**Анализ на решението на задача  
E1. Венко, Ценко и Денка**

Логиката за решаване на задачата е доста очевидна: към времето, изминало от началото на петъка до момента на започване писането на домашно, добавяме времето за писане на задачите от домашното и после добавяме времето за почивка, за да получим времето на присъединяване на приятел към видеоплатформата. Изчисленото време за присъединяване към платформата на поредния от приятелите сравняваме с времето на последно присъединилия се и ако е по-голямо, коригираме времето на последния присъединил се да съвпада с времето на присъединяване на поредния от приятелите.

Известно неудобство при реализиране на описаната логика е, че описаните времена са представени в тази задача с данни, имащи няколко съставни части (час, минута и секунда). За облекчаване на пресмятанията преобразуваме многосъставните данни в данни, имащи само една съставляваща – в нашия случай всички данни за време ги преобразуваме да са изразени в секунди. Накрая извършваме обратнния процес – преобразуваме времето на започване на играта от израено в секунди отново в многосъставна форма на изразяване – в часове, минути и секунди. Не забравяме, че часовата съставляваща на крайния резултат може да е число по-голямо от часовете в едно денонощие. В такъв случай преобразуваме часовата съставляваща до остатъка ѝ при делене с 24.

Особености при реализацията

След като се изчисли времето на поредния приятел за присъединяване към платформата и преценката дали това е последният присъединил се, всички променливи, свързани с този човек, вече стават ненужни. Това позволява да ги използваме отново при сметките на времето за присъединяване на следващия от приятелите. Всичко това води до писане на код само за един от приятелите и двукратното му копиране без каквито и да са допълнителни редакции за останалите двама от приятелите (вижте, че редове 13÷15 са еднакви със 17÷19 и 21÷23 по-долу).

Отбележете, че този подход за писане на кода, предполага обработките на данните и за тримата да бъдат абсолютно еднакви. Значи и на първия трябва да сравняваме времето с последния присъединил се. Да, ама преди първия присъединил се няма друг, чието време да използваме като рекордно късен момент. Затова сме задали рекорда за най-късен момент T първоначално да е в най-ранния възможен момент (полунощ срещу петъка) – вижте инициализирането на променливата T на ред 10 в кода по-долу.

Сорс-кодът на програмата

Отбележете, че в приложения по-долу сорс-код пред всеки ред е изписан номерът му. Ако копирате кода в някакъв редактор на код, трябва да изтриете номерата в началото на всеки ред!

1 #include <iostream>

2 **using namespace std**;

3

4 **int** main() {

5 **int** h,m,s, **/// час, минута и секунда, когато човек започва решаването**

6 c, **/// колко задачи решава**

7 mm,ss, **/// време за решаване (минути и секунди) на една задача**

8 t, **/\*\* време на присъединяване на човек в платформата**

9 **(в секунди след началото на петъка) \*/**

10 T=0; **/\*\* време на последния присъединил се (първоначално няма присъединили**11  **се-времето на несъществуващия "последен" е в началото на петъка) \*/**

12 **/// Данните за първия**

13 **cin** >> h >> m >> s >> c >> mm >> ss;

14 t = h\*3600+m\*60+s + c\*(mm\*60+ss) + 600; **/// време за присъединяване**

15 **if** (t>T) T=t;

16 **/// Данните за втория**

17 **cin** >> h >> m >> s >> c >> mm >> ss;

18 t = h\*3600+m\*60+s + c\*(mm\*60+ss) + 600; **/// време за присъединяване**

19 **if** (t>T) T=t;

20 **/// Данните за третия**

21 **cin** >> h >> m >> s >> c >> mm >> ss;

22 t = h\*3600+m\*60+s + c\*(mm\*60+ss) + 600; **/// време за присъединяване**

23 **if** (t>T) T=t;

24

25 **cout** << (T/3600)%24 << ' ' << (T%3600)/60 << ' ' << T%60 << **endl**;

26

27 **return** 0;

28 }

*Автор: Евгений Василев*