```
//
// Metoda Hare-Niemeyer'a przydzialu mandatow
// www.algorytm.org
// (c)2006 Tomasz Lubinski
//


#include "stdio.h"

int main()
{
   int mandates[100];
   int votes[100];
   float calcTab[100];
   float max;
   int groupCount, mandatesCount, votesCount, mandatesCounter, i,
      maxInd;

   //get all data
   printf("Podaj liczbe partii (max. 100)\n");
   scanf("%d", &groupCount);

   printf("Podaj liczbe mandatow\n");
   scanf("%d", &mandatesCount);

   for (i=0; i<groupCount; i++)
   {
      printf("Podaj liczbe glosow na partie %d\n", i+1);
      scanf("%d", &votes[i]);
   }

   //calculate number of total votes
   votesCount = 0;
   for (i=0; i<groupCount; i++)
   {
      votesCount += votes[i];
   }

   //initial number of mandates (Hare-Niemeyer)
   mandatesCounter = 0;
   for (i=0; i<groupCount; i++)
   {
      calcTab[i] = (votes[i]*mandatesCount*1.0) / votesCount;
      mandates[i] = (int)(calcTab[i]);
      calcTab[i] -= mandates[i];
      mandatesCounter += mandates[i];
   }
```

```
    //calculate rest of mandates
    while (mandatesCounter < mandatesCount)
    {
       max = -1;
       for (i=0; i<groupCount; i++)
       {
          if (max<calcTab[i])
          {
             max = calcTab[i];
             maxInd = i;
          }
       }

       //give mandate for party with max
       mandates[maxInd]++;
       mandatesCounter++;
       calcTab[maxInd] = 0;
    }

    //present results
    for (i=0; i<groupCount; i++)
    {
       printf("Partia %d ma %d mandatow\n", i+1, mandates[i]);
    }

    return 0;
}
```