```
#include <cstdio>
#include <iostream>
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <iostream>
#include "Event.hh"
#include "TFile.h"
#include "TTree.h"
#include "DHF.hh"
#include "Trace.hh"
#include "Align.hh"
#include "Calib.hh"
using namespace std;


int main(int argc,char** argv){          // Inizializzazione variabili
  char name[255];
  char filename[255];
  char dir1[255];
  int run;
  int align;
  int maxev=1000000;
  char scanname[20];
  char namein[255];
  char ffnom[255];

  //  sprintf(dir1,"/data4/RandD/wjb" );
  sprintf(dir1,"/data2/crystal/crystal_2/root" );
  //  sprintf(dir1,"/disk2/home/wjb/sim/p_cern/cc/code2" );

  if(argc==2){
    run=atoi(argv[1]);
    maxev= 99999999;
    align = 0;
  }
  else if(argc==3){
    run=atoi(argv[1]);
    maxev= atoi(argv[2]);
    align = 0;
  }
  else if(argc==4){
    run=atoi(argv[1]);
    maxev= atoi(argv[2]);
    align = atoi(argv[3]);
  }else{
    printf(" Usage %s   <run> [maxev] [align]\n",argv[0]);
    printf(" It seaches for files in the directory:\n");
    printf("  %s \n",dir1);
    printf("Write in ouput Houtdst_<run>.root\n");
    exit(1);
  }

  Align* alg=new Align(align,run);
  alg->LireFichAmasPar(run);
  alg->LireFonctionEta(run);
  //  if (alg->amaspar[1][3] == 3) alg->LireFonctionEta3(run);
  if (align > 1 && align < 6) alg->LireAlgPar();
  if (align == 6) alg->LireFichAlgPar(run);
  printf("run %d\n",run);
  if (run == 1239 || run == 1243 || run == 1244 || run == 1246 || run == 1247 ||
 run == 1263) alg->ppiste_n[3]=641.;
  for (int i=0; i<6; i++) printf("i %d premiere piste n %f\n",i,alg->ppiste_n[i]);

  Event* dst=new Event(); //nuova variabile dst

  sprintf(namein,"%s/run_00%d.root",dir1,run);
  printf("fichier %s\n",namein);
  TFile *f= new TFile(namein);
```

```

  //lettura TTree4
  TTree *t4=(TTree*) f->Get("t4");
  TBranch *bra=t4->GetBranch("cluster_branch");
  bra->SetAddress(&dst);
  Int_t ntot=t4->GetEntries(); // total number of events
  float pperc=0;
  float perc;
  int nentries= ((maxev<ntot)? maxev : ntot);

  RHClass *rh = ((RHClass *)t4->GetUserInfo()->First());
   printf("ntdrCmp %d\n",rh->ntdrCmp);
   for (int ii=0; ii<rh->ntdrCmp; ii++)
     //     printf(" ii %d tdrCmpMap %d\n",ii,rh->tdrCmpMap[ii]);
   printf(" ii %d tdrCmpMap %d pos %d\n",ii,rh->tdrCmpMap[ii],rh->FindPos(rh->tdrCmpMa
p[ii]));

  /* ici creer directoires et histogram en memoire Rint:/ */
  printf("run %d\n",run);
  sprintf(name,"Histos_%06d.root",run);
  DHF* Histos= new DHF(name,name,rh->ntdrCmp,rh->tdrCmpMap,run);
  Histos->Init(alg);
  Calib* cal=new Calib(run);
  cal->RepererCalibs(rh);
  Histos->hcalib(cal,rh);
  f->cd();

  cout<< " Number of entries : "<<ntot<< " maxev " <<maxev<<endl;

  int accept = 0;
  int daccept = 0;
  for (int ii=0;ii<nentries;ii++){
    perc=ii/(nentries*1.);
    //    printf("ii %d perc %f pperc %f\n",ii,perc,pperc);
    if (perc>=pperc){printf("Processed %5.0f%%\n",pperc*100);pperc+=0.1;}
    dst->Clear();
    t4->GetEntry(ii);
    dst->Evtnum = ii;
    accept = dst->trace(rh,alg);
    Histos->Fill1D("Accept",((float) accept),1.);
    Histos->FillAll(dst,rh,alg);
    if (accept >= 12) {
       daccept = Histos->divergences(dst,alg);
       //       if (daccept == 1) {
          Histos->Fill1D("Accept_ev",((float) dst->Evtnum),1.);
          Histos->FillSelect(dst,rh,alg);
          if (align == 5 || align == 6) Histos->residus(dst,alg);
          if (align == 6) {
             Histos->residus_petits(dst,alg);
             Histos->residus_echelles(dst,alg);
             Histos->residus_eff(dst,alg,0);
             Histos->residus_scan_va(dst,alg,cal);
          }
          //       }
    }
    else if (accept < 0 && align == 6) Histos->residus_eff(dst,alg,0);
  }

  f->Close();

  sprintf(ffnom,"Hist_%d.root",run);
  printf("ffnom %s align %d\n",ffnom,align);
  if (align == 1 || align == 5) Histos->align_xy_positions(alg);
  if (align == 2) Histos->align_ref_dyx_dxy(alg);
  if (align == 3) Histos->align_dxx_dyy(alg);
  if (align == 4) Histos->align_dxy_dyx(alg);
  if (align == 6) Histos->residus_eff(dst,alg,1);
  Histos->EcrireHistos(ffnom);
}
```