```cpp
#include<iostream>
#include<cassert>
#include"header.h"

using namespace std;

StatClass::StatClass(unsigned aSize):mSize(aSize),mCurrentPosition(0){
  mpData=new unsigned[mSize];
  assert(mpData);
}

StatClass::~StatClass(){
  delete [] mpData;
}

void StatClass::Insert(unsigned aItem){
  (*this)<<aItem;
}

StatClass& StatClass::operator<<(unsigned aItem){
  //se la posizione corrente coincide con la dimensione massima
  //allora si deve allocare piu' spazio
  if(mCurrentPosition >= mSize){
    unsigned size=mSize*2;
    unsigned * pdata=new unsigned[size];
    assert(pdata);
    //copia del vettore dati
    for(unsigned i=0;i<mSize;i++) pdata[i]=mpData[i];
    //si distrugge la vecchia copia
    delete [] mpData;
    //si sostituisce il puntatore
    mpData=pdata;
    mSize=size;
  }
  mpData[mCurrentPosition++]=aItem;

  return *this;
}

double StatClass::Average(){
  double average=0;
  for(unsigned i=0;i<mCurrentPosition;i++) average+=mpData[i];
  average/=mCurrentPosition;
  return average;
}

unsigned StatClass::Max(){
  unsigned max=mpData[0];
  for(unsigned i=1;i<mCurrentPosition;i++)
   if (max<mpData[i]) max=mpData[i];
  return max;
}

void StatClass::PrintHistogram()
{
 unsigned max=Max();
```

```
 unsigned* histo=new unsigned[max+1];
 for(unsigned i=0;i<=max;i++)
  histo[i]=0;

 for(unsigned i=0;i<mCurrentPosition;i++)
   histo[mpData[i]]++;

 for(unsigned i=0;i<=max;i++)
 {
   cout<<i<<" ";
   for (unsigned j=0;j<histo[i];++j) cout<<"*";
   cout<<endl;
 }
}
```