

# Pythia

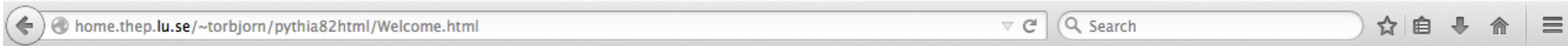


## Çalışması

Orhan Çakır (Ankara Ü. ve Aydın Ü.)

Sertaç Öztürk (Gaziosmanpaşa Ü.)

# Pythia8 Çevrimiçi Rehber



## PYTHIA 8

### Welcome to PYTHIA - The Lund Monte Carlo!

PYTHIA 8 is the successor to PYTHIA 6, rewritten from scratch in C++. At the release of the first public version, 8.100, it was untried and also still missed a few relevant features. This has changed over the years since then, and the release of 8.200 marks the end of a transition period. PYTHIA 8.2 has a significantly extended physics scope, notably for LHC physics, relative to what PYTHIA 6.4 could offer. There are only a few areas left, such as  $\gamma\gamma$  physics, where 6.4 still has a role to fill.

### Documentation

On these webpages you will find the up-to-date manual for PYTHIA 8.2. Use the left-hand index to navigate this documentation of program elements, especially of all possible program settings. All parameters are provided with sensible default values, however, so you need only change those of relevance to your particular study, such as choice of beams, processes and phase space cuts. The pages also contain a fairly extensive survey of all methods available to the user, e.g. to study the produced events. What is lacking on these webpages is an overview, on the one hand, and an in-depth physics description, on the other.

The overview can be found in the attached PDF file

#### **An Introduction to PYTHIA 8.2**

T. Sjöstrand et al, arXiv:1410.3012 [hep-ph].

You are strongly recommended to read this summary when you start out to learn how to use PYTHIA 8.2.

For the physics description we refer to the complete

#### **PYTHIA 6.4 Physics and Manual**

T. Sjöstrand, S. Mrenna and P. Skands, JHEP05 (2006) 026,

which in detail describes the physics (largely) implemented also in PYTHIA 8, and also provides a more extensive bibliography than found here. When you use PYTHIA 8.2, you should therefore cite both.

## PYTHIA 8 Index

### Program Overview

- Frontpage
- Program Flow
- Settings Scheme
- Particle Data Scheme
- Program Files
- Program Classes
- Program Methods
- Sample Main Programs

### Setup Run Tasks

- Save Settings**
- Main-Program Settings
- Beam Parameters
- Random-Number Seed
- PDF Selection
- Master Switches
- Process Selection
- QCD

<http://home.thep.lu.se/~torbjorn/pythia8html/Welcome.html>

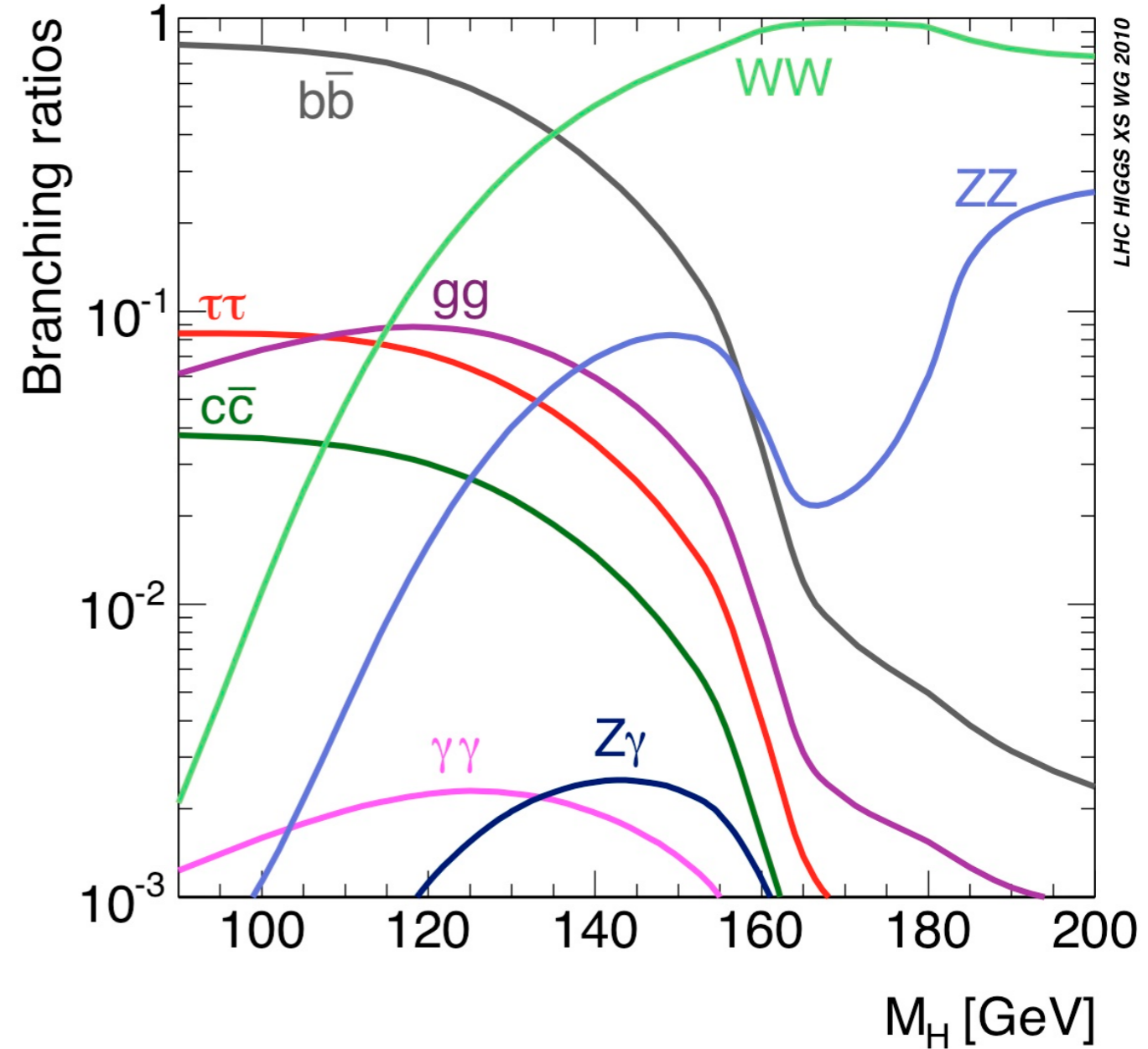
# Pythia8 I X Kurulumu

```
wget http://home.thep.lu.se/~torbjorn/pythia8/pythia8I75.tgz
tar xvfz pythia8I75.tgz
cd pythia8I75
./configure --enable-shared true
make
```

```
pwd
export PYTHIA8=.../pythia8I75
export PYTHIA8DATA=$PYTHIA8/xmldoc
```

