SPX _ S&P 500

- 資料定義:

無風險利率 r : 3.82% (取美國10年期公債利率)
股價s : 4346.3 (取期貨價格)
資料長度 length : 313

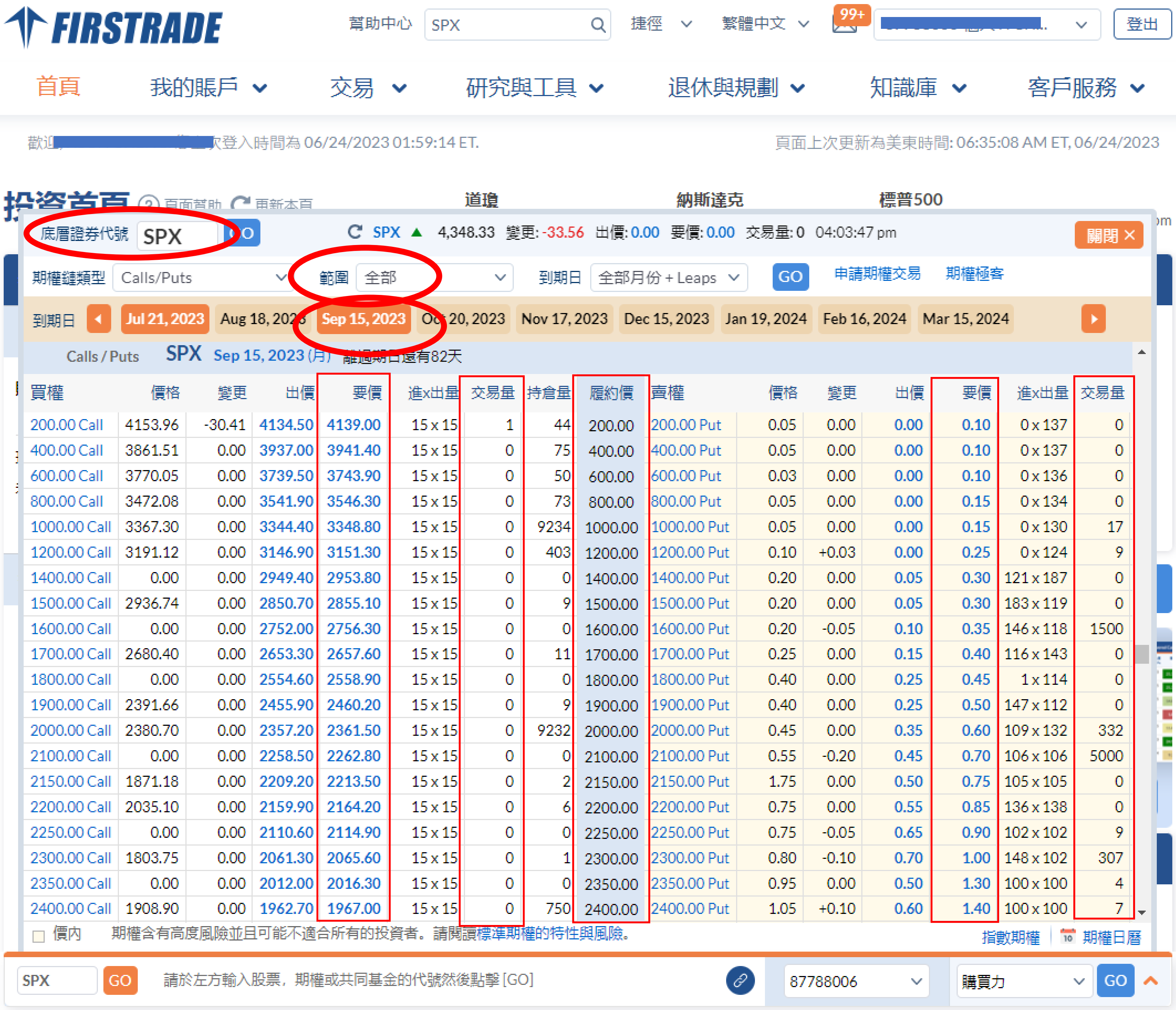
- S&P500 資料 :

到期日 : 9/15
資料日期: 6/23
期權資料 : first trade 帳號獲取, 代號(SPX) #資料長度 = 313
期貨資料 : investing.com
輸入資料格式(表頭) : (strike_price) (call_price) (call_volume) (put_price) (put_volume)
(從上到下 <-> 履約價 高到低)
(價格一律取 *要價*)

程式資料輸入順序：

- Length = 313
- r = 3.82
- S = 4346.3
- (strike_price) (call_price) (call_volume) (put_price) (put_volume)

S&P資料來源



[redacted]

歡迎[redacted]次登入時間為 06/24/2023 01:59:14 ET.

頁面上次更新為美東時間: 06:35:08 AM ET, 06/24/2023

投資首頁 ⓘ 頁面幫助 ↻ 更新本頁

道瓊 納斯達克 標普500

底層證券代號 SPX GO

↻ SPX ▲ 4,348.33 變更: -33.56 出價: 0.00 要價: 0.00 交易量: 0 04:03:47 pm 關閉 ✕

期權鏈類型 Calls/Puts 範圍 全部 到期日 全部月份 + Leaps GO 申請期權交易 期權極客

到期日 ◀ Jul 21, 2023 Aug 18, 2023 Sep 15, 2023 Oct 20, 2023 Nov 17, 2023 Dec 15, 2023 Jan 19, 2024 Feb 16, 2024 Mar 15, 2024 ▶

Calls / Puts SPX Sep 15, 2023 (月) 離到期日還有82天

買權	價格	變更	出價	要價	進x出量	交易量	持倉量	履約價	賣權	價格	變更	出價	要價	進x出量	交易量
200.00 Call	4153.96	-30.41	4134.50	4139.00	15 x 15	1	44	200.00	200.00 Put	0.05	0.00	0.00	0.10	0 x 137	0
400.00 Call	3861.51	0.00	3937.00	3941.40	15 x 15	0	75	400.00	400.00 Put	0.05	0.00	0.00	0.10	0 x 137	0
600.00 Call	3770.05	0.00	3739.50	3743.90	15 x 15	0	50	600.00	600.00 Put	0.03	0.00	0.00	0.10	0 x 136	0
800.00 Call	3472.08	0.00	3541.90	3546.30	15 x 15	0	73	800.00	800.00 Put	0.05	0.00	0.00	0.15	0 x 134	0
1000.00 Call	3367.30	0.00	3344.40	3348.80	15 x 15	0	9234	1000.00	1000.00 Put	0.05	0.00	0.00	0.15	0 x 130	17
1200.00 Call	3191.12	0.00	3146.90	3151.30	15 x 15	0	403	1200.00	1200.00 Put	0.10	+0.03	0.00	0.25	0 x 124	9
1400.00 Call	0.00	0.00	2949.40	2953.80	15 x 15	0	0	1400.00	1400.00 Put	0.20	0.00	0.05	0.30	121 x 187	0
1500.00 Call	2936.74	0.00	2850.70	2855.10	15 x 15	0	9	1500.00	1500.00 Put	0.20	0.00	0.05	0.30	183 x 119	0
1600.00 Call	0.00	0.00	2752.00	2756.30	15 x 15	0	0	1600.00	1600.00 Put	0.20	-0.05	0.10	0.35	146 x 118	1500
1700.00 Call	2680.40	0.00	2653.30	2657.60	15 x 15	0	11	1700.00	1700.00 Put	0.25	0.00	0.15	0.40	116 x 143	0
1800.00 Call	0.00	0.00	2554.60	2558.90	15 x 15	0	0	1800.00	1800.00 Put	0.40	0.00	0.25	0.45	1 x 114	0
1900.00 Call	2391.66	0.00	2455.90	2460.20	15 x 15	0	9	1900.00	1900.00 Put	0.40	0.00	0.25	0.50	147 x 112	0
2000.00 Call	2380.70	0.00	2357.20	2361.50	15 x 15	0	9232	2000.00	2000.00 Put	0.45	0.00	0.35	0.60	109 x 132	332
2100.00 Call	0.00	0.00	2258.50	2262.80	15 x 15	0	0	2100.00	2100.00 Put	0.55	-0.20	0.45	0.70	106 x 106	5000
2150.00 Call	1871.18	0.00	2209.20	2213.50	15 x 15	0	2	2150.00	2150.00 Put	1.75	0.00	0.50	0.75	105 x 105	0
2200.00 Call	2035.10	0.00	2159.90	2164.20	15 x 15	0	6	2200.00	2200.00 Put	0.75	0.00	0.55	0.85	136 x 138	0
2250.00 Call	0.00	0.00	2110.60	2114.90	15 x 15	0	0	2250.00	2250.00 Put	0.75	-0.05	0.65	0.90	102 x 102	9
2300.00 Call	1803.75	0.00	2061.30	2065.60	15 x 15	0	1	2300.00	2300.00 Put	0.80	-0.10	0.70	1.00	148 x 102	307
2350.00 Call	0.00	0.00	2012.00	2016.30	15 x 15	0	0	2350.00	2350.00 Put	0.95	0.00	0.50	1.30	100 x 100	4
2400.00 Call	1908.90	0.00	1962.70	1967.00	15 x 15	0	750	2400.00	2400.00 Put	1.05	+0.10	0.60	1.40	100 x 100	7

