```
/*--- hw31p1.cpp ---*/
#include <iostream>
#include <string>

using namespace std;

string to_string(int v);

int main()
{
  int n;
  cout << "How many cards? ";
  cin >> n;

  cout << "Enter card values: ";
  int* cards = new int[n];
  for(int i=0; i < n; i++)
    cin >> cards[i];

  for(int i=0; i < n; i++)
    cout << to_string(cards[i]) << "  ";
  cout << endl;

  return 0;
}

string to_string(int v)
{
  // note: clubs="\u2663", diamonds="\u2666",
  //       hearts="\u2665", spades="\u2660"
  // initialize the suits
  string suits[5] = {" ", "\u2663", "\u2666", "\u2665", "\u2660"};

  // initialize the faces
  string faces[15] = {"  ", "  ", " 2", " 3", " 4",
                      " 5", " 6", " 7", " 8", " 9",
                      "10", " J", " Q", " K", " A"};

  return faces[v % 100] + suits[v / 100] ;
}

/*--- hw31p2.cpp ---*/
// hw2.cpp
#include "vector.h"

int main()
{
  char c;
  Vector x, y, z;
  cin >> c >> c >> x
      >> c >> c >> y
      >> c >> c >> z;

  cout << "X-Y+Z = " << x-y+z << endl;

  return 0;
}

/*--- vector.cpp ---*/
#include "vector.h"


Vector operator-(Vector a, Vector b)
{
  Vector C;
  C.vals[0] = a.vals[0] - b.vals[0];
  C.vals[1] = a.vals[1] - b.vals[1];
  C.vals[2] = a.vals[2] - b.vals[2];
  C.vals[3] = a.vals[3] - b.vals[3];
  return C;
}

Vector operator+(Vector a, Vector b)
{
  Vector C;
  C.vals[0] = a.vals[0] + b.vals[0];
  C.vals[1] = a.vals[1] + b.vals[1];
  C.vals[2] = a.vals[2] + b.vals[2];
  C.vals[3] = a.vals[3] + b.vals[3];
  return C;
}



istream& operator>>(istream &in, Vector &a)
{
  char c;
  return in >> c >> a.vals[0] >> c >> a.vals[1] >> c >> a.vals[2] >> c >> a.vals[3]
 >> c;
}
ostream& operator<<(ostream &out, Vector a)
{
  return out << '(' << a.vals[0] << ", " << a.vals[1] << ", " << a.vals[2] << ", "
<< a.vals[3] << ")";
}

/*--- vector.h ---*/
#include <iostream>
#include <string>
#include <cmath>
using namespace std;

struct Vector
{
  int vals[4];
};

Vector operator-(Vector a, Vector b);
Vector operator+(Vector a, Vector b);

istream& operator>>(istream &in, Vector &a);
ostream& operator<<(ostream &out, Vector a);
```