**题目描述**

在一个果园里，多多已经将所有的果子打了下来，而且按果子的不同种类分成了不同的堆。多多决定把所有的果子合成一堆。

每一次合并，多多可以把两堆果子合并到一起，消耗的体力等于两堆果子的重量之和。可以看出，所有的果子经过 n−1n-1n−1 次合并之后， 就只剩下一堆了。多多在合并果子时总共消耗的体力等于每次合并所耗体力之和。

因为还要花大力气把这些果子搬回家，所以多多在合并果子时要尽可能地节省体力。假定每个果子重量都为 111 ，并且已知果子的种类 数和每种果子的数目，你的任务是设计出合并的次序方案，使多多耗费的体力最少，并输出这个最小的体力耗费值。

例如有 333 种果子，数目依次为 111 ， 222 ， 999 。可以先将 111 、 222 堆合并，新堆数目为 333 ，耗费体力为 333 。接着，将新堆与原先的第三堆合并，又得到新的堆，数目为 121212 ，耗费体力为 121212 。所以多多总共耗费体力 =3+12=15=3+12=15=3+12=15 。可以证明 151515 为最小的体力耗费值。

**输入输出格式**

**输入格式：**

共两行。  
第一行是一个整数 n(1≤n≤10000)n(1\leq n\leq 10000)n(1≤n≤10000) ，表示果子的种类数。

第二行包含 nnn 个整数，用空格分隔，第 iii 个整数 ai(1≤ai≤20000)a\_i(1\leq a\_i\leq 20000)ai​(1≤ai​≤20000) 是第 iii 种果子的数目。

**输出格式：**

一个整数，也就是最小的体力耗费值。输入数据保证这个值小于 2312^{31}231 。

**输入输出样例**

**输入样例#1：** 复制

3

1 2 9

**输出样例#1：** 复制

15

#include<bits/stdc++.h>

using namespace std;

int read()

{

int x=0;int f=1;char ch=getchar();

while(ch>'9'||ch<'0'){if(ch=='-'){f=-1;}ch=getchar();}

while(ch>='0'&&ch<='9'){x=x\*10+ch-'0';ch=getchar();}

return x\*f;

}

int main()

{

//freopen("a.txt","r",stdin);

//freopen("b.txt","w",stdout);

int n;

n=read();int x;int tot=0;

priority\_queue<int,vector<int>,greater<int> >q;

for(int i=1;i<=n;i++){x=read();q.push(x);}

while(q.size()>=2)

{

int a=q.top();q.pop();

int b=q.top();q.pop();

tot+=a+b;

q.push(a+b);

}

cout<<tot;

}