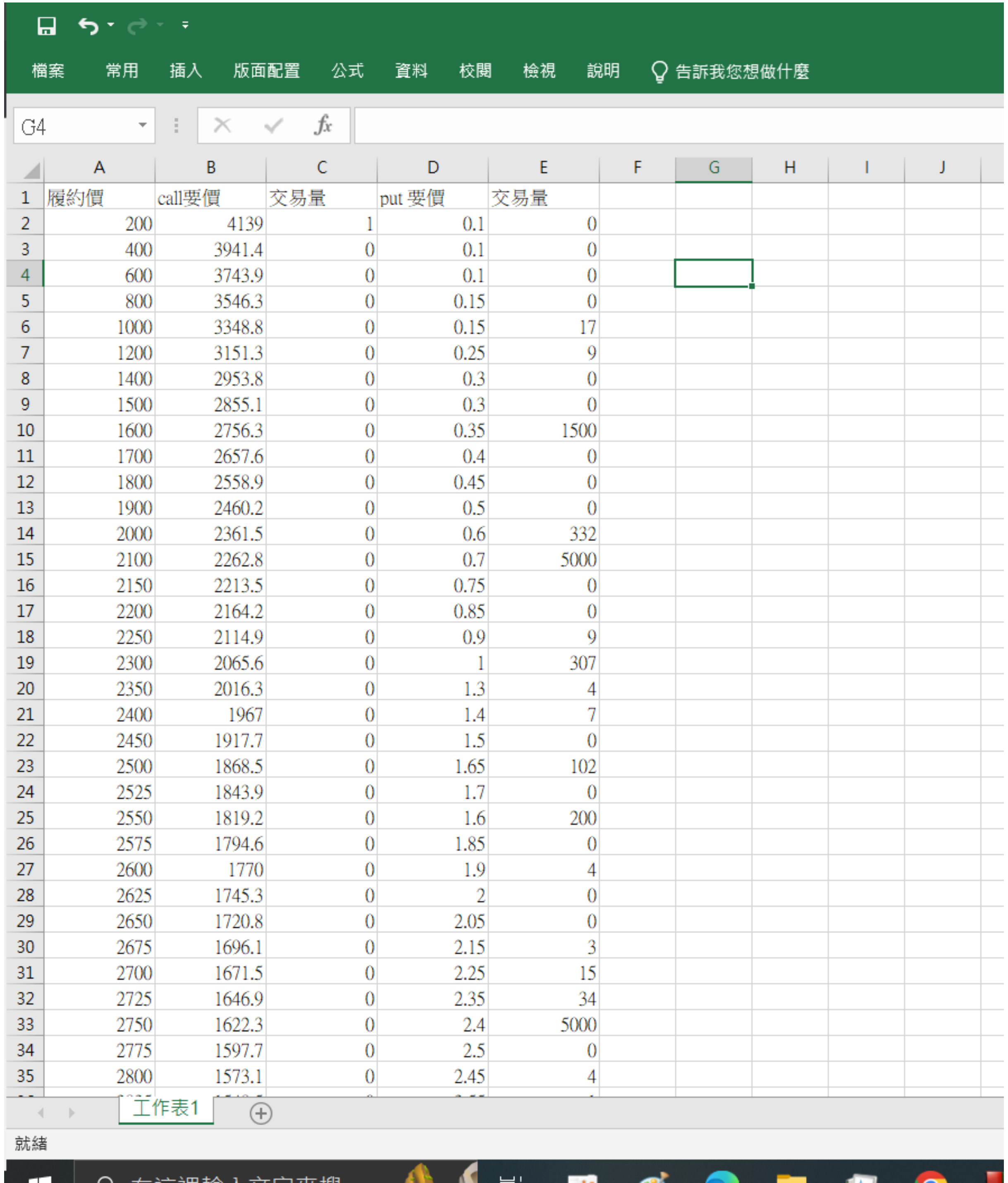
□ 價內 期權含有高度風險並且可能不適合所有的投資者。請閱讀標準期權的特性與風險。

指數期權 期權日曆

SPX GO 請於左方輸入股票，期權或共同基金的代號然後點擊 [GO]

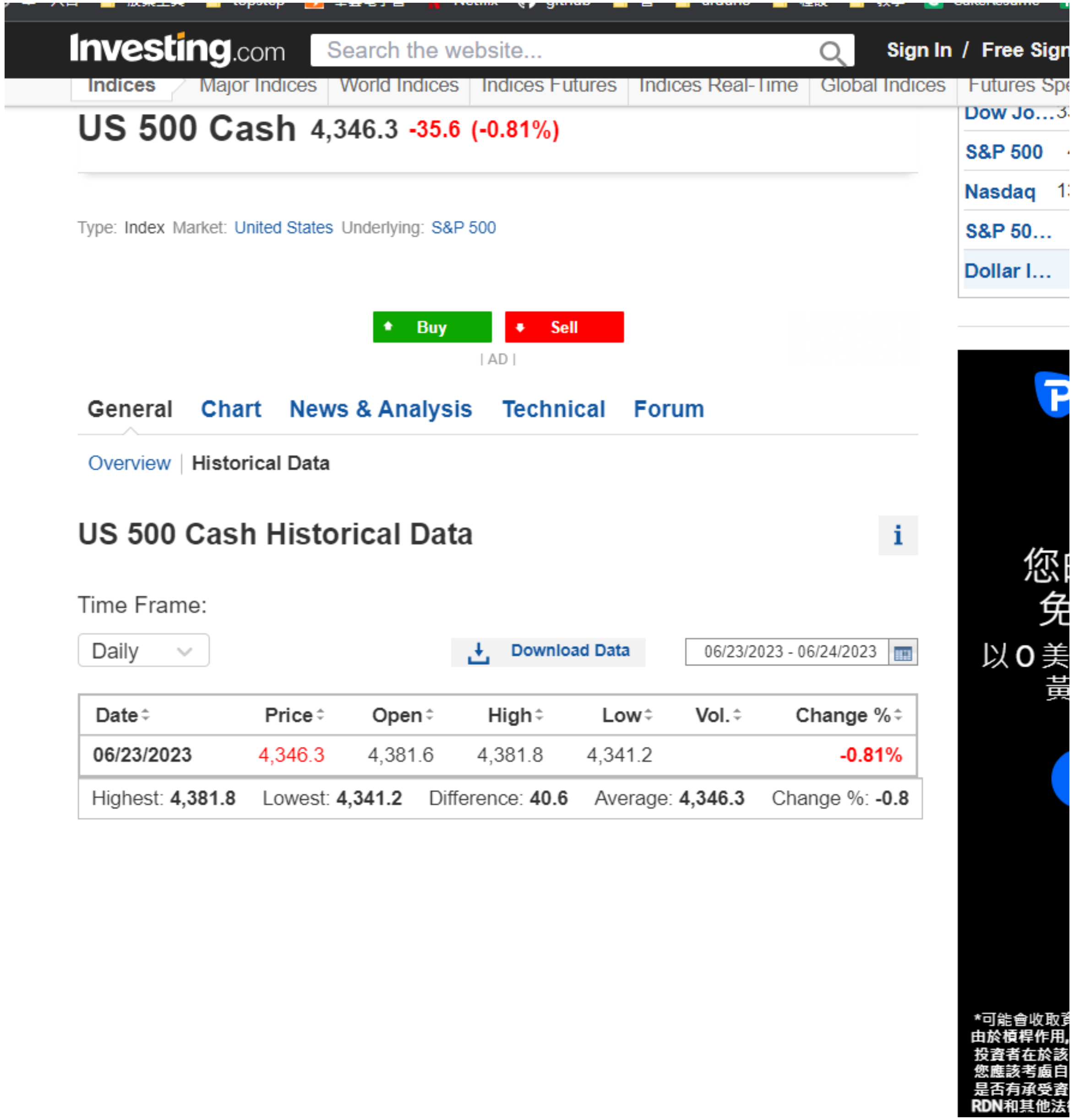
87788006 購買力 GO

整理後資料



履約價	call要價	交易量	put 要價	交易量
200	4139	1	0.1	0
400	3941.4	0	0.1	0
600	3743.9	0	0.1	0
800	3546.3	0	0.15	0
1000	3348.8	0	0.15	17
1200	3151.3	0	0.25	9
1400	2953.8	0	0.3	0
1500	2855.1	0	0.3	0
1600	2756.3	0	0.35	1500
1700	2657.6	0	0.4	0
1800	2558.9	0	0.45	0
1900	2460.2	0	0.5	0
2000	2361.5	0	0.6	332
2100	2262.8	0	0.7	5000
2150	2213.5	0	0.75	0
2200	2164.2	0	0.85	0
2250	2114.9	0	0.9	9
2300	2065.6	0	1	307
2350	2016.3	0	1.3	4
2400	1967	0	1.4	7
2450	1917.7	0	1.5	0
2500	1868.5	0	1.65	102
2525	1843.9	0	1.7	0
2550	1819.2	0	1.6	200
2575	1794.6	0	1.85	0
2600	1770	0	1.9	4
2625	1745.3	0	2	0
2650	1720.8	0	2.05	0
2675	1696.1	0	2.15	3
2700	1671.5	0	2.25	15
2725	1646.9	0	2.35	34
2750	1622.3	0	2.4	5000
2775	1597.7	0	2.5	0
2800	1573.1	0	2.45	4

Future



US 500 Cash 4,346.3 -35.6 (-0.81%)

Type: Index Market: United States Underlying: S&P 500

General Chart News & Analysis Technical Forum

Overview | Historical Data

US 500 Cash Historical Data

Time Frame:

Daily

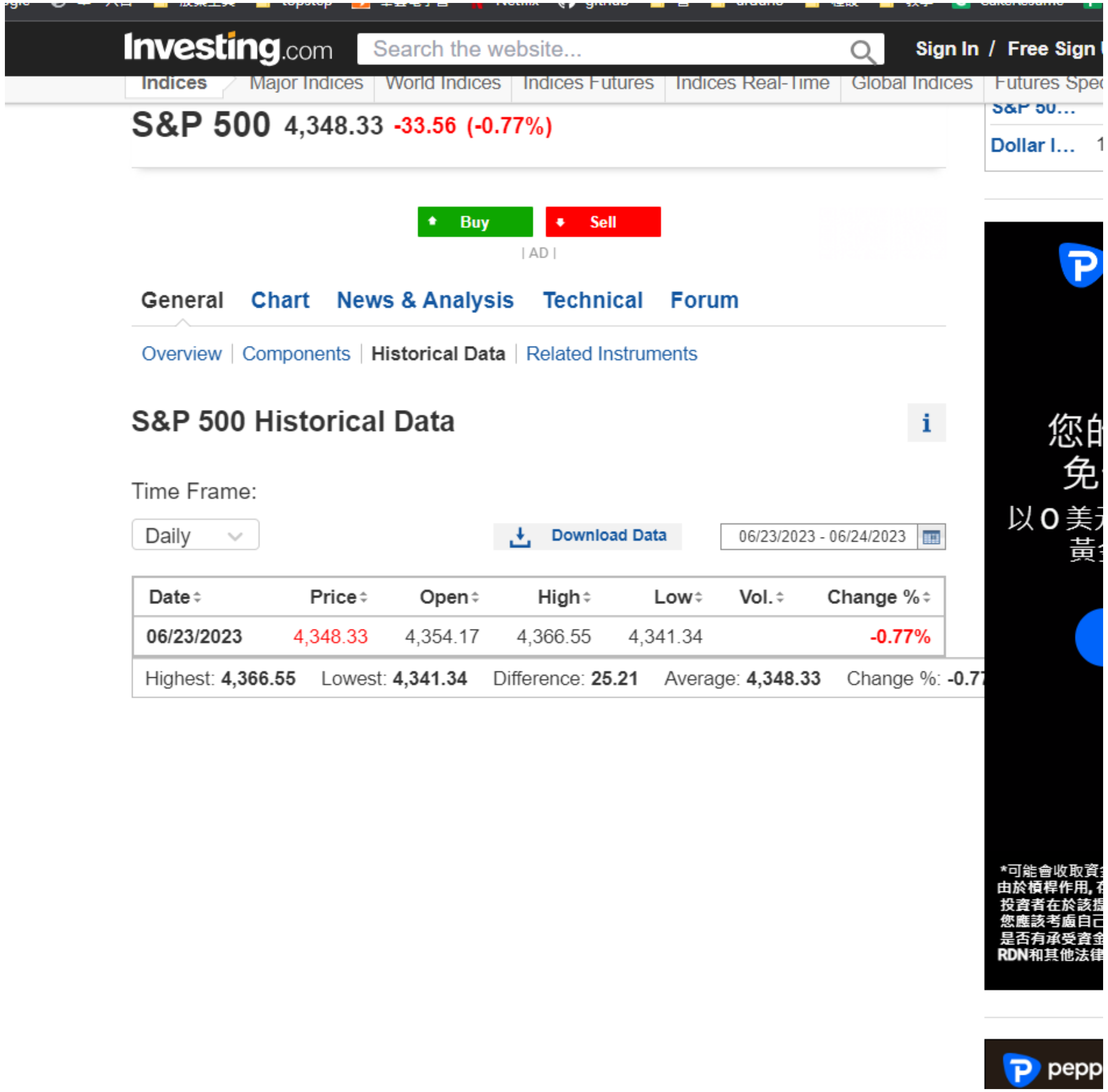
Download Data

06/23/2023 - 06/24/2023

Date	Price	Open	High	Low	Vol.	Change %
06/23/2023	4,346.3	4,381.6	4,381.8	4,341.2		-0.81%

