Help

```
#include <iostream>
#include <cmath>
#include <cstdlib>
#include <cstring>

using namespace std;

#include "
href../../../../common/math/ImportanceSampling_jl/src/AsianOption_h_src.pdfmath/
#include "
href../../../../common/math/jlparser/include/jlparser/parser_h_src.pdfjlparser/p



double AsianCallOption::payoff(const PnlMat *path_val)
{
  double sum = pnl_mat_scalar_prod(path_val, ones, lambda);
  sum /= path_val->m;
  return max(sum - K, 0.0);
}

AsianCallOption::AsianCallOption() : lambda(NULL), ones(NULL) { }

AsianCallOption::AsianCallOption(const Param &P) :
  BaseOption(P)
{
  label = "asian";
  P.extract("strike", K);
  if ((! P.extract("payoff coefficients", lambda, size, true)) && (size == 1))
    {
      lambda = pnl_vect_create_from_scalar(1, 1.);
    }
    ones = pnl_vect_create_from_scalar(nTimeSteps + 1, 1.);
}

void AsianCallOption::print() const
{
  cout << "**** Asian Option Characteristics ****" << endl;
  BaseOption::print();
```

```
  cout << " strike : " << K << endl;
  cout << " payoff coefficients : "; pnl_vect_print_asrow(lambda); cout << endl;
}

AsianCallOption::~AsianCallOption()
{
  pnl_vect_free(&lambda);
  pnl_vect_free(&ones);
}
```