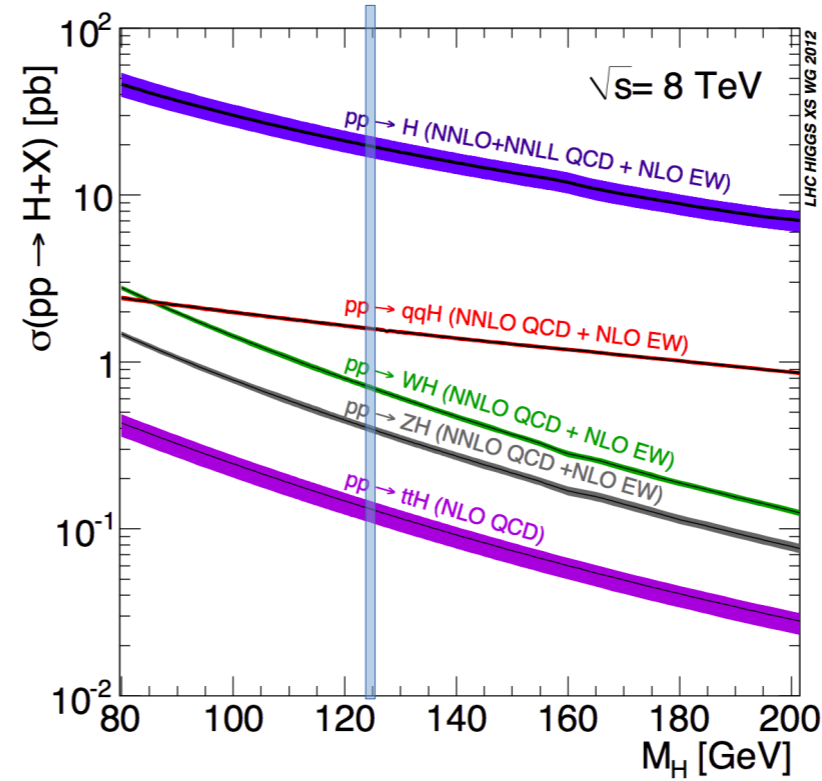
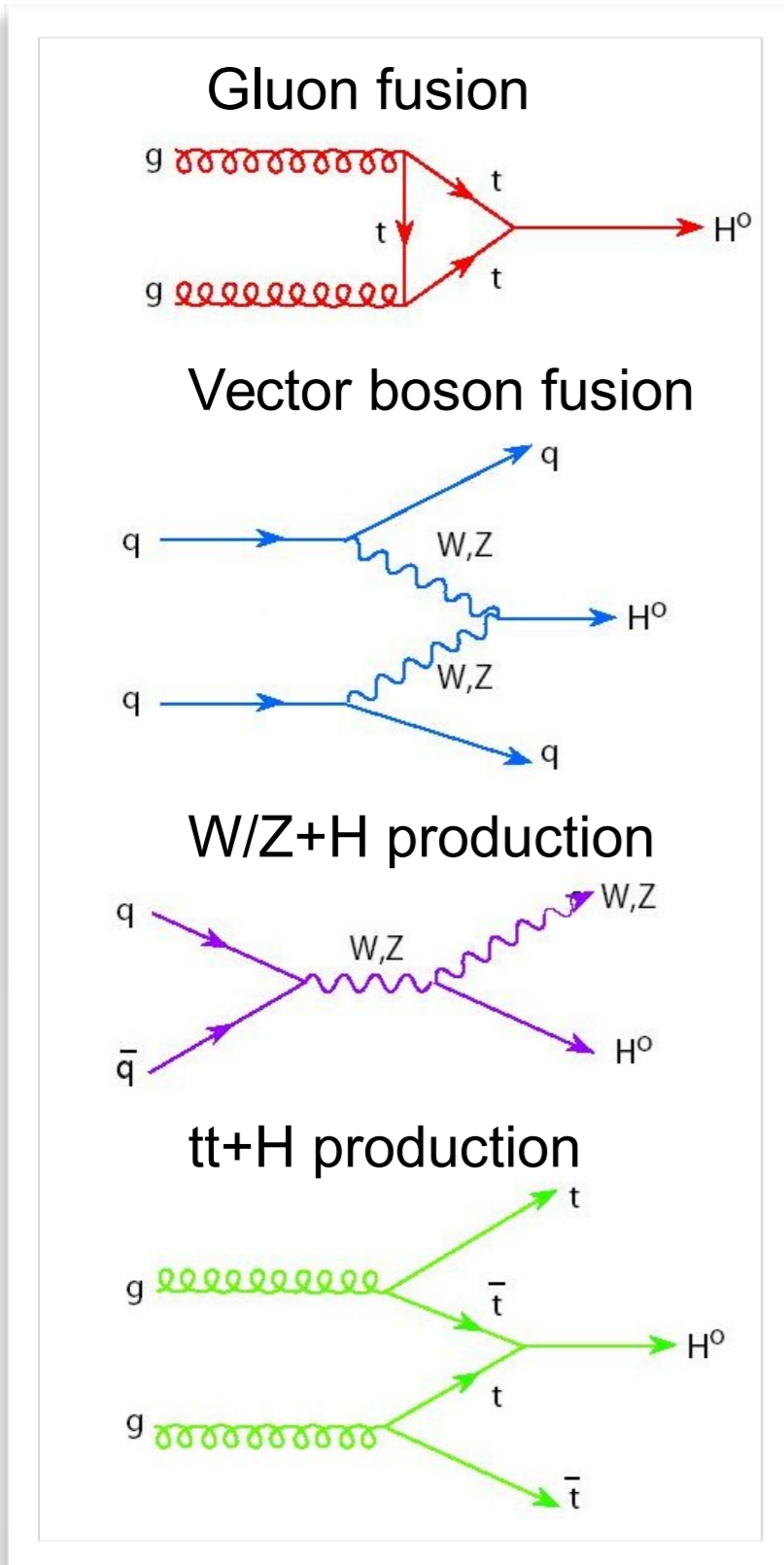
# Çalışma 1

Higgs bozonu dallanma oranlarının Higgs bozonu kütlesi ile değişiminin çizdirilmesi.



