Digital Pakistan Speed Programing Competition Online Qualifier Round

Instructions

- Do not open the booklet unless you are explicitly told to do so. You can only read these instructions below.
- If you have any question regarding the problems, seek a clarification from the judges using DOMJudge.
- Before submitting a run, make sure that it is executable via command line. For Java, it must be executable via "javac" and for GNU C++ via "g++". Java programmers need to remove any "package" statements and source code's file name must be the same as of main class. C++ programmers need to remove any getch() / system("pause") like statements.
- Do not attach input files while submitting a run, only submit/attach source code files, i.e., *.java or *.cpp or *.py.
- Language supported: C/C++, Java and Python3
- Source code file name should not contain white space or special characters.
- You must take input from Console i.e.: Standard Input Stream (stdin in C, cin in C++, System.in in Java, stdin in Python)
- You must print your output to Console i.e.: Standard Output Stream (stdout in C, cout in C++, System.out in Java)
- Please, don't create/open any file for input or output.
- Please strictly meet the output format requirements as described in problem statements, because your program will be auto judged by computer. Your output will be compared with judge's output byte-by-byte and not tolerate even a difference of single byte. So, be aware! **Pay special attention to spaces, commas, dots, newlines, decimal places, case sensitivity etc.**
- All your programs must meet the time constraint specified.
- The decision of judges will be absolutely final.

Problem 07: Project Optimization

Time limit: 1 seconds

EP (Efficient Partners) Inc. is a leading consultancy firm that provides optimization solutions to software houses to increase efficiency and reduce completion time. EP looks at a software house to establish the kind of projects it implements and suggests how the employees can be divided into small, specialized teams, such as **Requirement Engineering team**, **Conceptual Design** … up to the **Testing and Deployment** teams. In this way, each project can be streamlined with specialized teams working on different aspects of the project. To ensure optimization and offer redundancy, EP advises all software houses to have two sets of each team – the **main team (*m*)** and an **alternate team (*a*)**, so that if one team is too busy, falls sick, or lacks expertise, then that phase of the project can be handled by the other team if they can do it quicker. Each project must pass through all ***n*** phases of a project in a sequential order to reach closure.

As part of its software, EP also provides optimization solutions so that software houses can use the **parallel teams** to gain maximum throughput. In doing so, it takes into account the time (in days) each phase of the project will take by the main/alternate teams and optimize assignments. Assuming each project has to pass through all ***n*** phases, let m_i and a_i denote the time taken on the ***i*-th** ($1 \leq i \leq n$) phase of the project by the main and alternate teams, respectively. At the start of the project, there is a cost associated in understanding the project denoted by m_0 and a_0 (for the main team and the alternate team, respectively). Similarly, if a project is passed from the **(*i*-1)-th** main team to the ***i*-th** alternate team, the cost is given by $t_{m,i}$ ($t_{a,i}$ if it is passed from the alternate team to the main team). There is no cost if the project is passed from one main team to the next main team (or alternate team to the next alternate team). The deployment phase takes d_m and d_a time by the main and alternate teams, respectively.

EP has a software to compute all these mentioned values but wants you to write an optimization algorithm that uses these values as input and give (output) the minimum time it will take to run the project by selecting different (main or alternate) teams for different phases of the project, so that the project can be successfully completed and deployed in the shortest time.

Input Format

The first line in the input file is the number of phases, ***n***, in the project. The next two lines contain the values of m_i and a_i, respectively, for ***i*=1** to ***n***. The subsequent two lines contain the values $t_{m,i}$ and $t_{a,i}$, respectively, for ***i*=1** to ***n***. The fifth and final line has four values, namely m_0, a_0, d_m, and d_a. The ranges are ($1 \leq n \leq 10^4$), while other values range between **1** and **32,768**.

Output Format

The output is a single value that gives the minimum time needed to execute the project if all phases are optimized. In the sample output, phases 1, 2, 3, and 4 are implemented by the main team, while phase 5 is implemented by the alternate team, and the minimum execution time is 31 days.

Sample Input	Sample Output
5 2 5 3 2 3 2 8 3 5 3 0 8 2 6 3 0 6 3 7 3 8 10 20 5	31