**4-7多次最优服务次序问题**

问题描述：设有n 个顾客同时等待一项服务。顾客i需要的服务时间为t i，1≤i≤n。共有s处可以提供此项服务。应如何安排n 个顾客的服务次序才能使平均等待时间达到最小?平均等待时间是n 个顾客等待服务时间的总和除以n。  
算法设计：对于给定的n 个顾客需要的服务时间和s的值，计算最优服务次序。

数据输入:输入数据的第1行有2 个正整数n （n≤10000）和s（s≤1000），表示有n 个顾客且有s 处可以提供顾客需要的服务。接下来的1 行中，有n个正整数，表示n个顾客需要的服务时间。

数据输出：输出数据只有一个整数（计算结果四舍五入），表示计算出的最小平均等待时间。  
代码如下：

贪心策略：最短服务时间优先

#include <iostream>

#include <algorithm>

#include <cstring>

#include <cstdio>

using namespace std;

int main()

{

int i,n,j,k,minx;//n为顾客数量

int s;//工作岗位

double t;//服务时间

int a[10005],b[1005];

while(cin>>n>>s) // 输入顾客数量和可供等待的工位

{

for(i=0; i<n; i++)

cin>>a[i]; //输入顾客使用时间数组

sort(a,a+n); //将数组从小到大排序

memset(b,0,sizeof(b)); // 将b中的数组初始化为0

for(i=0; i<n; i++) //遍历顾客的等待时间

{

minx=0x7fffffff; //定义一个非常大的数

k=0; //

for(j=0; j<s; j++) //有几个工位 进行几次循环 j作为变量存储对应的工位

{

if(minx>b[j]) // 这个和上个小的进行对比

{

minx=b[j]; //记录小的

k=j; //用k来记录小的对应的工位

}

}

b[k]+=a[i]; // b[k] = b[k] + a[i] 记录每个工位的总时间

a[i]=b[k]; // 把每个工位的时间赋给对应顾客 形成 顾客总时间数组

}

t=0;

for(i=0; i<n; i++) // 计算顾客平均等待时间

t+=a[i];

t/=n; //t/n个顾客的时间

printf("%d\n",(int)(t));

}

return 0;

}

例如：

输入：10 2 56 12 1 99 1000 234 33 55 99 812

结果：336