

Q1

```
for(int k=0;k<5;k=k+1)
{ //計算第i次payoff的現值
    float Discount=1;
    Discount = pow((1 + ZeroRate[k]), -k-1);
    BondValue=BondValue+C*Discount;
    if(k==4)
    {
        BondValue=BondValue+100*Discount;
    }
}
```

```
-Method-in-Finance/Program/L3/homework5/"hw5-1_109705056
請輸入債息:1
輸入Zero rate
輸入第1期的Zero rate:0.01
輸入第2期的Zero rate:0.01
輸入第3期的Zero rate:0.01
輸入第4期的Zero rate:0.01
輸入第5期的Zero rate:0.01
債券價格=100.000008
○ (base) johnsonhsiao@Johnsons-MacBook-Pro homework5 %
```

這裡是用 `pow` 函式替代原本的 `for loop`，然後帶入上課中老師帶的程式範例的測試資料來驗證是否與範例程式相同。

Q2

```
double calc_price(double r) {
    double price = 0;

    for (int i = 1; i <= n; i++) {
        price += S[i-1] / pow(1 + r, i);
    }

    price += principal / pow(1 + r, n);
    return price;
}
```

`calc_price` 的函數，用於計算給定利率下現金流的現值。該函數採用一個雙精度

浮點數 `r` 作為其唯一參數，返回一個雙精度浮點數表示現值。

```
double bisection(double lower, double upper, double tolerance) {
    double mid = (lower + upper) / 2;

    while (abs(upper - lower) > tolerance) {
        double tmp_a = calc_price(lower) - principal;
        double tmp_b = calc_price(upper) - principal;
        double tmp_c = calc_price(mid) - principal;
        if (tmp_c == 0) {
            return mid;
        } else if (tmp_a * tmp_c < 0) {
            upper = mid;
        } else if (tmp_b * tmp_c < 0) {
            lower = mid;
        }
        mid = (lower + upper) / 2;
    }
    return mid;
}
```

`bisection()` 函數使用二分法來計算給定誤差下的固定利率。該函數採用三個雙精度浮點數參數，即 `lower`、`upper` 和 `tolerance`，其中 `lower` 和 `upper` 分別表示利率的下限和上限，`tolerance` 表示誤差的容限。函數返回一個雙精度浮點數表示所計算出的利率。

```
int main(int argc, char *argv[]){
    double r = bisection(lower, upper, tolerance);

    printf("The fix rate is: %lf\n", r);

    return 0;
}
```

最後，程式在 `main()` 函數中定義了三個變數：`lower`、`upper` 和 `tolerance`，它們分別表示利率的下限、上限和誤差的容限。程式使用 `bisection()` 函數計算給定誤差下的固定利率，然後輸出該利率。