/\*

NUMERO DE GRUPO:

NOMBRRE Y APELLIDOS DE LOS AUTORES:

\*/

#include <iostream>

using namespace std;

// Representación de un tema

typedef struct {

float tiempo\_req; // tiempo requerido para estudiar

float nota\_prim; // Nota que supone en la primera parte

float nota\_sec; // Nota que supone en la segunda parte

} tInfoTema;

/\*

(1) En caso de utilizar una generalización, indicar su parámetros, y explicar para que

sirven cada uno de ellos

(2) ¿Cómo son las soluciones parciales de este problema?

(3) ¿Cuándo una solución parcial es viable?

(4) ¿Cuándo una solución parcial es una solución final al problema?

(5) Dada una solución parcial, ¿cómo se generan las siguientes soluciones

parciales viables?

(6) Análisis de casos

(6.1) Caso(s) base

(6.2) Caso(s) recursivos

(7) En caso de utilizar una generalización, explicar cómo se define el algoritmo final

a partir de la misma, por inmersión.

\*/

float mejor\_puntuacion(const tInfoTema info\_temas[], unsigned int num\_temas, float tiempo\_disponible) {

// Punto de entrada al algoritmo: a implementar

}

const unsigned int MAX\_TEMAS = 20;

bool procesa\_caso() {

int num\_temas;

tInfoTema info\_temas[MAX\_TEMAS];

cin >> num\_temas;

if (num\_temas != -1) {

float tiempo\_disponible;

cin >> tiempo\_disponible;

for (int i = 0; i < num\_temas; i++) {

cin >> info\_temas[i].tiempo\_req;

cin >> info\_temas[i].nota\_prim;

cin >> info\_temas[i].nota\_sec;

}

cout << mejor\_puntuacion(info\_temas, num\_temas, tiempo\_disponible) << endl;

return true;

}

else {

return false;

}

}

int main() {

while (procesa\_caso());

}