Highest: 4,381.8 Lowest: 4,341.2 Difference: 40.6 Average: 4,346.3 Change %: -0.8

Underlying asset



S&P 500 4,348.33 -33.56 (-0.77%)

S&P 500 Historical Data

Time Frame:

Daily

06/23/2023 - 06/24/2023

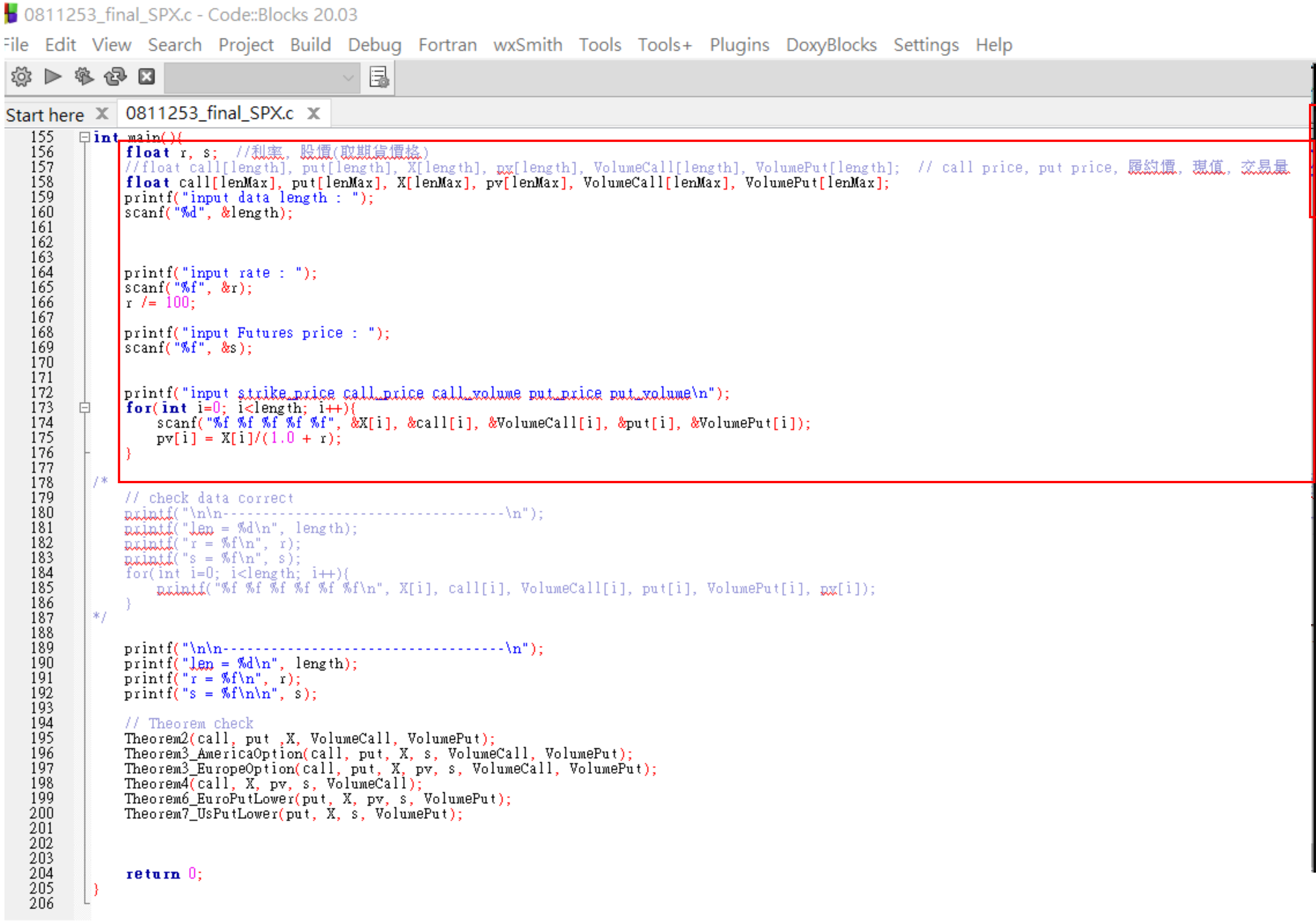
Date	Price	Open	High	Low	Vol.	Change %
06/23/2023	4,348.33	4,354.17	4,366.55	4,341.34		-0.77%

Highest: **4,366.55** Lowest: **4,341.34** Difference: **25.21** Average: **4,348.33** Change %: **-0.7**

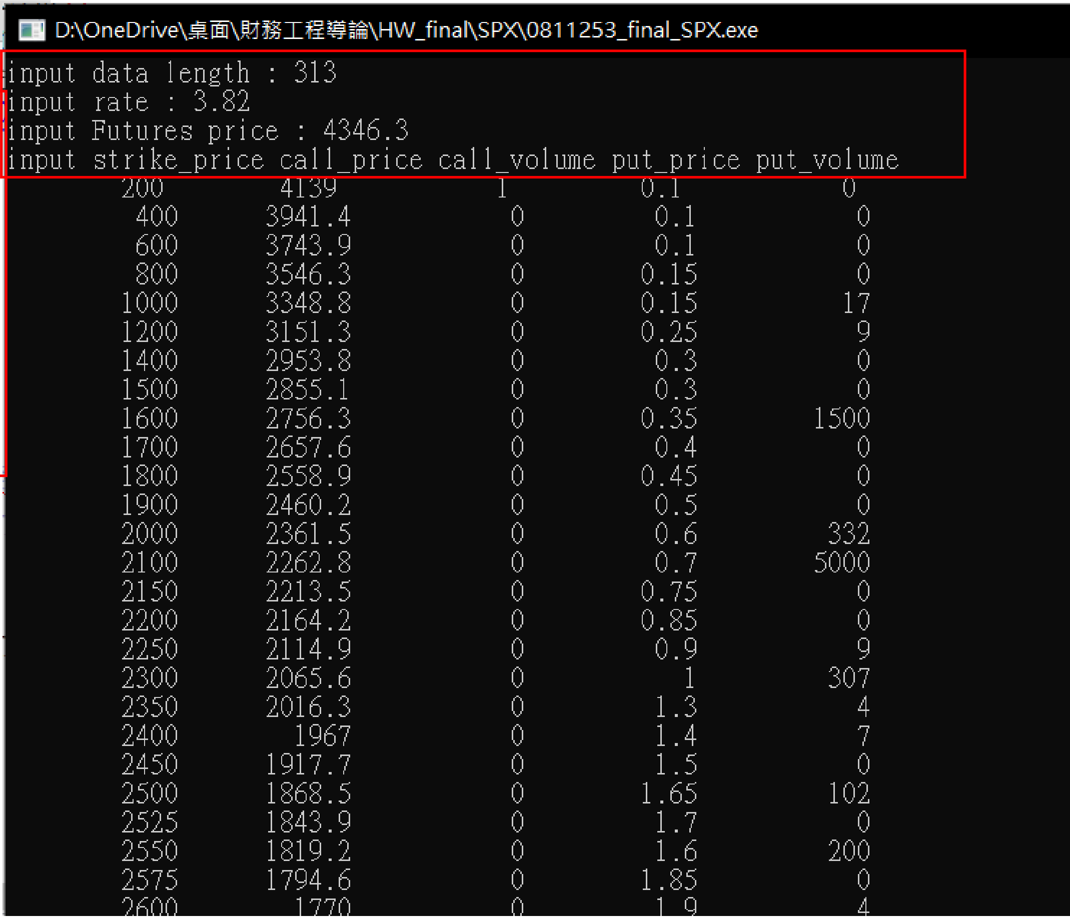
程式介紹

- 程式由老師的Theorem2、3、4、6、7條件製作成，每個子程式個跑一個Theorem
- main主程式負責導入資料與呼叫子程式
- Output : 包含履約價、期權價格、條件價格、成交量、套利空間

