```cpp
#include <iostream>
#include <vector>
#include <climits>
// T(i,l) = "your turn now, with subarray starting at i with length l"
// T(i,l) = max( pick first or last , then minimum of other players options)

int main() {
   std::ios_base::sync_with_stdio(false);

   int t(0);
   std::cin >> t;
   for(int tt=0;tt<t;tt++) {
      int n(0),m(0),k(0);
      std::cin >> n >> m >> k;

      std::vector<int> x;
      x.resize(n);
      for(int i=0; i<n; i++) {
         std::cin >> x[i];
      }

      int beforeTurns(k);
      int numTurns(0);
      int afterTurns(n-k);
      while(afterTurns>=m) {
         numTurns++;
         afterTurns-=m;
      }
      if(afterTurns>0) {
         numTurns++;
         afterTurns--;
      } else {
         afterTurns+=m-1;
      }

      if(numTurns==0) {
         std::cout << 0 << std::endl;
         continue;
      }

      // calculate base (last turn):
      std::vector<int> table;
      table.resize(n);
      int turnMin = INT_MAX;
      for(int i=0; i<n-afterTurns; i++) {
         table[i] = std::max(x[i], x[i+afterTurns]);
         turnMin = std::min(turnMin, table[i]);
      }

      for(int turn=1; turn<numTurns; turn++) {// for each turn:
         turnMin = INT_MAX;
         int turnGap(afterTurns+turn*m);
         for(int i=0; i<n-afterTurns-turn*m; i++) {
            // calculate min of middle section:
            int midMin = INT_MAX;
            for(int j=1; j<m; j++) { midMin = std::min(midMin, table[i+j]); }
            table[i] = std::max( x[i] + std::min(midMin, table[i+m]) ,
                                 std::min( table[i] , midMin) + x[i+turnGap]);
            turnMin = std::min(turnMin, table[i]);
         }
      }
      std::cout << turnMin << std::endl;
   }
}
```