

Southeastern European Regional Programming Contest
Bucharest, Romania - Vinnitsa, Ukraine
October 13, 2012

Problem G
Miraculous Drug

Input File: G.IN
Output File: standard output
Program Source File: G.C, G.CPP, G.JAVA

Joe is an enthusiast biomedical researcher. He is very close to discover the cure for a terrible disease. In order to prepare the miraculous drug he needs to buy a special enzyme, that is quite expensive and unfortunately loses its properties after a fixed period of time. Now Joe is in the clinical trial phase. He needs a drug available at each hour. Thus he has to prepare exactly the same quantity of drug every hour. The price of the enzyme might vary from hour to hour. The cost of the enzyme on hour `i` is c_i. The life time of the enzyme is `h` hours. Given the prices for the next `n` hours, Joe has to find out the optimal cost to purchase enzymes such that the drug is available in each of the `n` hours. If the price is the same Joe prefers to buy fresh enzymes, not to stock them. We assume an unlimited quantity of enzymes is available each hour. Can you help Joe?

The program input is from a text file. The file contains several data sets. A data set starts with the number `n` ($n<10000$) of hours. Follows `h` ($h<10000$) the number of hours of the enzyme life time, `b` the starting point, `e` the ending point of the printing interval ($1 \le b, e \le n$), and the enzyme costs c_i ($c_i<10000$), `i=1..n`. The program prints for each hour in the interval `[b,e]` the number of enzymes Joe decides to buy.

White spaces can occur freely in the input. The input data are correct and terminate with an end of file. For each set of data the program prints the result to the standard output from the beginning of a line. An input/output sample is in the table below. There are two data sets. For the first one `n=6`, `h=3`, `b=1`, `e=6`, and the costs are `5 4 4 3 5 6`. The result consists of the numbers of enzymes bought every hour, printed from the beginning of the line, separated with tabs.

Sample input	Sample output
6 3 1 6 5 4 4 3 5 6 3 3 2 3 9000 9000 9000	1 1 1 3 0 0 1 1