```cpp
#include "event.h"

 /* CONSTRUCTEURS */

  // Constructeur simple
  event::event(){
    label="";
    occured=false;
    nb_occurence=0;
    date.clear();
  }

  // Constructeur évolué, créé l'event à partir des données
  event::event(string nom){
    label=nom;
    occured=false;
    nb_occurence=0;
    date.clear();
  }

  /* OPERATEURS */

  event event::operator =(const event &e){
    label=e.label;
    occured=e.occured;
    nb_occurence=e.nb_occurence;
    date=date;
    return *this;
  }

  /* ASSESSEURS */

  // Assesseur, ressort le nom de l'event
  string event::get_label(){
    return label;
  }

  // Assesseur, ressort l'état de l'event (occured ou non)
  bool event::get_occured(){
    return occured;
  }

  // Assesseur, ressort le nombre d'occurence
  int event::get_nb_occurence(){
    return nb_occurence;
  }

  // Assesseur, ressort le tableau des heures d'occurence
  std::vector<int> event::get_h_occur(){
    return date;
  }

  // Assesseur, ressort l'heure de la dernière occurence
  int event::get_last_h_occur(){
    return date.back();
  }

  // Mutateur, change le label de l'élément
  void event::set_label(string name){
    label = name;
  }

  // Mutateur, change la valeur du nombre de validation de la chronique
  void event::set_occurence(int nb){
    nb_occurence = nb;
  }
```

```
67
68    // Mutateur, change la valeur du booleen occured
69    void event::set_occured(bool occur){
70      occured = occur;
71    }
72
73    // Mutateur, ajoute une valeur dans les heures d'apparition de l'event
74    void event::set_h_event(int heure){
75      date.push_back(heure);
76    }
77
78      // Reinitialisateur, remet à 0 le tableau des heures de validation
79    void event::reset_h_event(){
80      date.clear();
81    }
82
83    // Reinitialisateur, réinitialise tous les attributs
84    void event::reinit_event_all(){
85      label="";
86      occured=false;
87      nb_occurence=0;
88      date.clear();
89    }
90
91    /* METHODES */
92
93    // Affiche certains éléments dans la console
94    void event::afficher(){
95      cout<<"nom = "<<label<<endl;
96      cout<<"event occured = "<<occured<<endl;
97      if (occured) {
98          cout<<"nombre d'occurence = "<<nb_occurence<<endl;
99          cout<<"derniere validation = "<<date.back()<<endl;
100     }
101   }
102
103   // Affiche les heures d'occurence de l'event dans le terminal
104   void event::afficher_heures_occur(){
105     vector<int>::iterator it; // iterateur
106     for (it = date.begin(); it != date.end() ; it++)
107     {
108         cout<<"occurence à "<<(*it)<<endl;
109     }
110   }
111
112   // Affiche le nombre de fois où l'event est arrivé
113   void event::afficher_nb_occurence(){
114     cout<<"nombre d'occurence = "<<nb_occurence<<endl;
115   }
```