1. Main function (input)

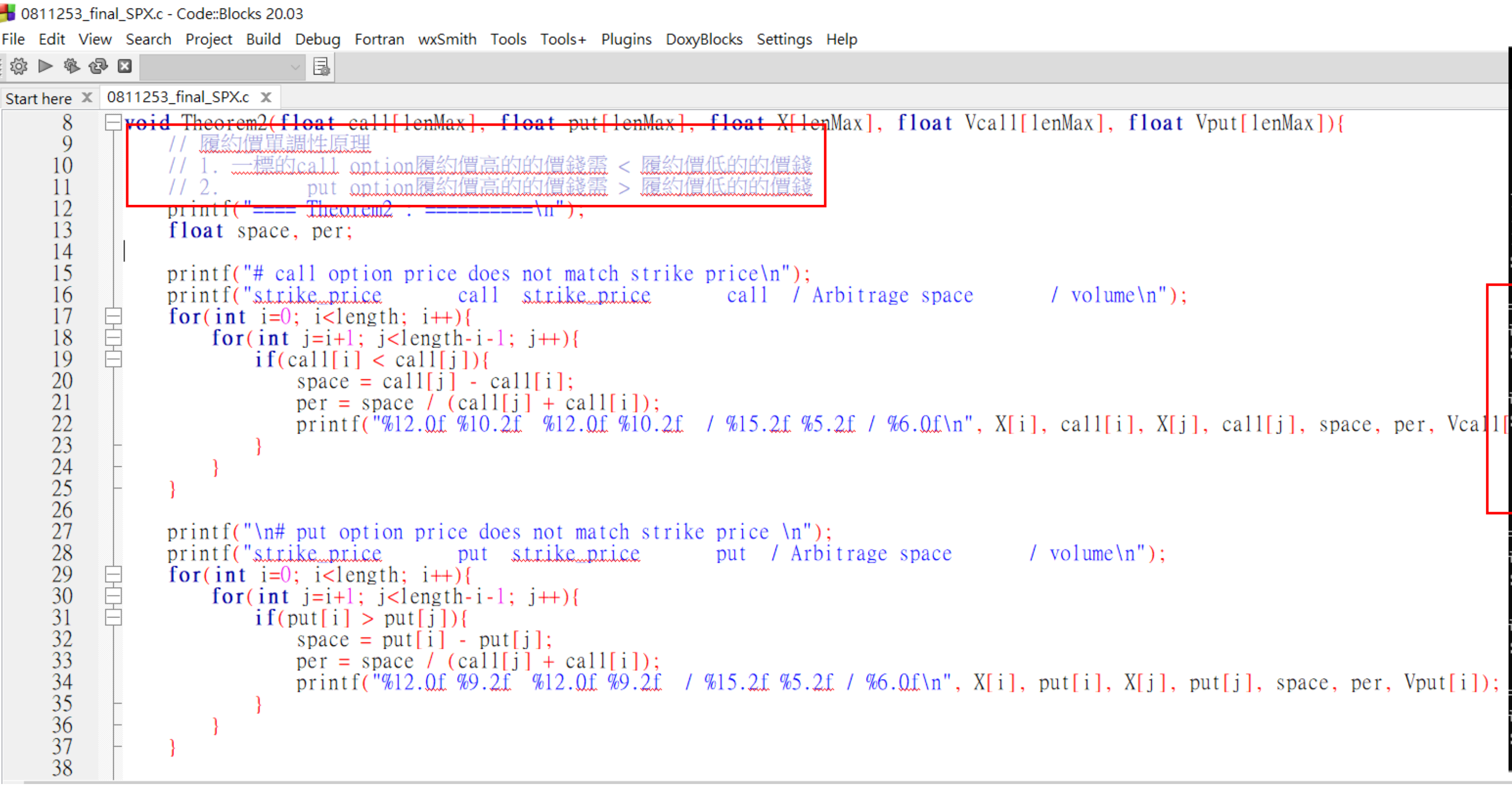

0811253_final_SPX.c - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Start here
0811253_final_SPX.c

```c
int main(){
    float r, s;  //利率, 股價(取期貨價格)
    //float call[length], put[length], X[length], px[length], VolumeCall[length], VolumePut[length];  // call price, put price, 履約價, 現值, 交易量
    float call[lenMax], put[lenMax], X[lenMax], pv[lenMax], VolumeCall[lenMax], VolumePut[lenMax];
    printf("input data length : ");
    scanf("%d", &length);



    printf("input rate : ");
    scanf("%f", &r);
    r /= 100;

    printf("input Futures price : ");
    scanf("%f", &s);


    printf("input strike_price call_price call_volume put_price put_volume\n");
    for(int i=0; i<length; i++){
        scanf("%f %f %f %f %f", &X[i], &call[i], &VolumeCall[i], &put[i], &VolumePut[i]);
        pv[i] = X[i]/(1.0 + r);
    }

/*
    // check data correct
    printf("\n\n----------------------------------\n");
    printf("len = %d\n", length);
    printf("r = %f\n", r);
    printf("s = %f\n", s);
    for(int i=0; i<length; i++){
        printf("%f %f %f %f %f %f\n", X[i], call[i], VolumeCall[i], put[i], VolumePut[i], px[i]);
    }
*/

    printf("\n\n----------------------------------\n");
    printf("len = %d\n", length);
    printf("r = %f\n", r);
    printf("s = %f\n\n", s);

    // Theorem check
    Theorem2(call, put ,X, VolumeCall, VolumePut);
    Theorem3_AmericaOption(call, put, X, s, VolumeCall, VolumePut);
    Theorem3_EuropeOption(call, put, X, pv, s, VolumeCall, VolumePut);
    Theorem4(call, X, pv, s, VolumeCall);
    Theorem6_EuroPutLower(put, X, pv, s, VolumePut);
    Theorem7_UsPutLower(put, X, s, VolumePut);



    return 0;
}
```


D:\OneDrive\桌面\財務工程導論\HW_final\SPX\0811253_final_SPX.exe

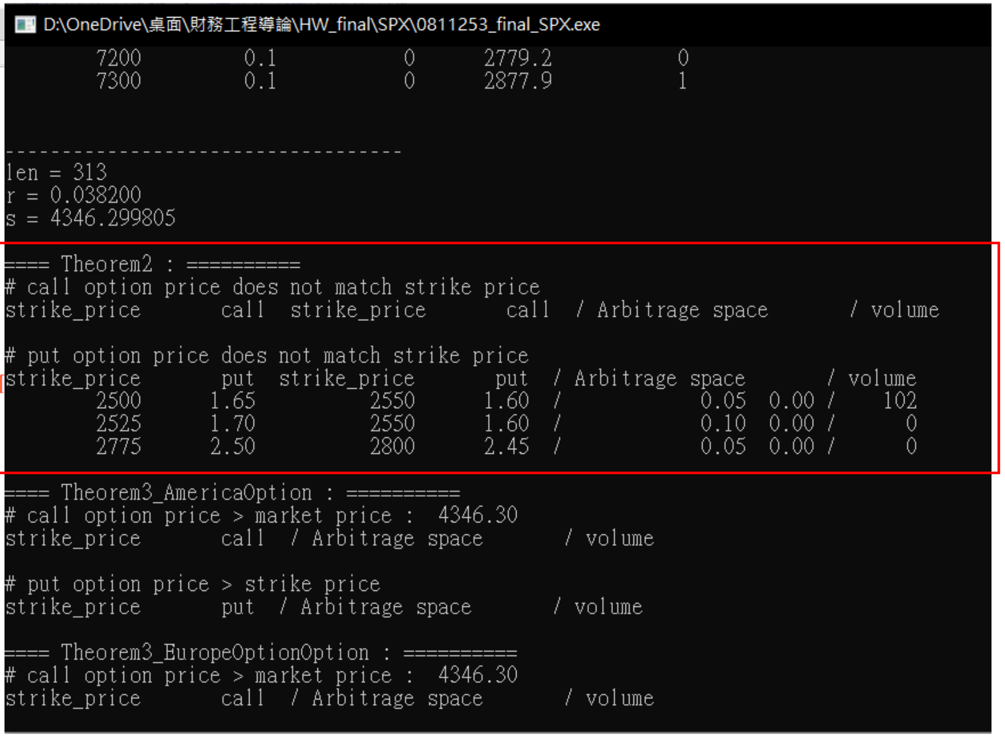
```
input data length : 313
input rate : 3.82
input Futures price : 4346.3
input strike_price call_price call_volume put_price put_volume
         200      4139        1     0.1       0
         400    3941.4        0     0.1       0
         600    3743.9        0     0.1       0
         800    3546.3        0    0.15       0
        1000    3348.8        0    0.15      17
        1200    3151.3        0    0.25       9
        1400    2953.8        0     0.3       0
        1500    2855.1        0     0.3       0
        1600    2756.3        0    0.35    1500
        1700    2657.6        0     0.4       0
        1800    2558.9        0    0.45       0
        1900    2460.2        0     0.5       0
        2000    2361.5        0     0.6     332
        2100    2262.8        0     0.7    5000
        2150    2213.5        0    0.75       0
        2200    2164.2        0    0.85       0
        2250    2114.9        0     0.9       9
        2300    2065.6        0       1     307
        2350    2016.3        0     1.3       4
        2400      1967        0     1.4       7
        2450    1917.7        0     1.5       0
        2500    1868.5        0    1.65     102
        2525    1843.9        0     1.7       0
        2550    1819.2        0     1.6     200
        2575    1794.6        0    1.85       0
        2600      1770        0     1.9       4
```

2. Void Theorem2



0811253_final_SPX.c - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Start here
0811253_final_SPX.c

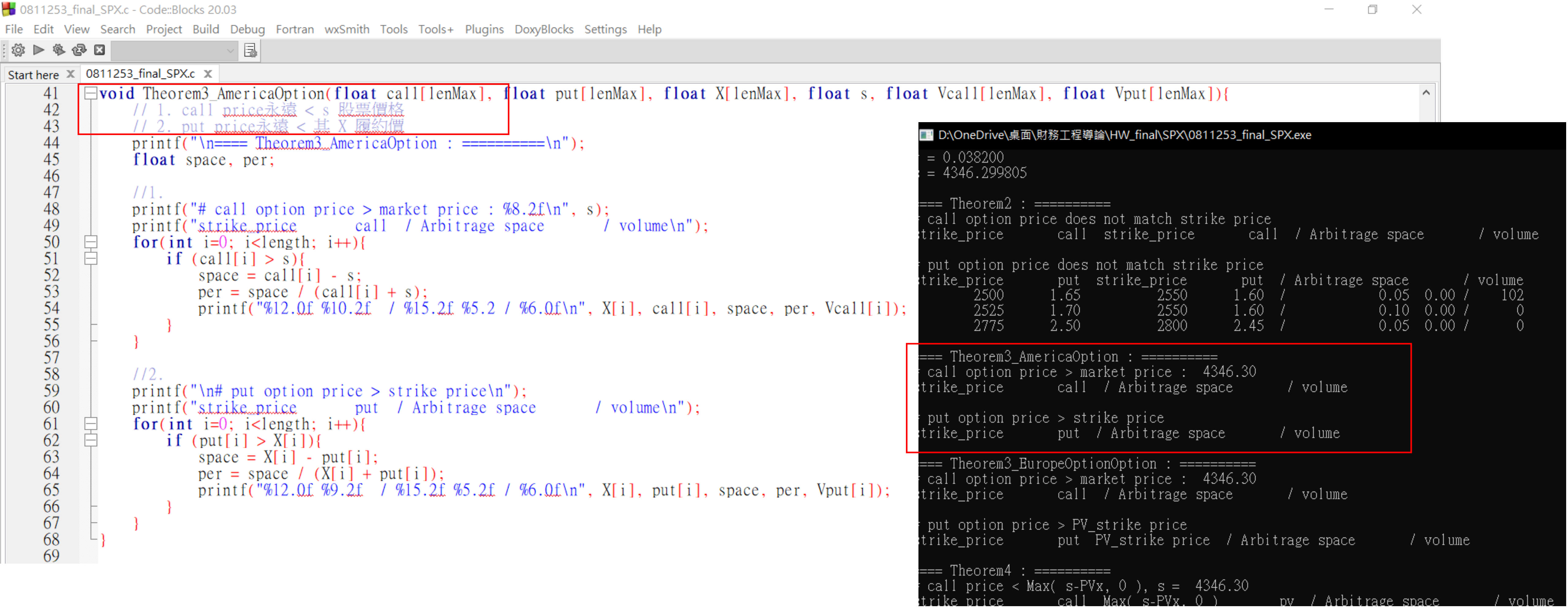
```c
void Theorem2(float call[lenMax], float put[lenMax], float X[lenMax], float Vcall[lenMax], float Vput[lenMax]){
    // 履約價單調性原理
    // 1. 一標的call option履約價高的的價錢需 < 履約價低的的價錢
    // 2.        put option履約價高的的價錢需 > 履約價低的的價錢
    printf("==== Theorem2 : ==========\n");
    float space, per;

    printf("# call option price does not match strike price\n");
    printf("strike_price       call  strike_price       call  / Arbitrage space       / volume\n");
    for(int i=0; i<length; i++){
        for(int j=i+1; j<length-i-1; j++){
            if(call[i] < call[j]){
                space = call[j] - call[i];
                per = space / (call[j] + call[i]);
                printf("%12.0f %10.2f  %12.0f %10.2f  / %15.2f %5.2f / %6.0f\n", X[i], call[i], X[j], call[j], space, per, Vcall[
            }
        }
    }

    printf("\n# put option price does not match strike price \n");
    printf("strike_price       put  strike_price       put  / Arbitrage space       / volume\n");
    for(int i=0; i<length; i++){
        for(int j=i+1; j<length-i-1; j++){
            if(put[i] > put[j]){
                space = put[i] - put[j];
                per = space / (call[j] + call[i]);
                printf("%12.0f %9.2f  %12.0f %9.2f  / %15.2f %5.2f / %6.0f\n", X[i], put[i], X[j], put[j], space, per, Vput[i]);
            }
        }
    }

