

每题时间限制均为 1s，空间限制均为 512MB，开启 O2 优化开关（请不要自己手动打开任何优化开关，会直接 CE），使用 C++11 编译器。

评测环境：

处理器:	Intel(R) Xeon(R) E-2176M CPU @ 2.70GHz 2.71 GHz
已安装的内存(RAM):	16.0 GB (15.7 GB 可用)
系统类型:	64 位操作系统，基于 x64 的处理器

简单题 (solo)

Solo 生成了一个一条长度为 N 的链，链上的点标号 1 到 N ，其中 i 号点和 $i+1$ 号点之间的边的权值为 A_i ，现在 Solo 决定从中删除 K 条两两之间没有公共点的子链（长度可以为 0，即一个单独的节点），使得这 K 条链上涵盖的边的边权和最大。

输入格式(solo.in)

第一行两个数 N, K

第二行 $N-1$ 个数，第 i 个数代表 A_i

输出格式(solo.out)

第一行一个数代表最大值

接下来 K 行每行两个数，表示最大值对应的方案中选出的 K 条链的起点和终点

输入样例	输出样例
9 4 -10 2 1 3 6 -2 17 1	29 1 1 2 6 7 8 9 9

样例说明：方案可能不唯一，输出任意一种都可以

数据规模：

对于 100% 的数据， $|A_i| \leq 100000$ ， $k \leq n \leq 100000$

对于 20% 的数据， $k \leq n \leq 5000$ 。

对于每个点假如最大值正确但是方案不正确，可以获得这个点 50% 的分数。

大水题(archies)

Atopos 有一个长度为 N 的数组，数组里的数都是大于 0 的整数，现在 Atopos 想要知道一些事情。

Atopos 共有四种询问，每一种询问她会先指定一段区间 $[l,r]$ ，然后给定一个幸运数 x ，询问区间内的数是否能找到两个使得他们的运算后为 x ，四种询问对应的操作分别为加，减，乘，除（不是整除）。

输入格式(archies.in)

第一行两个数 N,M

第二行 N 个数，第 i 个数代表 A_i

之后 M 行，每行四个数 $type,l,r,x$ ，其中 $type=1$ 代表加法， $type=2$ 代表减法， $type=3$ 代表乘法， $type=4$ 代表除法。

输出格式(archies.out)

共 M 行，每行一个数，代表每个询问的结果，假如存在这样的一对数则输出 1，否则输出 0。

输入样例	输出样例
7 4	0
1 2 3 5 7 8 9	1
1 1 3 6	1
2 1 7 8	0
3 2 5 15	
4 4 7 3	

数据规模：

对于 100% 的数据， $1 \leq A_i \leq 100000$ ， $1 \leq x \leq 100000$ ， $N,M \leq 100000$

对于 30% 的数据，只有操作 1 和操作 2

对于 20% 的数据，只有操作 3

送分题(atopos)

Archies 最近学习了上下文无关语法(CFG), CFG 由若干生成式, 生成符号和终结符号组成, 在这题中, 0, 1 和空 (长度为 0 的字符串) 是唯一的终结符。

每一个生成式由如下形式给出: $X \rightarrow (A-Z, 0, 1)^*$

左侧 X 对应了一个生成符号, 在这题中生成符号只有 A-Z 这 26 种。右侧代表任意一个由 A-Z, 0, 1 这 28 个字母组成的字符串 (长度可以为 0)。

S 是开始符号, 即一开始我们的纸上只有一个符号 S, 之后可以进行若干次生成式的替换, 当某个时刻纸上只有 0 和 1 (也可能什么都没有), 我们就结束操作。

现在 Archies 想要知道所有操作结束后纸上可能出现的字符串中, 长度恰好为 k 的前缀有多少不同的可能。假如生成的串长度不超过 k, 我们认为这也是一个长度不超过 k 的合法的前缀。(后面的位置视为用空补上)

输入格式(atopos.in)

第一行两个数 n, k, n 代表生成式的个数

之后 n 行每行一个生成式

输出格式(atopos.out)

第一行一个数 n, 代表能够达到的长度为 k 的合法串的个数。

之后 n 行, 每行一个串代表一个能达到的前缀, 用 eps 代表空串。

这 n 个串可以按照任意顺序输出。

输入样例	输出样例
2 3 S->0S1S S->	5 eps 01 000 001 010

数据规模：

对于 100%的数据， $n \leq 200, k \leq 7$

对于 20%的数据， $k \leq 4$

对于另外 20%的数据，生成符号只有 A-G