<http://www.quantumdiaries.org/wp-content/uploads/2011/08/higgsbr.jpg>

# Higgs Bozonu Üretimi



**Pythia8 Ayarları**

**HiggsSM:gg2H** (gg füzyon)

**HiggsSM:ff2Hff(t:ZZ)** (ZZ füzyon)

**HiggsSM:ff2Hff(t:WW)** (WW füzyon)

**HiggsSM:gg2Httbar** (t-tbar + H)

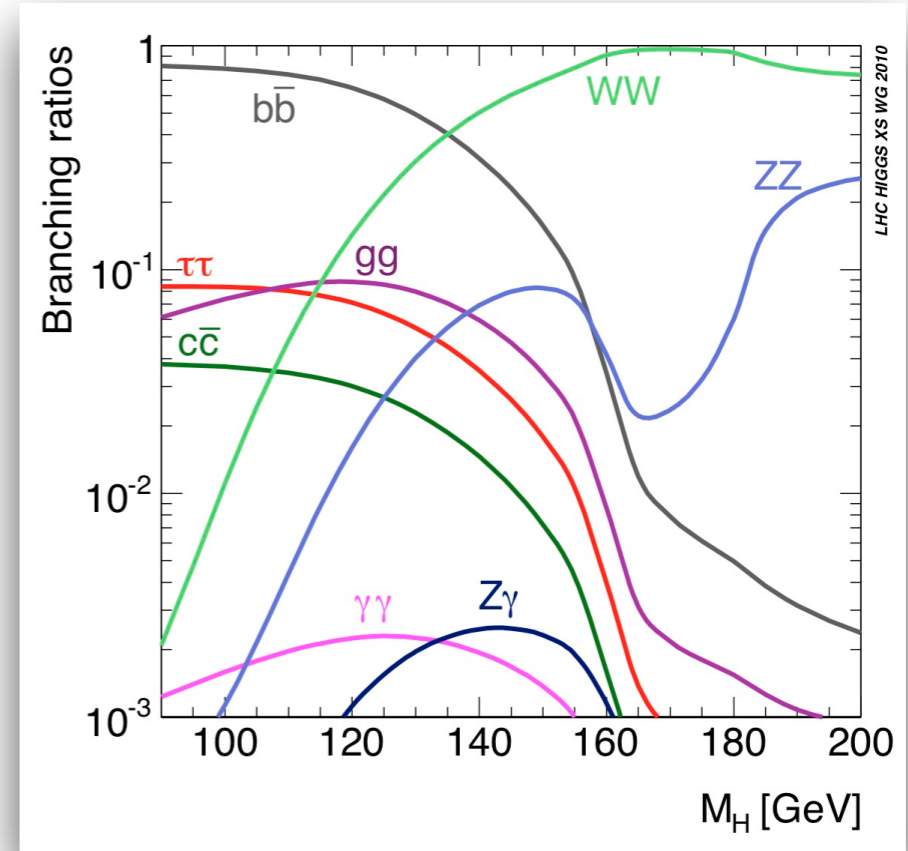
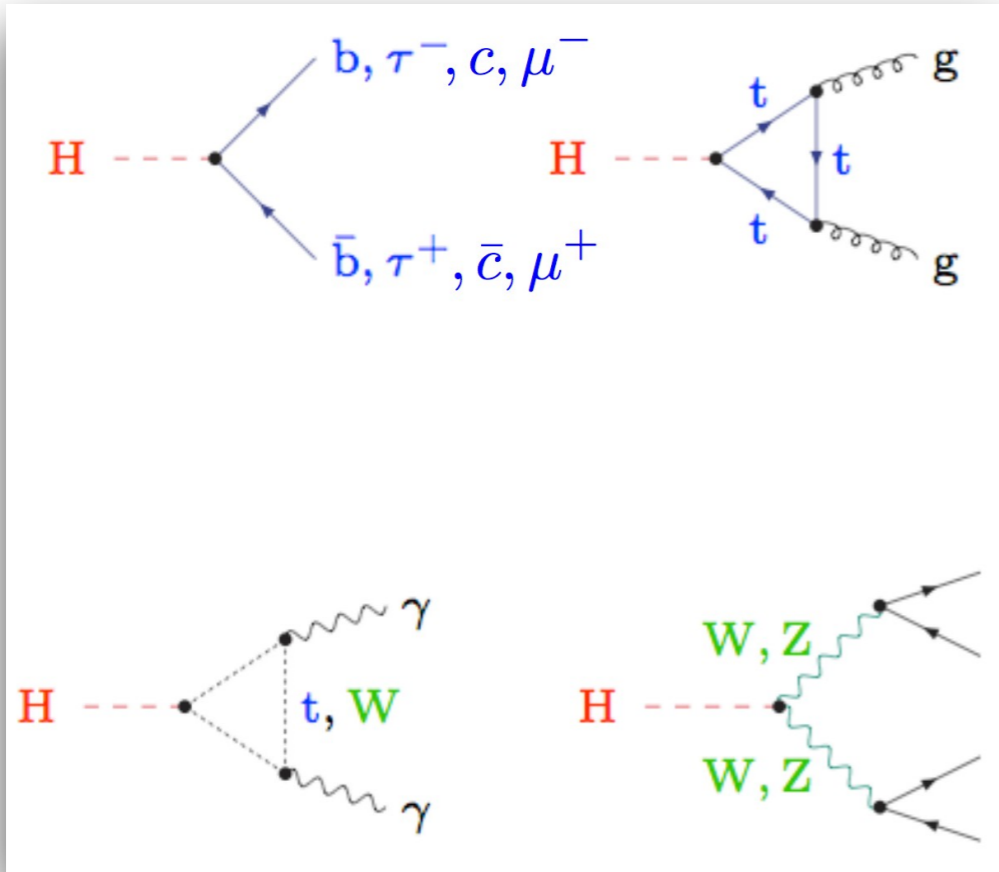
**HiggsSM:ffbar2HZ** (Z+ H)