```



D:\OneDrive\桌面\財務工程導論\HW_final\SPX\0811253_final_SPX.exe

```
        7200         0.1            0      2779.2           0
        7300         0.1            0      2877.9           1


-----------------------------------
len = 313
r = 0.038200
s = 4346.299805
==== Theorem2 : ==========
# call option price does not match strike price
strike_price       call  strike_price       call  / Arbitrage space       / volume

# put option price does not match strike price
strike_price       put  strike_price       put  / Arbitrage space       / volume
        2500      1.65          2550      1.60  /            0.05  0.00 /    102
        2525      1.70          2550      1.60  /            0.10  0.00 /      0
        2775      2.50          2800      2.45  /            0.05  0.00 /      0
==== Theorem3_AmericaOption : ==========
# call option price > market price :  4346.30
strike_price       call  / Arbitrage space       / volume

# put option price > strike price
strike_price       put  / Arbitrage space       / volume

==== Theorem3_EuropeOptionOption : ==========
# call option price > market price :  4346.30
strike_price       call  / Arbitrage space       / volume
```

3. Void Theorem3_對美式



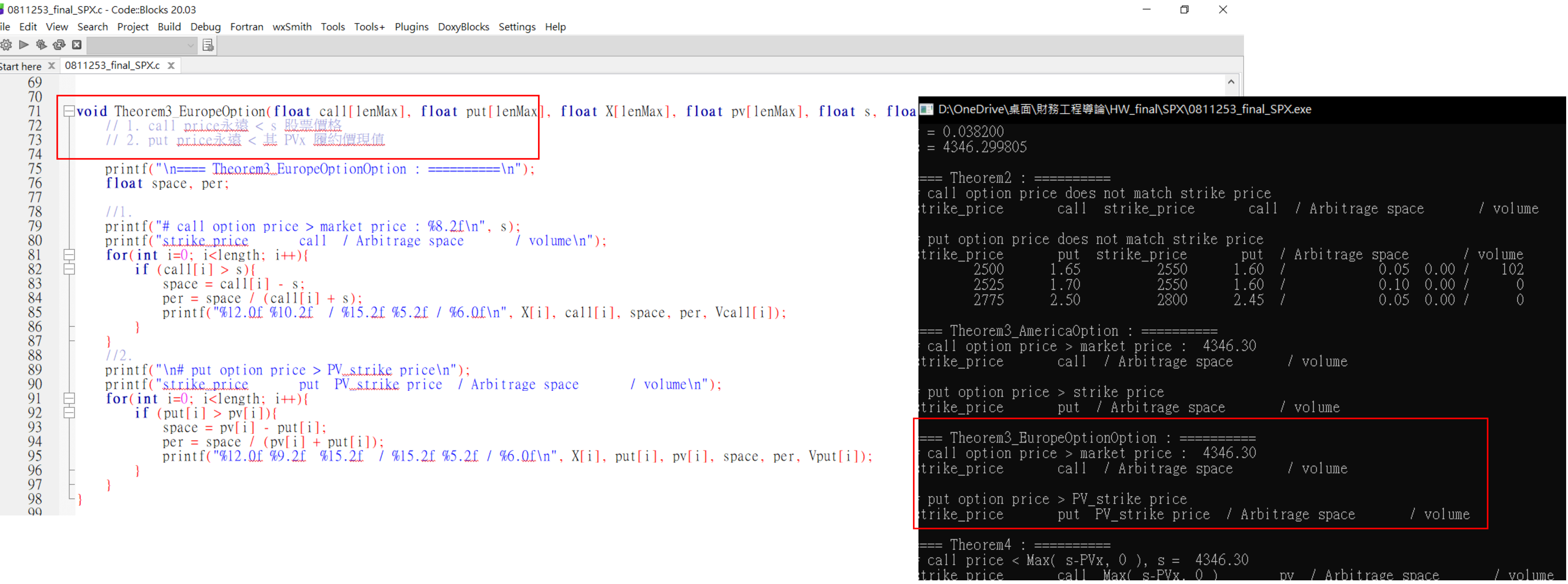
0811253_final_SPX.c - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Start here
0811253_final_SPX.c

```c
void Theorem3_AmericaOption(float call[lenMax], float put[lenMax], float X[lenMax], float s, float Vcall[lenMax], float Vput[lenMax]){
    // 1. call price永遠 < s 股票價格
    // 2. put price永遠 < 其 X 履約價
    printf("\n==== Theorem3_AmericaOption : ==========\n");
    float space, per;

    //1.
    printf("# call option price > market price : %8.2f\n", s);
    printf("strike_price        call  / Arbitrage space        / volume\n");
    for(int i=0; i<length; i++){
        if (call[i] > s){
            space = call[i] - s;
            per = space / (call[i] + s);
            printf("%12.0f %10.2f  / %15.2f %5.2 / %6.0f\n", X[i], call[i], space, per, Vcall[i]);
        }
    }

    //2.
    printf("\n# put option price > strike price\n");
    printf("strike_price        put  / Arbitrage space        / volume\n");
    for(int i=0; i<length; i++){
        if (put[i] > X[i]){
            space = X[i] - put[i];
            per = space / (X[i] + put[i]);
            printf("%12.0f %9.2f  / %15.2f %5.2f / %6.0f\n", X[i], put[i], space, per, Vput[i]);
        }
    }
}
```

D:\OneDrive\桌面\財務工程導論\HW_final\SPX\0811253_final_SPX.exe

```
 = 0.038200
 = 4346.299805

=== Theorem2 : ==========
 call option price does not match strike price
trike_price        call  strike_price        call  / Arbitrage space        / volume

 put option price does not match strike price
trike_price        put  strike_price        put  / Arbitrage space        / volume
       2500       1.65          2550        1.60  /             0.05  0.00 /    102
       2525       1.70          2550        1.60  /             0.10  0.00 /      0
       2775       2.50          2800        2.45  /             0.05  0.00 /      0

=== Theorem3_AmericaOption : ==========
 call option price > market price :  4346.30
trike_price        call  / Arbitrage space        / volume

 put option price > strike price
trike_price        put  / Arbitrage space        / volume

=== Theorem3_EuropeOptionOption : ==========
 call option price > market price :  4346.30
trike_price        call  / Arbitrage space        / volume

 put option price > PV_strike price
trike_price        put  PV_strike price  / Arbitrage space        / volume

=== Theorem4 : ==========
 call price < Max( s-PVx, 0 ), s =  4346.30
trike_price        call  Max( s-PVx, 0 )        pv  / Arbitrage space        / volume
```

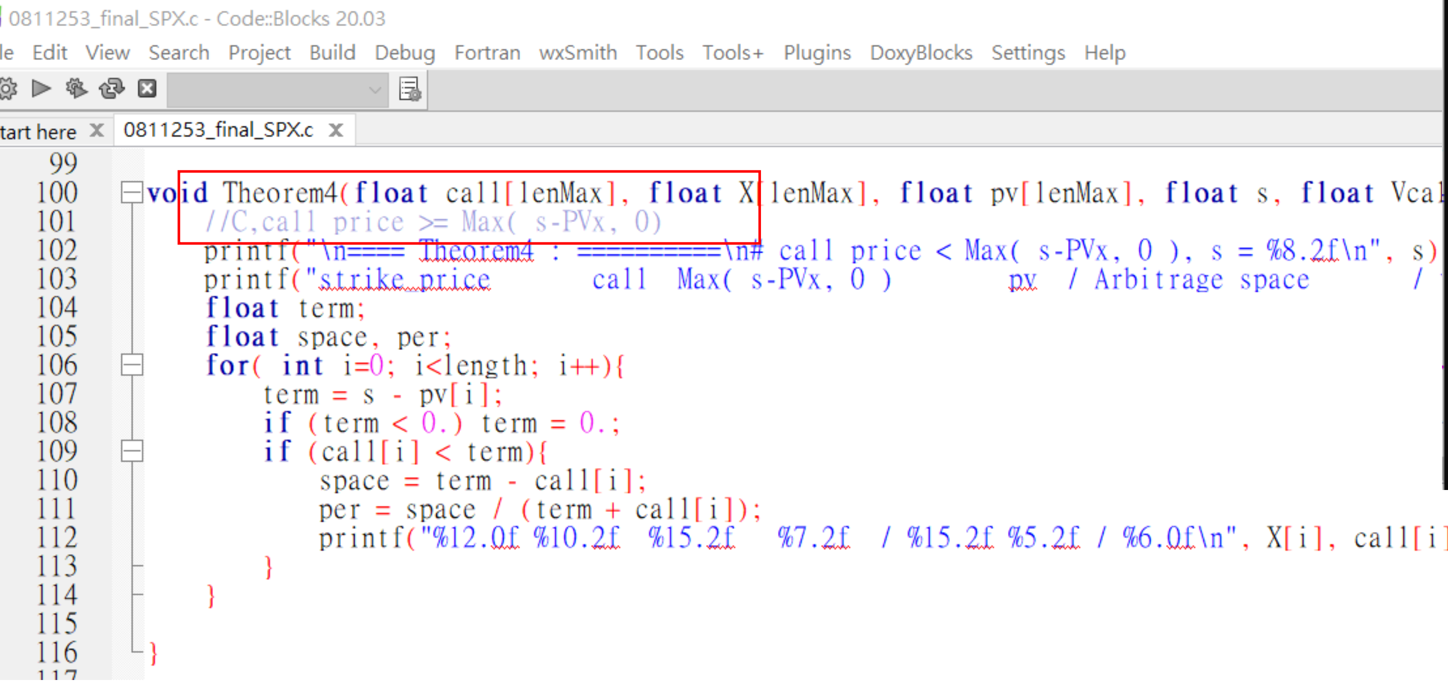
4. Void Theorem3_對歐式


0811253_final_SPX.c - Code::Blocks 20.03
ile Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Start here
0811253_final_SPX.c
D:\OneDrive\桌面\財務工程導論\HW_final\SPX\0811253_final_SPX.exe

