```
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;

// Add all occurances
int CheckSingles(vector<int>& valueDice, int goal) {
   int sum = 0;
   for(int i = 0 ; i < valueDice.size() ;i++){
      if(valueDice[i] == goal) sum += valueDice[i];
   }
return sum;
}

// Check for three of a kind (score = 30)
int CheckThreeOfKind(vector<int>& valueDice) {
   int start = 0;
   int end = 2;
   while(end < valueDice.size()){
      if(valueDice[start] == valueDice[end]){
         return 30;
      }
      start++;
      end++;
   }
return -1;
}


// Check for four of a kind (score when = 40)
int CheckFourOfKind(vector<int>& valueDice) {
int start = 0;
int end = 3;
while (end < valueDice.size())
{
      if (valueDice[start] == valueDice[end])
      {
         return 40;
      }
      start++;
      end++;
}
return -1;
}

// Check for five of a kind (score when = 50)
int CheckFiveOfKind(vector<int>& valueDice) {
   int n = valueDice.size();
   if(valueDice[0] == valueDice[n-1]) return 50;
return -1;
}

// Check for full house (score when when = 35)
int CheckFullHouse(vector<int>& valueDice) {
   int count[6] = {0, 0, 0, 0, 0, 0}; //ie. 4 4 5 5 5
   for(int i = 0; i < 5; i++) {
      int n = valueDice[i];
      count[n-1] += 1;
   }
   for (int i = 0; i < 6; i++) {
      if(!(count[i] == 0 || count[i] == 2 || count[i] == 3)) {
         return -1;
   }
   }
   return 35;
}

// Check for straight (score when = 45)
int CheckStraight(vector<int>& valueDice) {
   for(int i = 0 ; i < valueDice.size()-1 ; i++){
      if(valueDice[i] != valueDice[i+1]) continue;
      else return -1;
   }
return 45;
}

// Find highest score
int FindHighScore(vector<int>& valueDice) {
   int ans = 0;
   // CheckSingles
   for (int i = 1; i <= 6; i++)
   {
         ans = max(ans, CheckSingles(valueDice, i));
   }
   //Check for 3 ,4 and 5 and store max;
   ans = max({ans,CheckThreeOfKind(valueDice), CheckFourOfKind(valueDice), CheckFiveOfKind(valueDice)});
   // //Check for Full house
   ans = max(ans,CheckFullHouse(valueDice));
   //Check for straight
   ans = max(ans,CheckStraight(valueDice));
   return ans;
}

int main() {
vector<int> valueDice(5);
int highScore = 0;

// Fill array with five values from input
for(int i = 0; i < 5; ++i) {
cin >> valueDice.at(i);
}

// Place values in ascending order
sort(valueDice.begin(), valueDice.end());

// Find high ouput and score
highScore = FindHighScore(valueDice);
cout << "High score: " << highScore << endl;

return 0;
}
```