**HiggsSM:ffbar2HW** (W+ H)

**HiggsSM:all** (Bütün süreçler)



# Higgs Bozonu Bozunumu



## Pythia8 Ayarları

25:m0 = 125.  
 25:onMode = off

25:onlyMatch = 22 22 ( H →  $\gamma\gamma$  )  
 25:onlyAny = 22 ( H →  $\gamma\gamma$  ve H →  $Z\gamma$  )

QUARKS		LEPTONS		GAUGE AND HIGGS BOSONS	
$d$	1	$e^-$	11	$g$	(9) 21
$u$	2	$\nu_e$	12	$\gamma$	22
$s$	3	$\mu^-$	13	$Z^0$	23
$c$	4	$\nu_\mu$	14	$W^+$	24
$b$	5	$\tau^-$	15	$h^0/H_1^0$	25
$t$	6	$\nu_\tau$	16	$Z'/Z_2^0$	32
$b'$	7	$\tau'^-$	17	$Z''/Z_3^0$	33
$t'$	8	$\nu_{\tau'}$	18	$W'/W_2^+$	34
				$H^0/H_2^0$	35
				$A^0/H_3^0$	36
				$H^+$	37

## PDG Kodları

# Tesir Kesiti Hesabı

example/main01.cc

```
#include "Pythia8/Pythia.h"
using namespace Pythia8;
int main() {

    Pythia pythia;

    // Olay üretim ayarları
    pythia.readString("HiggsSM:all = on");
    pythia.readString("25:m0 = 125.");

    // Çarpışma ayarları
    pythia.init(2212, 2212, 14000 );

    / Olay döngüsü. Olay üret. Eğer hata olursa
    atla.
    for (int iEvent = 0; iEvent < 10; ++iEvent) {
        if (!pythia.next()) continue;

    // Olay döngüsünün sonu.
    }
    pythia.stat();

    return 0;
}
```

Üretim tesir kesiti  
 $\sigma(pp \rightarrow H)$

```
*----- PYTHIA Event and Cross Section Statistics -----*
```

Subprocess	Code	Number of events			sigma +- delta (estimated) (mb)	
		Tried	Selected	Accepted		
<b>10 olay</b>						
f fbar -> H (SM)	901	4	1	1	9.572e-12	9.572e-12
g g -> H (SM)	902	105	7	7	1.004e-10	2.106e-11
f fbar -> H0 Z0 (SM)	904	5	1	1	6.407e-12	6.407e-12
f fbar -> H0 W+- (SM)	905	15	0	0	0.000e+00	0.000e+00
f f' -> H0 f f' (Z0 Z0 fusion) (SM)	906	15	0	0	0.000e+00	0.000e+00
f_1 f_2 -> H0 f_3 f_4 (W+ W- fusion) (SM)	907	44	1	1	1.685e-11	1.685e-11
g g -> H t tbar (SM)	908	0	0	0	0.000e+00	0.000e+00
q qbar -> H t tbar (SM)	909	0	0	0	0.000e+00	0.000e+00
sum		188	10	10	1.332e-10	2.933e-11

```
*----- End PYTHIA Event and Cross Section Statistics -----*
```

```
*----- PYTHIA Event and Cross Section Statistics -----*
```

Subprocess	Code	Number of events			sigma +- delta (estimated) (mb)	
		Tried	Selected	Accepted		
<b>1000 olay</b>						
f fbar -> H (SM)	901	378	34	34	3.827e-12	5.324e-13
g g -> H (SM)	902	8356	705	705	9.069e-11	2.440e-12
f fbar -> H0 Z0 (SM)	904	564	51	51	7.018e-12	4.891e-13
f fbar -> H0 W+- (SM)	905	1299	96	96	1.360e-11	7.099e-13
f f' -> H0 f f' (Z0 Z0 fusion) (SM)	906	1497	32	32	4.072e-12	3.219e-13
f_1 f_2 -> H0 f_3 f_4 (W+ W- fusion) (SM)	907	4417	82	82	1.249e-11	6.763e-13
g g -> H t tbar (SM)	908	2	0	0	0.000e+00	0.000e+00
q qbar -> H t tbar (SM)	909	55	0	0	0.000e+00	0.000e+00
sum		16568	1000	1000	1.317e-10	2.746e-12

```
*----- End PYTHIA Event and Cross Section Statistics -----*
```

```
*----- PYTHIA Event and Cross Section Statistics -----*
```

Subprocess	Code	Number of events			sigma +- delta (estimated) (mb)	
		Tried	Selected	Accepted		
<b>10000 olay</b>						
f fbar -> H (SM)	901	3707	291	291	3.657e-12	1.556e-13
g g -> H (SM)	902	81843	6865	6865	9.018e-11	7.818e-13
f fbar -> H0 Z0 (SM)	904	5596	435	435	6.255e-12	1.529e-13
f fbar -> H0 W+- (SM)	905	13106	1090	1090	1.428e-11	2.278e-13
f f' -> H0 f f' (Z0 Z0 fusion) (SM)	906	14821	360	360	4.349e-12	1.116e-13
f_1 f_2 -> H0 f_3 f_4 (W+ W- fusion) (SM)	907	43785	955	955	1.275e-11	1.966e-13
g g -> H t tbar (SM)	908	27	0	0	0.000e+00	0.000e+00
q qbar -> H t tbar (SM)	909	558	4	4	3.075e-14	4.968e-15
sum		163443	10000	10000	1.315e-10	3.728e-13

```
*----- End PYTHIA Event and Cross Section Statistics -----*
```