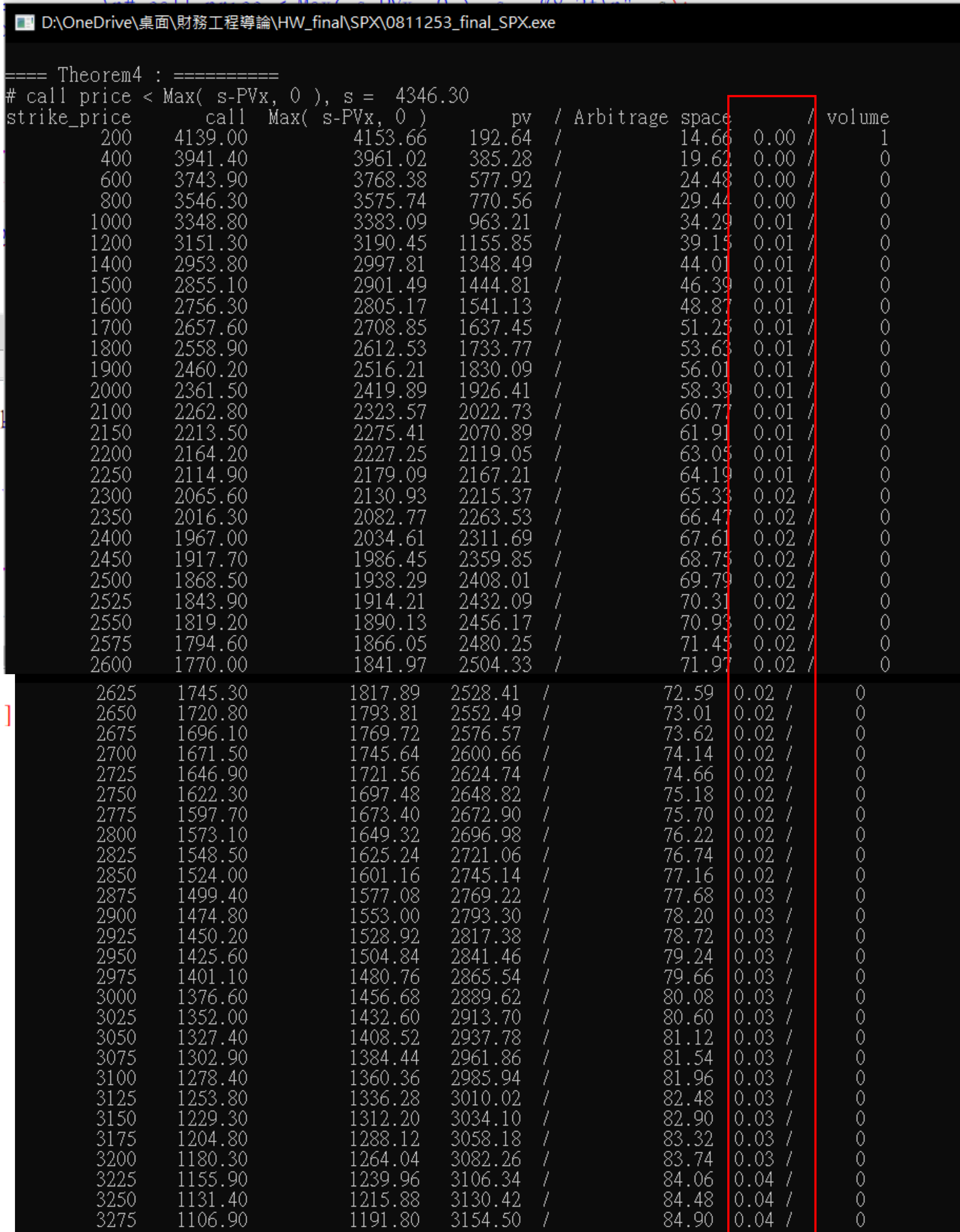
```c
void Theorem3_EuropeOption(float call[lenMax], float put[lenMax], float X[lenMax], float pv[lenMax], float s, floa
    // 1. call price永遠 < s 股票價格
    // 2. put price永遠 < 其 PVx 履約價現值

    printf("\n=== Theorem3_EuropeOptionOption : ==========\n");
    float space, per;

    //1.
    printf("# call option price > market price : %8.2f\n", s);
    printf("strike_price       call  / Arbitrage space       / volume\n");
    for(int i=0; i<length; i++){
        if (call[i] > s){
            space = call[i] - s;
            per = space / (call[i] + s);
            printf("%12.0f %10.2f  / %15.2f %5.2f / %6.0f\n", X[i], call[i], space, per, Vcall[i]);
        }
    }
    //2.
    printf("\n# put option price > PV_strike price\n");
    printf("strike_price       put  PV_strike price  / Arbitrage space       / volume\n");
    for(int i=0; i<length; i++){
        if (put[i] > pv[i]){
            space = pv[i] - put[i];
            per = space / (pv[i] + put[i]);
            printf("%12.0f %9.2f  %15.2f  / %15.2f %5.2f / %6.0f\n", X[i], put[i], pv[i], space, per, Vput[i]);
        }
    }
}
```

```
 = 0.038200
 = 4346.299805

=== Theorem2 : ==========
 call option price does not match strike price
trike_price       call  strike_price       call  / Arbitrage space       / volume

 put option price does not match strike price
trike_price       put  strike_price       put  / Arbitrage space       / volume
       2500      1.65          2550      1.60  /            0.05  0.00 /    102
       2525      1.70          2550      1.60  /            0.10  0.00 /      0
       2775      2.50          2800      2.45  /            0.05  0.00 /      0

=== Theorem3_AmericaOption : ==========
 call option price > market price :  4346.30
trike_price       call  / Arbitrage space       / volume

 put option price > strike price
trike_price       put  / Arbitrage space       / volume

=== Theorem3_EuropeOptionOption : ==========
 call option price > market price :  4346.30
trike_price       call  / Arbitrage space       / volume

 put option price > PV_strike price
trike_price       put  PV_strike price  / Arbitrage space       / volume

=== Theorem4 : ==========
 call price < Max( s-PVx, 0 ), s =  4346.30
trike_price       call  Max( s-PVx, 0 )       pv  / Arbitrage space       / volume
```

5. Void Theorem4

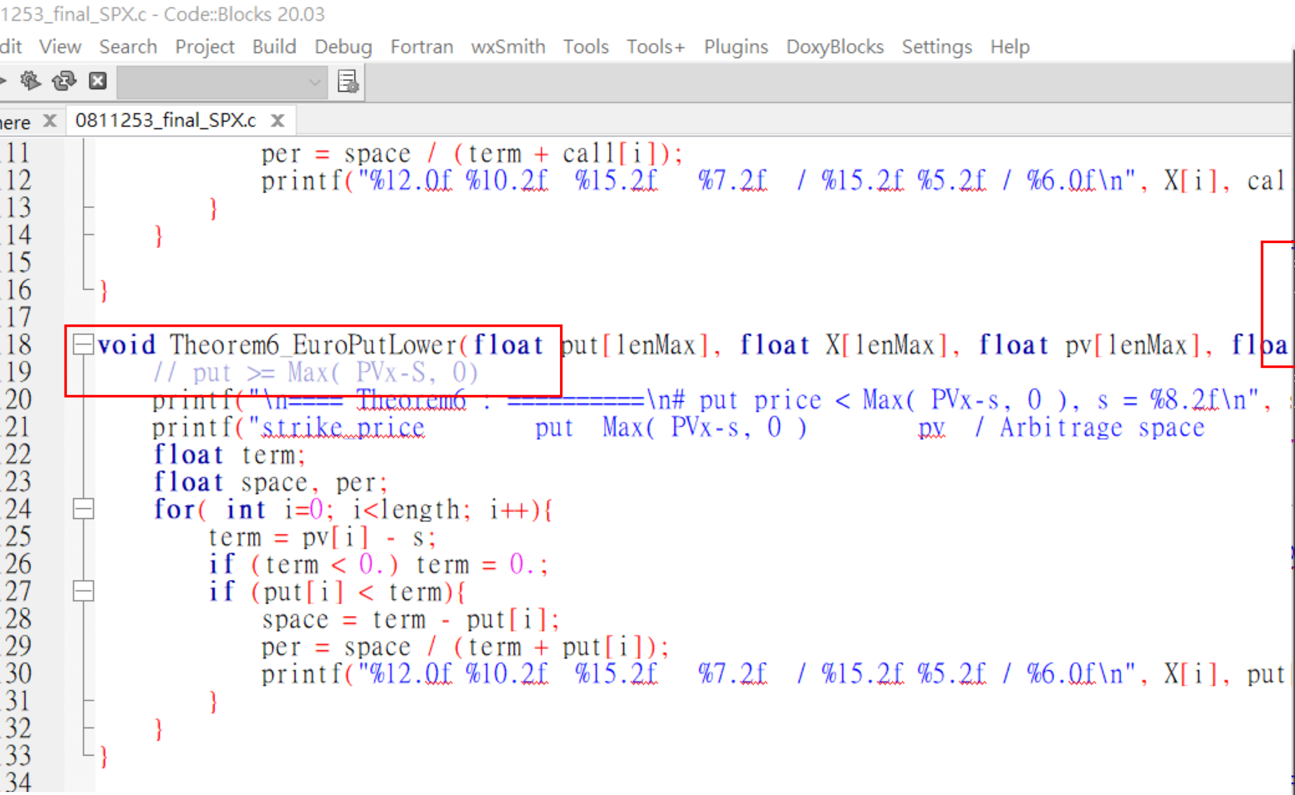

0811253_final_SPX.c - Code::Blocks 20.03
le Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
tart here
0811253_final_SPX.c

```c

void Theorem4(float call[lenMax], float X[lenMax], float pv[lenMax], float s, float Vca
    //C,call price >= Max( s-PVx, 0)
    printf("\n==== Theorem4 : ==========\n# call price < Max( s-PVx, 0 ), s = %8.2f\n", s)
    printf("strike_price        call  Max( s-PVx, 0 )          pv  / Arbitrage space       / 
    float term;
    float space, per;
    for( int i=0; i<length; i++){
        term = s - pv[i];
        if (term < 0.) term = 0.;
        if (call[i] < term){
            space = term - call[i];
            per = space / (term + call[i]);
            printf("%12.0f %10.2f  %15.2f   %7.2f  / %15.2f %5.2f / %6.0f\n", X[i], call[i]
        }
    }

}
```

