**CS/IV/18-22**

#include<iostream>

#include<algorithm>

using namespace std;

class unfairness

{

int N,K,\*list,\*kList;

public:

void getData();

void fair();

int max();

int min();

};

void unfairness::getData()

{

cin>>N>>K;

list=new int[N];

kList=new int[K];

for(int i=0;i<N;i++)

{

cin>>list[i];

}

}

int unfairness::max()

{

int max=0;

for(int i=0;i<K;i++)

{

if(kList[i]>max)

max=kList[i];

}

return max;

}

int unfairness::min()

{

int min=kList[0];

for(int i=0;i<K;i++)

{

if(kList[i]<min)

min=kList[i];

}

return min;

}

void unfairness::fair()

{

sort(list,list+N);

int ans=list[K-1]-list[0];

for (int i=0;i<N-K+1;i++)

if ((list[i+K-1]-list[i])<ans)

ans=list[i+K-1]-list[i];

cout<<ans;

}

int main()

{

unfairness U;

U.getData();

U.fair();

return 0;

}