# Tesir Kesiti Hesabı

```
#include "Pythia8/Pythia.h"
using namespace Pythia8;
int main() {

    Pythia pythia;

    // Olay üretim ayarları
    pythia.readString("HiggsSM:all = on");
    pythia.readString("25:m0 = 125.");
    pythia.readString("25:onMode = off");
    pythia.readString("25:onIfMatch = 22 22");

    // Çarpışma ayarları
    pythia.init(2212, 2212, 14000);

    // Olay döngüsü. Olay üret. Eğer hata
    // olursa atla.
    for (int iEvent = 0; iEvent < 100; ++iEvent)
    {
        if (!pythia.next()) continue;

    // Olay döngüsünün sonu.
    }
    pythia.stat();

    return 0;
}
```

Subprocess	Code	Number of events			sigma +- delta (estimated) (mb)	
		Tried	Selected	Accepted		
f fbar -> H (SM)	901	3804	271	271	1.013e-14	4.355e-16
g g -> H (SM)	902	83055	6953	6953	2.558e-13	2.204e-15
f fbar -> H0 Z0 (SM)	904	5768	478	478	1.746e-14	4.233e-16
f fbar -> H0 W+ (SM)	905	13029	1044	1044	4.048e-14	6.392e-16
f f' -> H0 f f'(Z0 Z0 fusion) (SM)	906	15154	305	305	1.187e-14	3.053e-16
f_1 f_2 -> H0 f_3 f_4 (W+ W- fusion) (SM)	907	44508	948	948	3.565e-14	5.179e-16
g g -> H t tbar (SM)	908	28	0	0	0.000e+00	0.000e+00
q qbar -> H t tbar (SM)	909	558	1	1	5.555e-17	5.555e-17
sum		165904	10000	10000	3.714e-13	2.449e-15

$$\text{Üretim tesir kesiti} \quad \times \quad \text{Dallanma oranı}$$
$$\sigma(pp \rightarrow H) \quad \times \quad BR(H \rightarrow \gamma\gamma)$$

$$M_h = 125 \text{ GeV için } BR(H \rightarrow \gamma\gamma) = 3.714 \cdot 10^{-13} / 1.315 \cdot 10^{-10} \\ = 0.00282$$



```

#include "Pythia8/Pythia.h"
using namespace Pythia8;
int main() {

    Pythia pythia;

    // Dışardan dosyayı oku
    pythia.readFile("main01.cmnd");

    int nEvent = pythia.mode("Main:numberOfEvents");

    pythia.init();


    // Olay döngüsü. Olay üret. Eğer hata olursa atla.
    for (int iEvent = 0; iEvent < nEvent; ++iEvent) {
        if (!pythia.next()) continue;

        // Olay döngüsünün sonu.
    }
    pythia.stat();

    return 0;
}

```

main01.cc



```

! main01.cmnd.

! Ana program ayarları.
Main:numberOfEvents = 1000          ! olay sayısı

! Demet ve çarpışma parametreleri
Beams:idA = 2212                    ! birinci ışın, p = 2212
Beams:idB = 2212                    ! ikinci ışın, p = 2212
Beams:eCM = 14000.                  ! kütle merkezi enerjisi

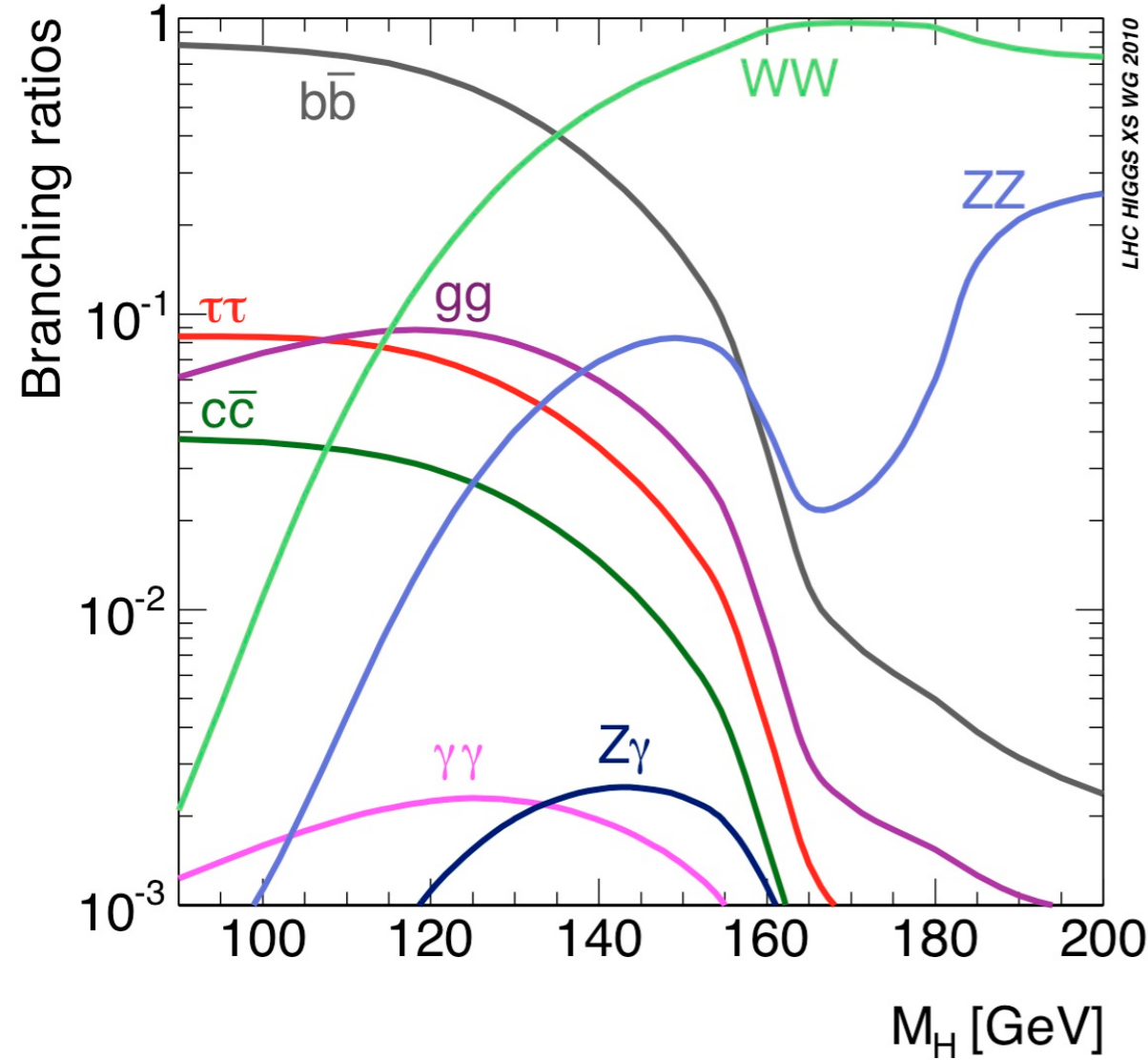
! Higgs bozonu üretim ve bozunum ayarları
HiggsSM:all = on
25:m0 = 125.
25:onMode = off
25:onIfMatch = 22 22

! Olay ayarları
#PartonLevel:MPI = off              ! multiparton etkileşmesi
#PartonLevel:ISR = off              ! ilk-durum radyasyonu
#PartonLevel:FSR = off              ! son-durum radyasyonu
#HadronLevel:Hadronize = off        ! hadronizasyon
#HadronLevel:Decay = off            ! bozunum
#PDF:pSet = 8                       ! PDF seçimi

