558738	10142.1

Table 1: A table showing different values for the financial contract $\prod(S,t)$ for different values of S.

```
#include <iostream>
#include <fstream>
#include <cmath>
using namespace std;

double normalDistribution(double x)
{
    // calculate \sqrt{2\pi} upfront once
    static const double RT2PI = sqrt(4.0*acos(0.0));
    // calculate 10/\sqrt{2} upfront once
    static const double SPLIT = 10. / sqrt(2);
    static const double a[] = { 220.206867912376, 221.213596169931,
   112.079291497871, 33.912866078383, 6.37396220353165, 0.700383064443688,
   3.52624965998911e-02 };
    static const double b[] = { 440.413735824752, 793.826512519948,
   637.333633378831, 296.564248779674, 86.7807322029461, 16.064177579207,
   1.75566716318264, 8.83883476483184e-02 };

    const double z = fabs(x);
    // Now N(x) = 1 - N(-x) = 1-\sqrt{2\pi}N'(x)\frac{P(x)}{Q(x)}
    //  so N(-x) = \sqrt{2\pi}N'(x)\frac{P(x)}{Q(x)}
    // now let \sqrt{2\pi}N'(z)\frac{P(x)}{Q(z)} = Nz
    // Therefore we have
    //      Nxm = N(x) = \sqrt{2\pi}N'(z)\frac{P(x)}{Q(z)} = Nz if x<0
    //      Nxp = N(x) = 1 - \sqrt{2\pi}N'(z)\frac{P(x)}{Q(z)} = 1-Nz if x
   >=0
    double Nz = 0.0;

    // if z outside these limits then value effectively 0 or 1 for machine
   precision
    if (z <= 37.0)
    {
        // NDash = N'(z) * sqrt{2\pi}
```

```
28          const double NDash = exp(-z*z / 2.0) / RT2PI;
29          if (z<SPLIT)
30          {
31              // here Pz = P(z) is a polynomial
32              const double Pz = (((((a[6] * z + a[5])*z + a[4])*z + a[3])*z +
     a[2])*z + a[1])*z + a[0];
33              // and Qz = Q(z) is a polynomial
34              const double Qz = ((((((b[7] * z + b[6])*z + b[5])*z + b[4])*z
    + b[3])*z + b[2])*z + b[1])*z + b[0];
35              // use polynomials to calculate N(z)  = \sqrt{2\pi}N'(x)\frac{P
    (x)}{Q(x)}
36              Nz = RT2PI*NDash*Pz / Qz;
37          }
38          else
39          {
40              // implement recurrence relation on F_4(z)
41              const double F4z = z + 1.0 / (z + 2.0 / (z + 3.0 / (z + 4.0 / (
    z + 13.0 / 20.0))));
42              // use polynomials to calculate N(z), note here that Nz = N' /
    F
43              Nz = NDash / F4z;
44          }
45      }
46
47      //
48      return x >= 0.0 ? 1 - Nz : Nz;
49  }
50
51
52  int main()
53  {
54      //Initial conditions
55      double S = 2800, X = 4000, r = 0.0472, q = 0.0269,
56          sigma =  0.2481, T = 1., t = 0.;
57      ofstream output("./Assignment_1/test.csv");
58
59      for (double i = 0; S < 5000; i++)
60      {
61          S += 200;
62          //& For easy import in latex later
63          output << S << " & ";
64          //Formulae for d1, d2
65          double d1 = (1-exp(1-(S/X))+sigma*exp((r-q)/(pow(sigma,2.0)))*sqrt(
    T-t))/log(1+sigma*sqrt(T-t));
66          double d2 = (1-exp(1-(S/X))-sigma*exp((r-q)/(pow(sigma,2.0)))*sqrt(
    T-t))/log(1+sigma*sqrt(T-t));
67          //Formula for contract
68          output << d1 << " & ";
69          output << d2 << " & ";
70          output << " " << S*exp(-r*(T - t))*exp(1-(pow(sigma,2.0)*(T - t)))*
     normalDistribution(d1)
71              - X*exp(-q*(T - t))*exp(q/r)*normalDistribution(d2) << "\\\\"
    << endl;
72      }
73      return 0;
74  }
```