1197：山区建小学

题目描述

政府在某山区修建了一条道路，恰好穿越总共m个村庄的每个村庄一次，没有回路或交叉，任意两个村庄只能通过这条路来往。已知任意两个相邻的村庄之间的距离为di(为正整数)，其中，$0<i<m$。为了提高山区的文化素质，政府又决定从m个村中选择n个村建小学(设$0<n\le m<500$)。请根据给定的m、n以及所有相邻村庄的距离，选择在哪些村庄建小学，才使得所有村到最近小学的距离总和最小，计算最小值。

输入

第1行为m和n，其间用空格间隔第2行为m-1个整数，依次表示从一端到另一端的相邻村庄的距离，整数之间以空格间隔。

例如:

10 3
2 4 6 5 2 4 3 1 3

表示在10个村庄建3所学校。第1个村庄与第2个村庄距离为2，第2个村庄与第3个村庄距离为4，第3个村庄与第4个村庄距离为6，...，第9个村庄到第10个村庄的距离为3。

输出

各村庄到最近学校的距离之和的最小值。

输入样例

```
10 2
3 1 3 1 1 1 1 1 3
```

输出样例

```
18
```

解析

一、前缀和

我们先来学习一个知识点：前缀和。

前缀和是指从第1个元素到第i个元素的总和，则题目中的数据就可以表示如下：

10个村庄间距数组 a

索引	1	2	3	4	5	6	7	8	9	10
数值	0	3	1	3	1	1	1	1	1	3

10个村庄间距前缀和 sum

索引	1	2	3	4	5	6	7	8	9	10
数值	0	3	4	7	8	9	10	11	12	15
		3+0	3+1	4+3	7+1	8+1	9+1	10+1	11+1	12+3

对于第二个表格中的sum[10] = 15代表的是1号村庄到10号村庄的总距离为15

如果我们想知道任意两个村庄i，j的间距，就可以直接用sum[j]-sum[i]，例如：
7,8号村庄的间距为： sum[8] - sum[7] = 11 - 10 = 1

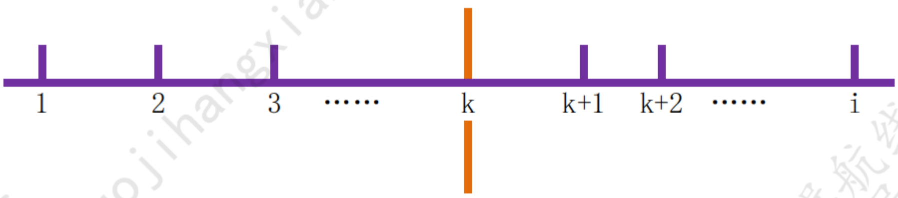
将村庄间距的前缀和计算好，备用。

二、建学校

现在开始分析如何建学校。

我们想在m个村庄中建立n个学校，使其距离和最短。那么一定会存在这样的情况：

在m个村庄中必然存在一个村庄k，将1至m号村庄拆分成两部分，如下图：



那么，我们只要能够保证以下两点，就一定能够造出最短距离和的学校：

1、在1-k号村庄中建造n-1个学校，使其距离和最短
2、在k+1到i号村庄间，建立1个学校，使其距离和最短

现在，我们的问题就拆分成了上面的两个子问题。

先看第一个问题，很明显，第一个问题与我们的原问题一模一样，只是规模变小了。这个问题可以按照原问题的思路继续拆分，即拆分为以下两个小问题：

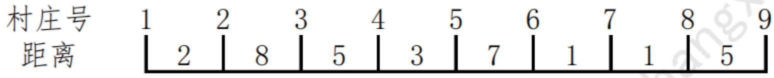
1、在1-k1号村庄中建造n-1-1个学校，使其距离和最短
2、在k1+1到i号村庄间，建立1个学校，使其距离和最短

怎么样，发现了吧，这个问题可以持续的递归和拆分下去，那么这个问题的极限情况是什么呢？答案如下：**在1到i个村庄中建立1所学校，使其距离和最短。i是任意一个村庄编号。**