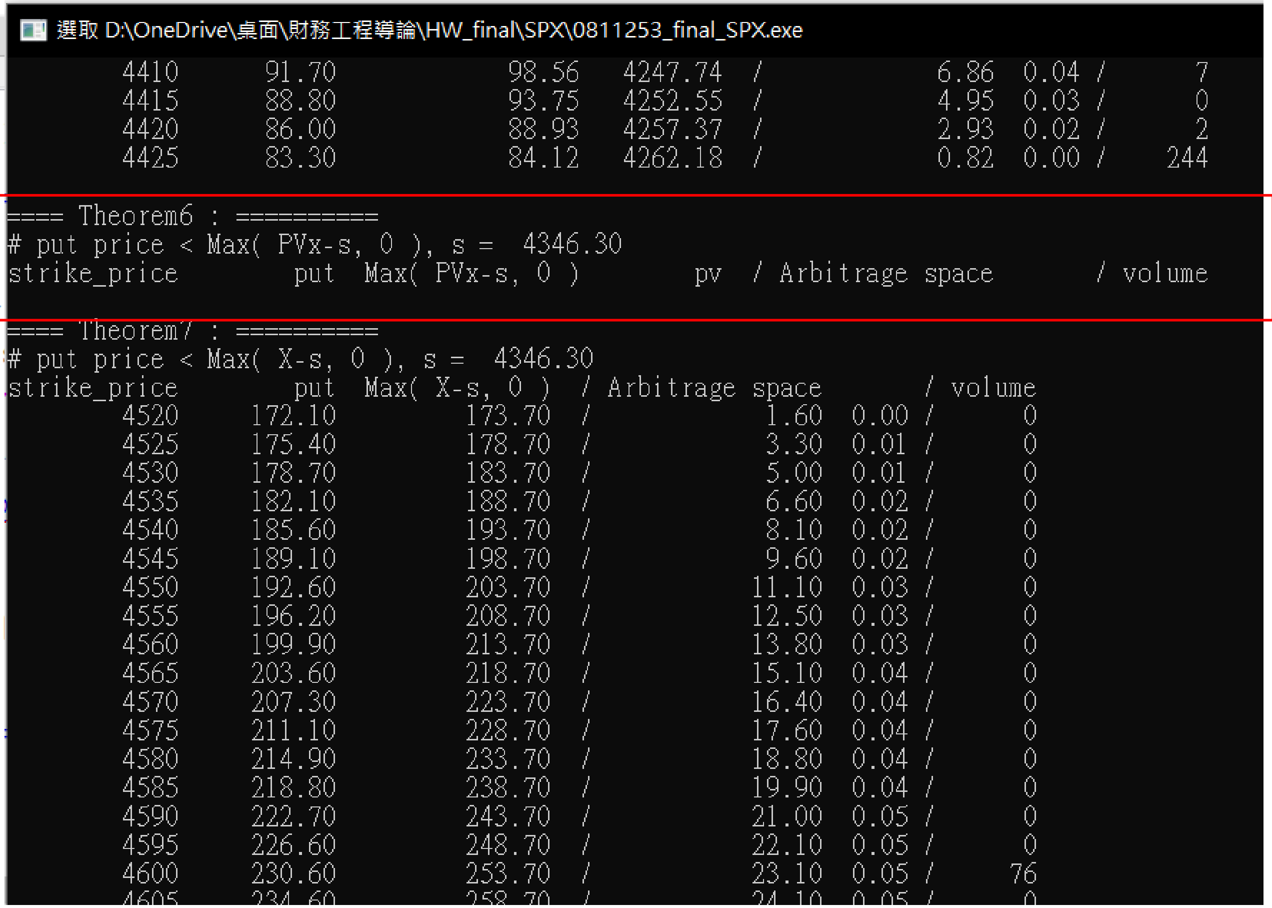

D:\OneDrive\桌面\財務工程導論\HW_final\SPX\0811253_final_SPX.exe

```
==== Theorem4 : ==========
# call price < Max( s-PVx, 0 ), s =  4346.30
strike_price        call  Max( s-PVx, 0 )          pv  / Arbitrage space       / volume
         200    4139.00          4153.66    192.64  /           14.66  0.00 /      1
         400    3941.40          3961.02    385.28  /           19.62  0.00 /      0
         600    3743.90          3768.38    577.92  /           24.48  0.00 /      0
         800    3546.30          3575.74    770.56  /           29.44  0.00 /      0
        1000    3348.80          3383.09    963.21  /           34.29  0.01 /      0
        1200    3151.30          3190.45   1155.85  /           39.15  0.01 /      0
        1400    2953.80          2997.81   1348.49  /           44.01  0.01 /      0
        1500    2855.10          2901.49   1444.81  /           46.39  0.01 /      0
        1600    2756.30          2805.17   1541.13  /           48.87  0.01 /      0
        1700    2657.60          2708.85   1637.45  /           51.25  0.01 /      0
        1800    2558.90          2612.53   1733.77  /           53.63  0.01 /      0
        1900    2460.20          2516.21   1830.09  /           56.01  0.01 /      0
        2000    2361.50          2419.89   1926.41  /           58.39  0.01 /      0
        2100    2262.80          2323.57   2022.73  /           60.77  0.01 /      0
        2150    2213.50          2275.41   2070.89  /           61.91  0.01 /      0
        2200    2164.20          2227.25   2119.05  /           63.05  0.01 /      0
        2250    2114.90          2179.09   2167.21  /           64.19  0.01 /      0
        2300    2065.60          2130.93   2215.37  /           65.33  0.02 /      0
        2350    2016.30          2082.77   2263.53  /           66.47  0.02 /      0
        2400    1967.00          2034.61   2311.69  /           67.61  0.02 /      0
        2450    1917.70          1986.45   2359.85  /           68.75  0.02 /      0
        2500    1868.50          1938.29   2408.01  /           69.79  0.02 /      0
        2525    1843.90          1914.21   2432.09  /           70.31  0.02 /      0
        2550    1819.20          1890.13   2456.17  /           70.93  0.02 /      0
        2575    1794.60          1866.05   2480.25  /           71.45  0.02 /      0
        2600    1770.00          1841.97   2504.33  /           71.97  0.02 /      0
        2625    1745.30          1817.89   2528.41  /           72.59  0.02 /      0
        2650    1720.80          1793.81   2552.49  /           73.01  0.02 /      0
        2675    1696.10          1769.72   2576.57  /           73.62  0.02 /      0
        2700    1671.50          1745.64   2600.66  /           74.14  0.02 /      0
        2725    1646.90          1721.56   2624.74  /           74.66  0.02 /      0
        2750    1622.30          1697.48   2648.82  /           75.18  0.02 /      0
        2775    1597.70          1673.40   2672.90  /           75.70  0.02 /      0
        2800    1573.10          1649.32   2696.98  /           76.22  0.02 /      0
        2825    1548.50          1625.24   2721.06  /           76.74  0.02 /      0
        2850    1524.00          1601.16   2745.14  /           77.16  0.02 /      0
        2875    1499.40          1577.08   2769.22  /           77.68  0.03 /      0
        2900    1474.80          1553.00   2793.30  /           78.20  0.03 /      0
        2925    1450.20          1528.92   2817.38  /           78.72  0.03 /      0
        2950    1425.60          1504.84   2841.46  /           79.24  0.03 /      0
        2975    1401.10          1480.76   2865.54  /           79.66  0.03 /      0
        3000    1376.60          1456.68   2889.62  /           80.08  0.03 /      0
        3025    1352.00          1432.60   2913.70  /           80.60  0.03 /      0
        3050    1327.40          1408.52   2937.78  /           81.12  0.03 /      0
        3075    1302.90          1384.44   2961.86  /           81.54  0.03 /      0
        3100    1278.40          1360.36   2985.94  /           81.96  0.03 /      0
        3125    1253.80          1336.28   3010.02  /           82.48  0.03 /      0
        3150    1229.30          1312.20   3034.10  /           82.90  0.03 /      0
        3175    1204.80          1288.12   3058.18  /           83.32  0.03 /      0
        3200    1180.30          1264.04   3082.26  /           83.74  0.03 /      0
        3225    1155.90          1239.96   3106.34  /           84.06  0.04 /      0
        3250    1131.40          1215.88   3130.42  /           84.48  0.04 /      0
        3275    1106.90          1191.80   3154.50  /           84.90  0.04 /      0
```

6. Void Theorem6_對歐式

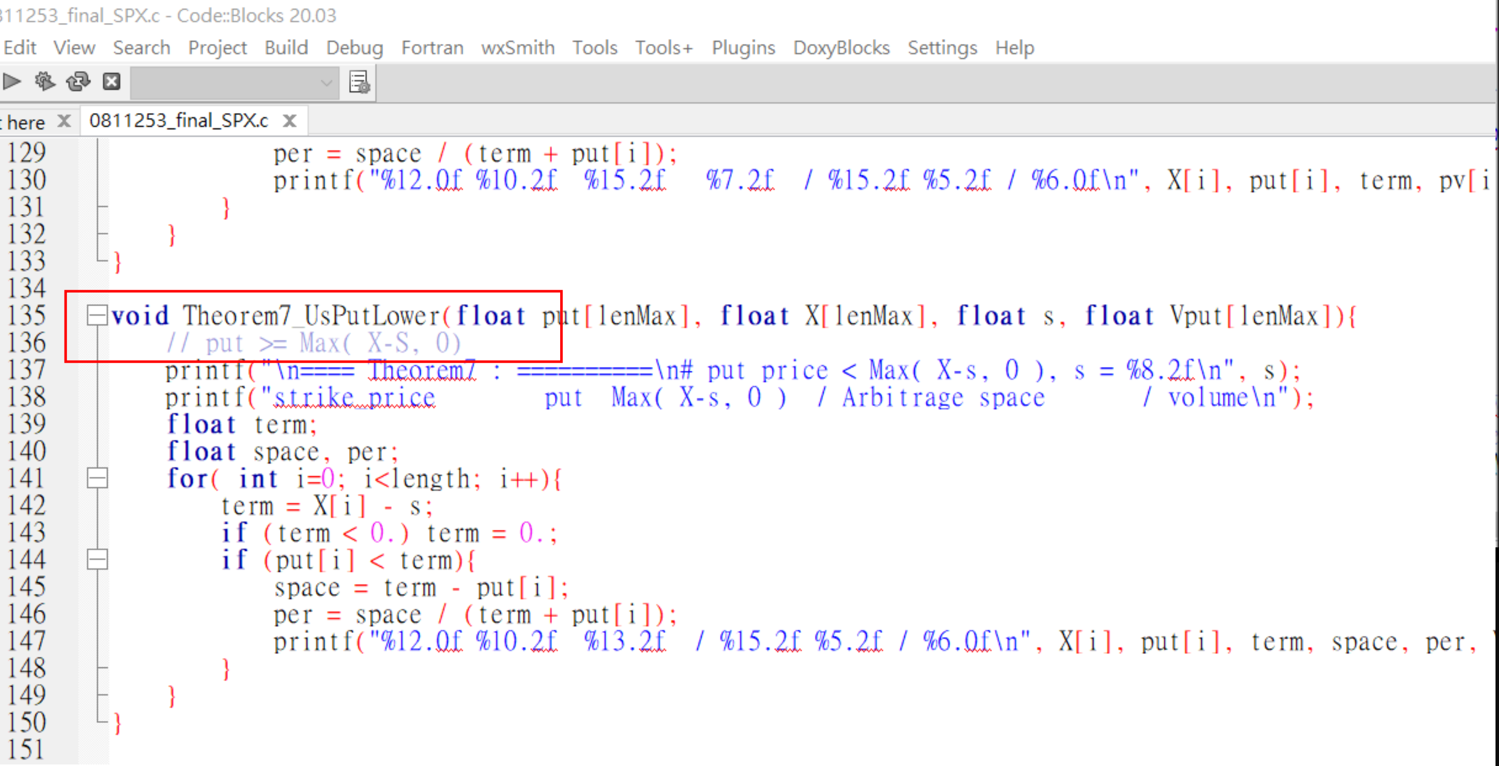

1253_final_SPX.c - Code::Blocks 20.03
dit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
ere
0811253_final_SPX.c

```c
                per = space / (term + call[i]);
                printf("%12.0f %10.2f  %15.2f   %7.2f  / %15.2f %5.2f / %6.0f\n", X[i], cal
            }
        }

}

void Theorem6_EuroPutLower(float put[lenMax], float X[lenMax], float pv[lenMax], floa
    // put >= Max( PVx-S, 0)
    printf("\n==== Theorem6 : ==========\n# put price < Max( PVx-s, 0 ), s = %8.2f\n", 
    printf("strike_price          put  Max( PVx-s, 0 )         pv  / Arbitrage space
    float term;
    float space, per;
    for( int i=0; i<length; i++){
        term = pv[i] - s;
        if (term < 0.) term = 0.;
        if (put[i] < term){
            space = term - put[i];
            per = space / (term + put[i]);
            printf("%12.0f %10.2f  %15.2f   %7.2f  / %15.2f %5.2f / %6.0f\n", X[i], put
        }
    }
}
```

