CSci 14 Lab Finals Practical Exam

Problem: Prisoner's Dilemma

Two prisoners are separated into individual rooms and cannot communicate with each other. The normal game is shown below:

Prisoner B / **Prisoner A**	**Prisoner B stays silent (*cooperates*)**	**Prisoner B testifies (*defects*)**
Prisoner A stays silent (*cooperates*)	Each serve 2 years	Prisoner A: 10 years Prisoner B: goes free
Prisoner A testifies (*defects*)	Prisoner A: goes free Prisoner B: 10 years	Each serve 5 years

Base class:

```
class Prisoner
{
    public:
        virtual bool getChoice()=0;
        virtual void setYearSentence(int)=0;
        virtual int getYearSentence()=0;

        ~Prisoner();
        Prisoner();
    protected:
        bool choice;
        int yearSentence;
};
```

Derived class:

```
#define COOP 0
#define DEF 1

class HumanPrisoner:public Prisoner
{
    public:
        void setChoice(bool x);
        bool getChoice();
        void setYearSentence(int);
        int getYearSentence();
        HumanPrisoner();
        ~HumanPrisoner();
};
```

```
class ComputerPrisoner: public Prisoner
{
    public:
        void generateChoice();
        bool getChoice();
        void setYearSentence(int);
        int getYearSentence();
        ComputerPrisoner();
        ~ComputerPrisoner();
};
```

Arbiter class:

```
class Arbiter
{
    public:
         void getDecision(Prisoner &x, Prisoner &y);
         Arbiter();
        ~Arbiter();

};
```

Try this on function main()

```
   HumanPrisoner h,h1;
    Arbiter a,a1;
    ComputerPrisoner c;

    h.setChoice(COOP);
    h1.setChoice(COOP);
    c.generateChoice();

    a.getDecision(h,h1);
    cout<<"HumanPrisoner1Sentence:"
    <<h.getYearSentence()<<endl;
    cout<<"HumanPrisoner2Sentence:"
    <<h1.getYearSentence()<<endl;

    a1.getDecision(h,c);
    cout<<"HumanPrisonerSentence:"
    <<h.getYearSentence()<<endl;
    cout<<"ComputerPrisonerSentence:"
    <<c.getYearSentence()<<endl;
    return 0;

/*Output: Assuming Computer chooses to Defect*/
HumanPrisoner1Sentence:2
HumanPrisoner2Sentence:2
HumanPrisonerSentence:10
ComputerPrisonerSentence:0
```

Functions to implement on client code:

```
/* Generate the choice using the appropriate function, then display the
result for ComputerPrisoner1 set and Computer Prisoner2 set with a size of n
void CVC (ComputerPrisoner C1Set[], ComputerPrisoner C2Set[], int n);
*/

Using getch() and toupper()function from <conio.h>, ask for the user input.
'C' or 'c' for cooperate and 'D' or 'c' for defect. Be sure that only legal
inputs would be accepted. If the input is invalid, the process would be
repeated all over again until the right input is satisfied. After that
display the result of the operation. Using the n as the size of the input.

void PVP (HumanPrisoner P1Set[], HumanPrisoner P2Set[], int n);

/*
Still using getch() and toupper()function from <conio.h>, ask for the user
input. 'C or c' for cooperate and 'D or d' for defect for HumanPrisoner. Be
sure that only legal inputs would be accepted. For computer player generate
the choice using the appropriate function. Using the n as the size of the
input (20pts)
*/
void PVC (HumanPrisoner PSet[], ComputerPrisoner CSet[], int n);
```