```

main01.cmnd

# Higgs Dallanma Oranı



Higgs kütesi deęerlerinin seęilmesi

- [100, 115, 130, 145, 160, 175, 190, 205] GeV

Bir bozonum kanalının seęilmesi

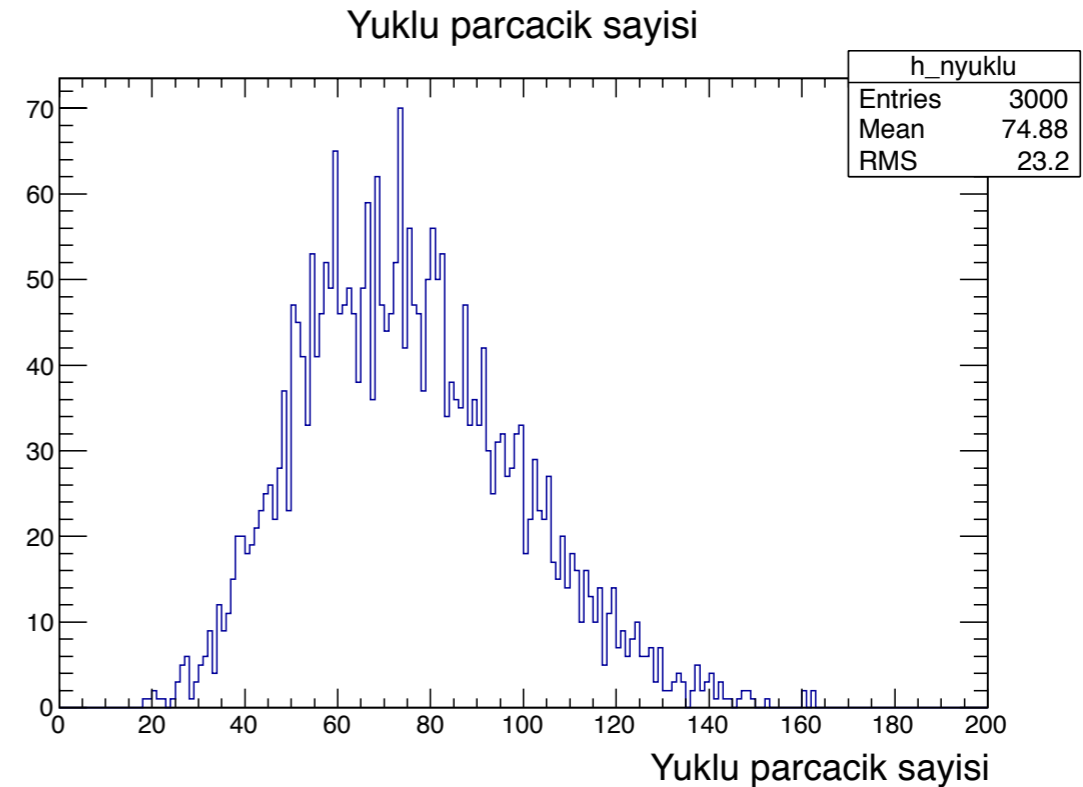
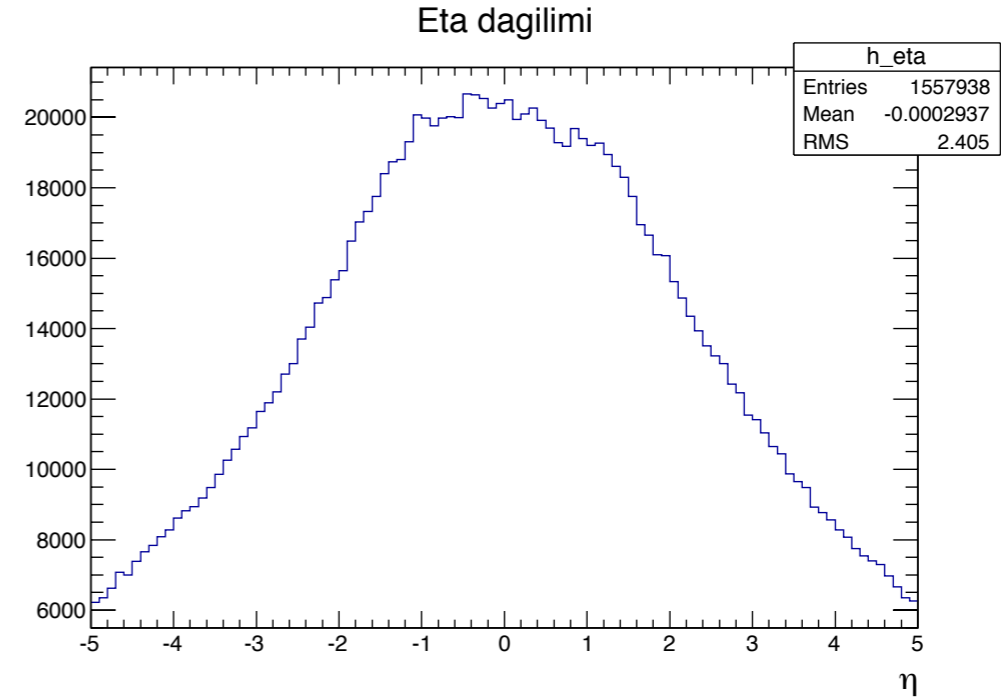
ROOT ile elde edilen deęerlerin grafik olarak çizdirilmesi

# Çalışma 2

Olay üretme ve dağılımları histogram olarak çizdirme.