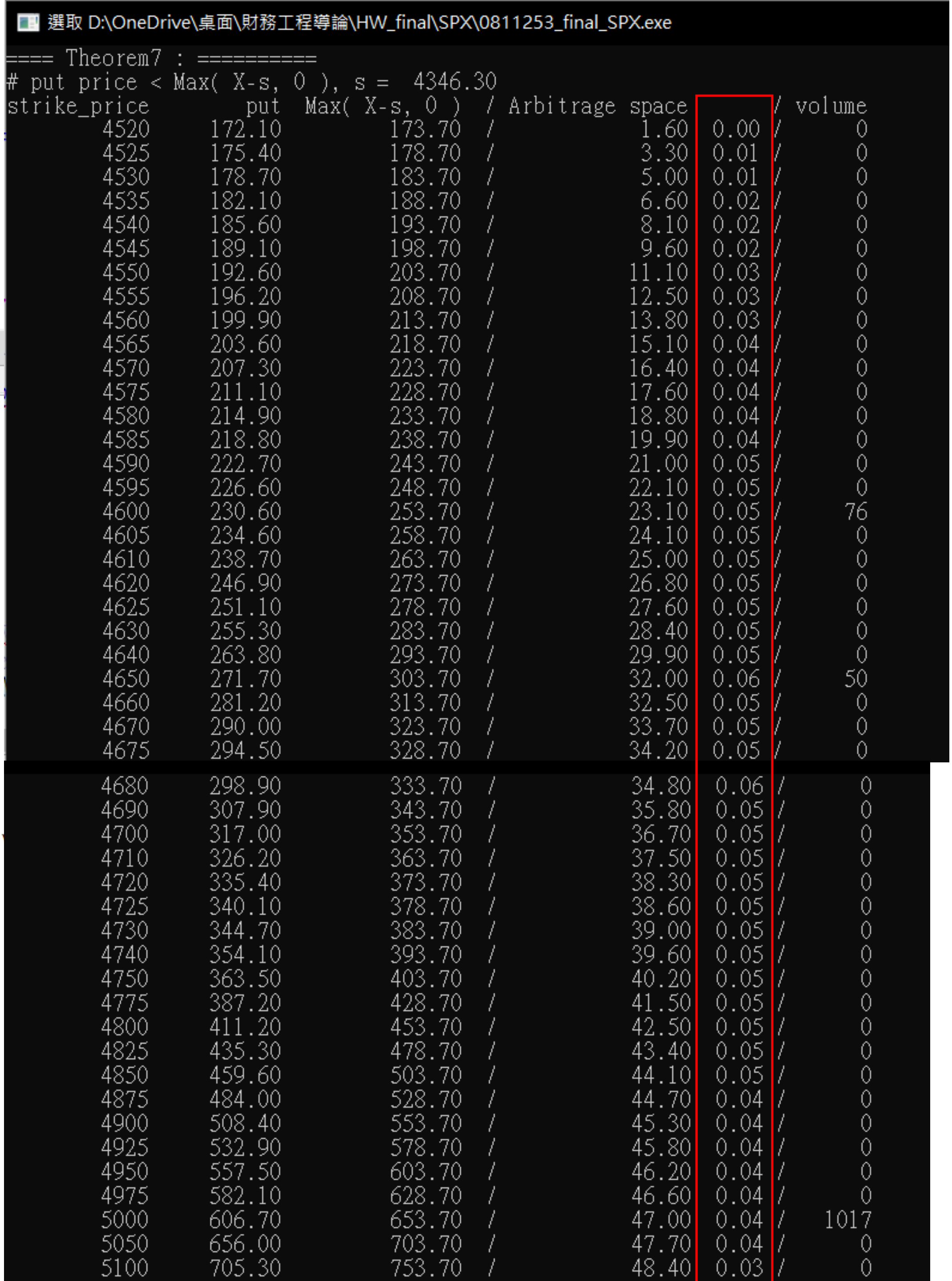

選取 D:\OneDrive\桌面\財務工程導論\HW_final\SPX\0811253_final_SPX.exe

```
        4410      91.70              98.56    4247.74   /              6.86  0.04 /        7
        4415      88.80              93.75    4252.55   /              4.95  0.03 /        0
        4420      86.00              88.93    4257.37   /              2.93  0.02 /        2
        4425      83.30              84.12    4262.18   /              0.82  0.00 /      244

==== Theorem6 : ==========
# put price < Max( PVx-s, 0 ), s =  4346.30
strike_price          put  Max( PVx-s, 0 )         pv  / Arbitrage space         / volume

==== Theorem7 : ==========
# put price < Max( X-s, 0 ), s =  4346.30
strike_price          put  Max( X-s, 0 )  / Arbitrage space         / volume
        4520     172.10           173.70  /            1.60  0.00 /        0
        4525     175.40           178.70  /            3.30  0.01 /        0
        4530     178.70           183.70  /            5.00  0.01 /        0
        4535     182.10           188.70  /            6.60  0.02 /        0
        4540     185.60           193.70  /            8.10  0.02 /        0
        4545     189.10           198.70  /            9.60  0.02 /        0
        4550     192.60           203.70  /           11.10  0.03 /        0
        4555     196.20           208.70  /           12.50  0.03 /        0
        4560     199.90           213.70  /           13.80  0.03 /        0
        4565     203.60           218.70  /           15.10  0.04 /        0
        4570     207.30           223.70  /           16.40  0.04 /        0
        4575     211.10           228.70  /           17.60  0.04 /        0
        4580     214.90           233.70  /           18.80  0.04 /        0
        4585     218.80           238.70  /           19.90  0.04 /        0
        4590     222.70           243.70  /           21.00  0.05 /        0
        4595     226.60           248.70  /           22.10  0.05 /        0
        4600     230.60           253.70  /           23.10  0.05 /       76
        4605     234.60           258.70  /           24.10  0.05 /        0
```

7. Void Theorem6_對美式


311253_final_SPX.c - Code::Blocks 20.03
Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
here
0811253_final_SPX.c

```c
            per = space / (term + put[i]);
            printf("%12.0f %10.2f  %15.2f   %7.2f  / %15.2f %5.2f / %6.0f\n", X[i], put[i], term, pv[i
        }
    }
}

void Theorem7_UsPutLower(float put[lenMax], float X[lenMax], float s, float Vput[lenMax]){
    // put >= Max( X-S, 0)
    printf("\n==== Theorem7 : ==========\n# put price < Max( X-s, 0 ), s = %8.2f\n", s);
    printf("strike_price          put  Max( X-s, 0 )  / Arbitrage space       / volume\n");
    float term;
    float space, per;
    for( int i=0; i<length; i++){
        term = X[i] - s;
        if (term < 0.) term = 0.;
        if (put[i] < term){
            space = term - put[i];
            per = space / (term + put[i]);
            printf("%12.0f %10.2f  %13.2f  / %15.2f %5.2f / %6.0f\n", X[i], put[i], term, space, per,
        }
    }
}
```


選取 D:\OneDrive\桌面\財務工程導論\HW_final\SPX\0811253_final_SPX.exe

```
==== Theorem7 : ==========
# put price < Max( X-s, 0 ), s =  4346.30
strike_price          put  Max( X-s, 0 )  / Arbitrage space       / volume
        4520     172.10         173.70  /            1.60  0.00 /      0
        4525     175.40         178.70  /            3.30  0.01 /      0
        4530     178.70         183.70  /            5.00  0.01 /      0
        4535     182.10         188.70  /            6.60  0.02 /      0
        4540     185.60         193.70  /            8.10  0.02 /      0
        4545     189.10         198.70  /            9.60  0.02 /      0
        4550     192.60         203.70  /           11.10  0.03 /      0
        4555     196.20         208.70  /           12.50  0.03 /      0
        4560     199.90         213.70  /           13.80  0.03 /      0
        4565     203.60         218.70  /           15.10  0.04 /      0
        4570     207.30         223.70  /           16.40  0.04 /      0
        4575     211.10         228.70  /           17.60  0.04 /      0
        4580     214.90         233.70  /           18.80  0.04 /      0
        4585     218.80         238.70  /           19.90  0.04 /      0
        4590     222.70         243.70  /           21.00  0.05 /      0
        4595     226.60         248.70  /           22.10  0.05 /      0
        4600     230.60         253.70  /           23.10  0.05 /     76
        4605     234.60         258.70  /           24.10  0.05 /      0
        4610     238.70         263.70  /           25.00  0.05 /      0
        4620     246.90         273.70  /           26.80  0.05 /      0
        4625     251.10         278.70  /           27.60  0.05 /      0
        4630     255.30         283.70  /           28.40  0.05 /      0
        4640     263.80         293.70  /           29.90  0.05 /      0
        4650     271.70         303.70  /           32.00  0.06 /     50
        4660     281.20         313.70  /           32.50  0.05 /      0
        4670     290.00         323.70  /           33.70  0.05 /      0
        4675     294.50         328.70  /           34.20  0.05 /      0
        4680     298.90         333.70  /           34.80  0.06 /      0
        4690     307.90         343.70  /           35.80  0.05 /      0
        4700     317.00         353.70  /           36.70  0.05 /      0
        4710     326.20         363.70  /           37.50  0.05 /      0
        4720     335.40         373.70  /           38.30  0.05 /      0
        4725     340.10         378.70  /           38.60  0.05 /      0
        4730     344.70         383.70  /           39.00  0.05 /      0
        4740     354.10         393.70  /           39.60  0.05 /      0
        4750     363.50         403.70  /           40.20  0.05 /      0
        4775     387.20         428.70  /           41.50  0.05 /      0
        4800     411.20         453.70  /           42.50  0.05 /      0
        4825     435.30         478.70  /           43.40  0.05 /      0
        4850     459.60         503.70  /           44.10  0.05 /      0
        4875     484.00         528.70  /           44.70  0.04 /      0
        4900     508.40         553.70  /           45.30  0.04 /      0
        4925     532.90         578.70  /           45.80  0.04 /      0
        4950     557.50         603.70  /           46.20  0.04 /      0
        4975     582.10         628.70  /           46.60  0.04 /      0
        5000     606.70         653.70  /           47.00  0.04 /   1017
        5050     656.00         703.70  /           47.70  0.04 /      0
        5100     705.30         753.70  /           48.40  0.03 /      0
```

套利分析

- 有套利空間的條件: T2，T4，T7
- 分析:

  可以看到這個程式跑出來有一些條件確實是可以進行選擇權套利的，但有一個原因使得其存在不確定性，因為我的套利空間並沒有扣除成本，而其中可以看到T2的套利空間對成本是趨近0%，T4、T7的空間最高也只有0.11%所以根據不同加券商或者大戶談條件，期權的手續費等成本確實有可能到0.11%下的機會，但是我們的可能就微乎其微，所以我判斷這一天的期權價格是無法套利的。