


## 实验四

 编程题 2 (总分: 100.00)

#	题目	分值
1	<b>实验4-1: MAX SUM</b>	50.00

**Description**

给定由 $n$ 整数(可能为负数)组成的序列  $\{a_1, a_2, \dots, a_n\}$ , 求该序列形如  $a_i + a_{i+1} + \dots + a_j$  的子段和的最大值。当所有的整数均为负数时定义其最大子段和为0。

**Input**

输入包含多组测试数据。第一行为一个整数 $C$ , 表示有 $C$ 组测试数据, 接下来有 $2 * C$ 行数据, 每组测试数据占2行, 每组测试数据第一行是1个整数 $n$ , 表示有 $n$ 个整数, 接下来一行有 $n$ 个整数, 它们之间用空格隔开。

**Output**

你的输出应该有 $C$ 行, 即每组测试数据的输出占一行, 它是计算出的最大子段和。

**Sample Input**

1

6

-2 11 -4 13 -5 -2

**Sample Output**

20

参考答案:

```
#include <stdio.h>
int maxsum(int n,int *a)
{
    int sum=0,b=0,i,j;
    for(i=1;i<=n;i++)
    {
        if(b>0) b+=a[i];
        else b=a[i];
        if(b>sum) sum=b;
    }
    return sum;
}
main()
{
    int m;
    scanf("%d",&m);
    while(m--)
    {
        int a[100],i,max,n;
        scanf("%d",&n);
        for(i=1;i<=n;i++)
            scanf("%d",&a[i]);
        max=maxsum(n,a);
        printf("%d\n",max);
    }
}
```

## 2 实验4-2: 矩阵连乘问题

50.00

### Description

给定 $n$ 个矩阵  $\{A_1, A_2, \dots, A_n\}$ ，其中 $A_i$ 与 $A_{i+1}$ 是可乘的， $i=1, 2, \dots, n-1$ 。如何确定计算矩阵连乘积的计算次序，使得依此次序计算矩阵连乘积需要的数乘次数最少。

### Input

输入包含多组测试数据。第一行为一个整数 $C$ ，表示有 $C$ 组测试数据，接下来有 $2 * C$ 行数据，每组测试数据占2行，每组测试数据第一行是1个整数 $n$ ，表示有 $n$ 个矩阵连乘，接下来一行有 $n+1$ 个数，表示是 $n$ 个矩阵的行及第 $n$ 个矩阵的列，它们之间用空格隔开。

### Output

你的输出应该有 $C$ 行，即每组测试数据的输出占一行，它是计算出的矩阵最少连乘积次数。

### Sample Input

1

3

10 100 5 50

Sample Output

7500

参考答案:

```
#include <stdio.h>
#include <limits.h>

int min(int a, int b) {
    return a < b ? a : b;
}

int matrix_chain_order(int p[], int n) {
    int m[n][n];

    for (int i = 1; i < n; i++) {
        m[i][i] = 0;
    }

    for (int l = 2; l < n; l++) {
        for (int i = 1; i < n - l + 1; i++) {
            int j = i + l - 1;
            m[i][j] = INT_MAX;
            for (int k = i; k <= j - 1; k++) {
                int q = m[i][k] + m[k+1][j] + p[i-1]*p[k]*
p[j];
                m[i][j] = min(m[i][j], q);
            }
        }
    }

    return m[1][n-1];
}

int main() {
    int test_cases;
    scanf("%d", &test_cases);

    while (test_cases--) {
        int n;
        scanf("%d", &n);
        int dimensions[n+1];
        for (int i = 0; i < n+1; i++) {
            scanf("%d", &dimensions[i]);
        }
        int result = matrix_chain_order(dimensions, n+1);
```

```
        printf("%d\n", result);  
    }  
    return 0;  
}
```