- Son durum parçacıkların eta dağılımı
- Son durum  $p_T > 1$  GeV olan yüklü parçacık sayısı dağılımı

```
cd rootexample  
make hist  
./hist.exe
```



```

// Stdlib kütüphane dosyası
#include <iostream>

// Pythia8 programına erişim için kütüphane dosyası
#include "Pythia8/Pythia.h"

// ROOT kütüphane dosyaları
#include "TH1.h"
#include "TH2.h"
#include "TVirtualPad.h"
#include "TApplication.h"
#include "TCanvas.h"
#include "TFile.h"

using namespace Pythia8;

int main(int argc, char* argv[]) {

    // Root uygulama ortamının ayarlanması.
    TApplication theApp("hist", &argc, argv);

    TFile* outFile = new TFile("hist.root", "RECREATE");

    // Pythia8 olay ayarlarının yapılması
    Pythia pythia;
    pythia.readString("HardQCD:all = on");
    pythia.readString("PhaseSpace:pTHatMin = 200 ");
    pythia.readString("PhaseSpace:pTHatMax = 300 ");

    // Çarpışma ayarları
    pythia.init( 2212, 2212, 14000 );

    // Histogram ayarları
    TH1F *h_eta = new TH1F("h_eta", "Eta dagilimi", 100, -5, 5);
    TH1F *h_nyuklu = new TH1F("h_nyuklu", "Yuklu parcacik
sayisi", 200, 0, 200);

```

```

// Olay döngüsüne başla. Eğer hata varsa olayı atla.
for (int iEvent = 0; iEvent < 3000; ++iEvent) {
    if (!pythia.next()) continue;

    int n_par = 0;

    for (int i = 0; i < pythia.event.size(); ++i){
        // Sadece son durumdaki parçacıkları seç
        if (!pythia.event[i].isFinal()) continue;

        h_eta->Fill(pythia.event[i].eta());

        // pT > 1 GeV olan yüklü parçacıkları seç
        if (pythia.event[i].isCharged() && pythia.event[i].pT()>1.0)
            n_par++;
    }

    h_nyuklu->Fill(n_par);
}

// Histogramları kayıt et.
outFile->cd();
h_eta->Write();
h_nyuklu->Write();
outFile->Close();

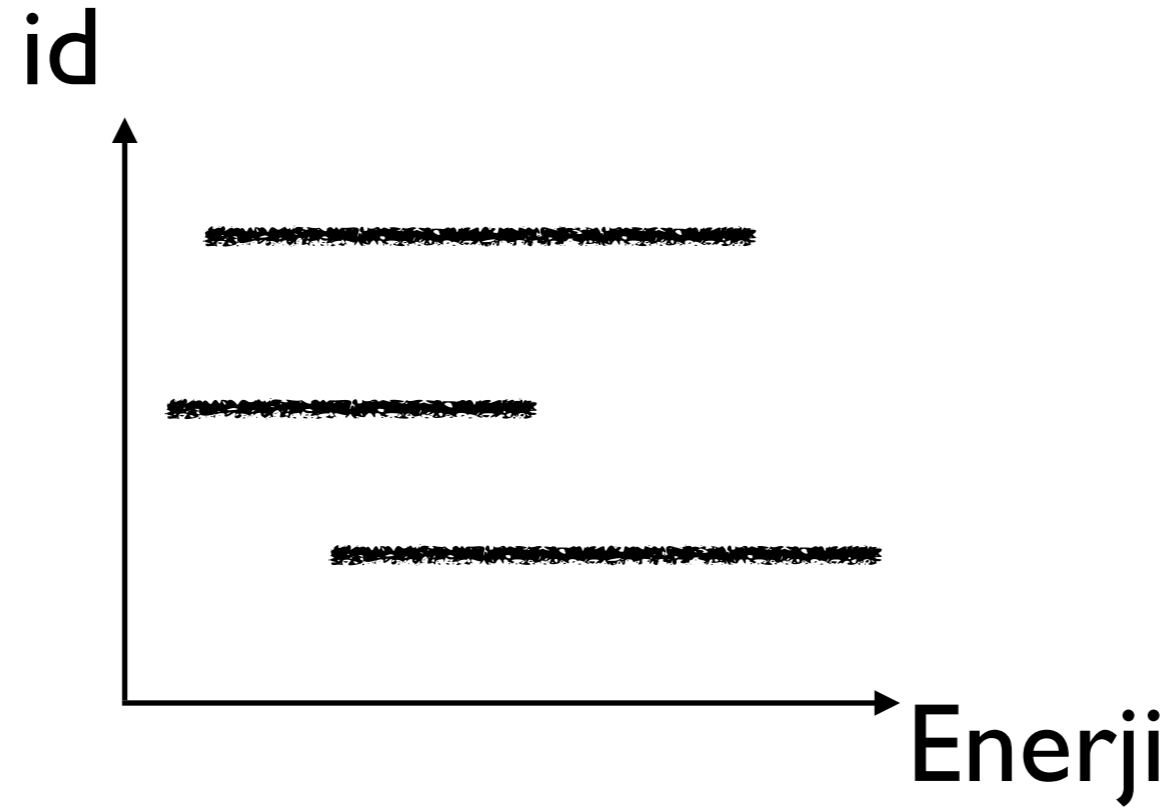
// İstatistik
pythia.stat();

// Son.
return 0;
}

```

# Ödev - En Yüksek Enerjili Hadron

- ✓ Olay içerisinde en yüksek enerjili son durum hadron parçacığının PDG kod numarasının ve enerjisinin 2-B histogramını çizdirin.





