Q1:

```cpp
double calculateBondPrice(double FV, double c, int n, double ytm) {
    double bondPrice = 0.0;
    for (int i = 1; i <= n; ++i) {
        bondPrice += c / std::pow(1 + ytm, i);  //將利息折現
    }
    bondPrice += FV / std::pow(1 + ytm, n);  //將面額折現
    return bondPrice;
}
```

這個 function 主要是方便直接折現出債券的現值，在下面的函示中呼叫即可

```cpp
double newYTM(double FV, double c, int n, double p) {
    double epsilon = 0.0001;
    double lower = 0.0;
    double upper = 1.0;
    double ytm = (lower + upper) / 2.0;
    double diff = calculateBondPrice(FV, c, n, ytm) - p;

    while (std::abs(diff) > epsilon) {
        if (diff < 0) {
            upper = ytm;
        }
        else {
            lower = ytm;
        }
        ytm = (lower + upper) / 2.0;
        diff = calculateBondPrice(FV, c, n, ytm) - p;
    }
    return ytm;
}
```

這個函示主要是要用 bisection 的方式求出他的殖利率，也就是用逼近法

```cpp
int main() {
    double FV; // Face value
    double c ; // 利息
    int n ; // 期數
    double p ; // 價格
    double r ; // Risk-free interest rate
    scanf("%lf",FV);
    scanf("%lf",c);
    scanf("%d",n);
    scanf("%lf",p);
    scanf("%lf",r);

    double ytm = newYTM(FV, c, n, p);
    double yieldSpread = ytm - r;

    std::cout << "Yield Spread (S): " << yieldSpread << std::endl;

    return 0;
}
```

再拿算出來的東西間掉無風險利率則可以符合題目要求

Q2:

由於程式內容與範例大致相似所以我將提供 terminal 以供查證