再来看第二个问题，这个问题和问题1的极限情况十分类似，因此，这两个问题可以合并解决。即：**在任意的x,y号村庄之间选择1个点建立学校，使其距离和最短。**

在这里直接给出这个点的计算公式：**mid = x + (y-x)/2**

即只要你选择的永远是x,y的中点，就一定能够保证距离和最短。我们以下面的图像做个测试：



我们现在来求3-7号之前距离和最短的点，根据公式可以得出：v = 3+(7-3)/2 = 5。

5号点就是我们的最短点，下面我们来验证一下：
3号到5号的距离：8
4号到5号的距离：3
6号到5号的距离：7
7号到5号的距离：8
因此，总距离和为26。同学们可以尝试一下其他的点，都不会小于这个数值的。

3-7号总共是5个点，是奇数个点，对于偶数个点，这个公式也是适用的，同学们可以自行尝试一下。

三、关键变量

1、sum[i]表示从第1个村庄到第i个村庄的距离，即

```
for (int i = 2; i <=m ; ++i) {
    cin>>sum[i];
    sum[i] = sum[i-1] + sum[i];
}
```

2、mid表示任意x到y号村庄之间距离和最短的点，即：

```
mid = k + (n-k) / 2
```

3、minDis[i][j]表示从第i号村庄到第j号村庄间建立1所学校的最短距离，即

```
for (int k = i; k <=j ; ++k) {
    minDis[i][j] += abs(sum[k]-sum[mid]);
}
```

4、dp[m][n]表示前m个村庄建立n所学校的距离和最短，即：

```
dp[m][n] = min(dp[m][n],dp[k][n-1]+minDis[k+1][m])
```

最后一个问题：dp数组的初始值应该是多少呢？

答案很简单，因为我们首先会尝试在不同的村庄中建立1所学校，而这一所学校的选址必然是距离所有村庄中最短的距离，即minDis数组的数值，因此则有如下式子：

```
dp[1][1] = minDis[1][1]
dp[2][1] = minDis[1][2]
dp[m][1] = minDis[1][m]
```

编码

```
#include<bits/stdc++.h>

using namespace std;

int m, n; //定义村庄数和学校数

int sum[501]; //存储从第1个村庄到第m个村庄的距离

int dp[501][501];  //在前i个村庄建立j个学校的最短距离
int minDis[505][505];  //在任意两个村庄间建立一所学校所需要的最小值

int CountMinDis(int l, int r) {
    int mid = l + (r - l) / 2;
    int v = 0;
    for (int i = l; i <= r; ++i) {
        v += abs(sum[i] - sum[mid]);
    }
    return v;
}

int main() {
```

```cpp
    memset(dp, 0x3f, sizeof(dp));
    cin >> m >> n;
    int dis;
    //开始记录村庄间的间距，编号与题目保持一致，便于理解
    for (int i = 2; i <= m; ++i) {
        cin >> dis;
        //计算前缀和
        sum[i] = sum[i - 1] + dis;
    }

    //计算在i到j号村庄中建立1所学校最短的距离
    for (int i = 1; i <= m; ++i) {
        for (int j = i; j <= m; ++j) {
            minDis[i][j] = CountMinDis(i, j);
        }
    }

    //初始化dp数组，即从1号到m号村庄间建立1所学校的最短距离
    for (int i = 1; i <= m; ++i) {
        dp[i][1] = minDis[1][i];
    }

    //遍历所有村庄
    for (int i = 1; i <= m; ++i) {
        //遍历所有的学校数量，建立的学校数量不能超过村庄数
        for (int j = 1; j <= n && j <= i; ++j) {
            //尝试不同的分割点
            for (int k = 1; k < i; ++k) {
                dp[i][j] = min(dp[i][j], dp[k][j - 1] + minDis[k + 1][i]);
            }
        }
    }

    //输出最后的结果
    cout << dp[m][n];

    return 0;
}
```