# Çalışma3

Üretilen bir LHE dosyasının pythia8 ile hadronize edilmesi.

LHE : Les Houches Event

Olay ve süreç bilgisini parton seviyesindeki çıktı dosyasıdır.

- MadGraph, CalcHEP/CompHEP, Pythia

Başka bir olay üreticilerinde elde edilen bir LHE dosyası, Pythia' da girdi olarak kullanılarak hadronizasyona uğrattılır.

# LHE Dosya Formatı

```
<LesHouchesEvents version="1.0">
```

```
<init>
```

```
  2212  2212  0.350000000000E+04  0.350000000000E+04  0 0 10042 10042 3 2  
  0.46446972880E+04  0.17139268058E+02  0.774890000000E+01  1  
  0.31042443040E+04  0.11454885420E+02  0.774890000000E+01  2
```

```
</init>
```

parçacık sayısı

```
<event>
```

```
5 2 0.7748900E+01 0.8040000E+02 0.7957747E-01 0.1297830E+00  
-2 -1 0 0 0 501 0.000000000000E+00 0.000000000000E+00 0.34637450368E+01 0.34637450368E+01 0.000000000000E+00 0. 1.  
1 -1 0 0 501 0 0.000000000000E+00 0.000000000000E+00 -0.48594881400E+03 0.48594881400E+03 0.000000000000E+00 0. -1.  
-24 2 1 2 0 0 0.000000000000E+00 0.000000000000E+00 -0.48248506897E+03 0.48941255904E+03 0.82053709061E+02 0. 0.  
11 1 3 3 0 0 0.51230888840E+01 -0.14990768056E+02 -0.46696969867E+03 0.46723834243E+03 0.000000000000E+00 0. -1.  
-12 1 3 3 0 0 -0.51230888840E+01 0.14990768056E+02 -0.15515370300E+02 0.22174216608E+02 0.000000000000E+00 0. 1.
```

```
</event>
```

PDG kod

durum

-1: ilk durum  
1: son durum  
2: ara durum

ebeveyin

renk akısı

4-vektör

kütle

spin

```

// Stdlib kütüphane dosyası
#include <iostream>

// Pythia8 programına erişim için kütüphane dosyası
#include "Pythia8/Pythia.h"

// ROOT kütüphane dosyaları
#include "TH1.h"
#include "TH2.h"
#include "TVirtualPad.h"
#include "TApplication.h"
#include "TCanvas.h"
#include "TFile.h"

using namespace Pythia8;

int main(int argc, char* argv[]) {

// Root uygulama ortamının ayarlanması.
TApplication theApp("hist", &argc, argv);

TFile* outFile = new TFile("hist.root", "RECREATE");

// LHE dosyasının okunması
Pythia pythia;
pythia.readString("Beams:frameType= 4");
pythia.readString("Beams:LHEF = test.lhe ");

// Çarpışma ayarları
pythia.init( );

// Histogram ayarları
TH1F *h_eta = new TH1F("h_eta","Eta dagilimi", 100, -5, 5);
TH1F *h_nyuklu = new TH1F("h_nyuklu","Yuklu parcacik sayisi",
200, 0, 200);

```

```

// Olay döngüsüne başla. Eğer hata varsa olayı atla.
for (int iEvent = 0; ; ++iEvent) {
if (!pythia.next()) continue;

int n_par = 0;

for (int i = 0; i < pythia.event.size(); ++i){
// Sadece son durumdaki parçacıkları seç
if (!pythia.event[i].isFinal()){
// Dosyanın sonuna geldiğinde döngüden çık.
if (pythia.info.atEndOfFile()) break;
continue;}

h_eta->Fill(pythia.event[i].eta());
// pT > 1 GeV olan yüklü parçacıkları seç
if (pythia.event[i].isCharged() && pythia.event[i].pT()>1.0)
n_par++;
}

h_nyuklu->Fill(n_par);
}

// Histogramları kayıt et.
outFile->cd();
h_eta->Write();
h_nyuklu->Write();
outFile->Close();

// İstatistik.
pythia.stat();

// Son.
return 0;
}

```

```

// Stdlib kütüphane dosyası.
#include <iostream>

// Pythia8 programına erişim için kütüphane dosyası.
#include "Pythia8/Pythia.h"

using namespace Pythia8;

int main(int argc, char* argv[]) {

// Pythia8 olay ayarlarının yapılması
Pythia pythia;
pythia.readString("HardQCD:gg2bbbar = on");
pythia.readString("HardQCD:qqbar2bbbar = on");
pythia.readString("PhaseSpace:pTHatMin = 200 ");
pythia.readString("PhaseSpace:pTHatMax = 300 ");

// Çarpışma ayarları
pythia.init( 2212, 2212, 14000 );

// LHA objesi oluştur.
LHAupFromPYTHIA8 myLHA(&pythia.process, &pythia.info); //
Olaydaki parton seviyesi bilgisi
// LHAupFromPYTHIA8 myLHA(&pythia.event, &pythia.info); //
Olaydaki tüm parçacık bilgisi

// LHE dosyası oluştur
myLHA.openLHEF("pythiaLHE_qcd.lhe");

// LHA dosyasına başlangıç bilgisini kayıt et ve yaz.
myLHA.setInit();
myLHA.initLHEF();

// Olay döngüsü
for (int iEvent = 0; iEvent< 100 ; ++iEvent) {
// Generate an event.
pythia.next();
}
}

```

```

// Olay bilgisini kayıt et ve yaz.
myLHA.setEvent();
myLHA.eventLHEF();

}

// Statistics: full printout.
pythia.stat();

// Tesir kesiti değerini günceller
myLHA.updateSigma();

// LHE dosyasını kapat.
myLHA.closeLHEF(true);

// Son.
return 0;
}

```



**Pythia8 de LHE  
dosyası üretilebilir.**