Help

```
#ifndef _BASKET_H
#define _BASKET_H

#include "pnl/pnl_matrix.h"
#include "
href../../../../common/math/jlparser/include/jlparser/parser_h_src.pdfjlparser/p
#include "
href../../../../common/math/mcam/src/Option_h_src.pdfOption.hpp"

namespace mcam {
class BasketOption : public Option
{
public:
    BasketOption();
    BasketOption(const Param &P);
    ~BasketOption();
    void print() const;

    double K;
    PnlVect *lambda;

    double payoff(const PnlMat *path_val, int t);
};

class GeometricPutOption : public Option
{
public:
    GeometricPutOption();
    GeometricPutOption(const Param &P);
    ~GeometricPutOption();
    void print() const;

    double K;

    double payoff(const PnlMat *path_val, int t);
private:
    PnlVect *sigma;
    int size;
};
```

```
class GeometricCallOption : public Option
{
public:
    GeometricCallOption();
    GeometricCallOption(const Param &P);
    ~GeometricCallOption();
    void print() const;

    double K;

    double payoff(const PnlMat *path_val, int t);
private:
    PnlVect *sigma;
    int size;
};

class BestOfOption : public Option
{
public:
    BestOfOption();
    BestOfOption(const Param &P);
    ~BestOfOption();
    void print() const;

    double K;
    PnlVect *lambda;

    double payoff(const PnlMat *path_val, int t);
};

class WorstOfOption : public Option
{
public:
    WorstOfOption();
    WorstOfOption(const Param &P);
    ~WorstOfOption();
    void print() const;

    double K;
    PnlVect *lambda;
```

```
    double payoff(const PnlMat *path_val, int t);
};
}
#endif
```