

**НАЦИОНАЛНО ОНЛАЙН СЪСТЕЗАНИЕ ПО ИНФОРМАТИКА**  
**„Д-р Младен Манев“**  
**2 септември 2021 г.**

**Задача В4. ХЕШИРАНЕ**

След последния урок, който г-н Петров преподаде на Ясмин, тя вече е експерт по хеширане. Затова той реши да поиграе с нея на една игра. Първоначално г-н Петров си намисля две числа **B** и **M** – съответно база и модул на следната хеш функция:

$$F(S) = (S_0 \cdot B^0 + S_1 \cdot B^1 + \dots + S_{N-1} \cdot B^{N-1}) \bmod M$$

\*  $F(S)$  приема като аргументи низове, съставени само от малки латински букви.  $S_i$  е номерът на  $i$ -тата буква от низа  $S$  в английската азбука. Номерацията на символите в низа започва от 0, а на буквите в азбуката от 1.

След това наред е Ясмин, чиято задача е да отгатне базата и модула на хеш функцията. За тази цел тя може да задава въпроси, състоящи се от по един непразен символен низ  $S$  с дължина не по-голяма от **100** символа. При всеки такъв въпрос г-н Петров ще въведе низа в специално подготвена за целта програма и ще каже на Ясмин стойността на хеш функцията за конкретния низ. Единственото ограничение е, че броят на въпросите не може да бъде повече от **10**, понеже г-н Петров никак не е склонен да отдели цял ден да смята хешове дори когато има готова програма за това. Помогнете на момичето да отгатне параметрите на намислената хеш функция в няколко поредни игри със своя учител.

**Задача:**

Напишете функция `play()`, която ще се компилира с програмата на журито и ще комуникира с нея, задавайки въпроси от гореописания вид. По време на изпълнението си тя трябва да открие базата и модула на няколко хеш функции.

**Детайли по имплементацията:**

Функцията `void play(int t)`, която трябва да напишете, ще бъде извикана само веднъж от програмата на журито и като аргумент ще получи цялото число **T** – броя тестови случаи, които трябва да обработи в рамките на изпълнението си.

За комуникация с програмата на журито Ви се предоставят следните две функции:

```
int calculate_hash(const string &s);  
void submit_hash(int base, int mod);
```

При всяко извикване на функцията `calculate_hash`, тя ще връща едно число, равно на стойността на текущата хеш функция за предадения като параметър низ. Низовете, които предавате като аргументи на функцията, трябва да отговарят на следните условия: да се състоят от положителен брой символи, но не повече от 100; да съдържат само малки латински букви. Ако в произволен момент нарушите някое от описаните условия или превишите броя допустими заявки, изпълнението на вашата функция ще бъде прекратено и няма да получите точки за съответния тест.

# НАЦИОНАЛНО ОНЛАЙН СЪСТЕЗАНИЕ ПО ИНФОРМАТИКА

„Д-р Младен Манев“

2 септември 2021 г.

След като откриете базата и модула на поредния хеш, вашата функция ще трябва да извика функцията `submit_hash` и да предаде като аргумент двете числа. След това, вашата функция трябва да продължи с намирането на параметрите на следващата хеш функция, ако има такава. След  $T$ -тото извикване на `submit_hash` изпълнението на вашата функция ще бъде прекратено.

Към системата изпратете файл **hashing.cpp**. В него, освен функцията `play`, може да има всякакви помощни функции, структури, променливи и т. н. Той не трябва да съдържа функция `main()` и задължително трябва да включва хедър файла `hashing.h` чрез указание към предпроцесора `#include "hashing.h"` в началото си.

## Пример:

Функция на участника	Програма на журито
	<code>play(1)</code>
<code>calculate_hash("helloworld")</code>	247607872
<code>calculate_hash("gabrovo")</code>	2176790
<code>calculate_hash("qwertyuiopasdf")</code>	1219313707
<code>calculate_hash("abbccdddeeeeee")</code>	698639851
<code>calculate_hash("goodluck")</code>	9795898
<code>calculate_hash("hashingnihsah")</code>	1178030339
<code>submit_hash(7, 1234567890)</code>	

## Ограничения:

$$1 \leq T \leq 20$$

$$1 < B \leq 100$$

$$1 \cdot 10^9 \leq M \leq 2 \cdot 10^9$$

## Оценяване:

Оценяването се извършва посредством 20 системни теста,  $i$ -тият от които се състои от  $i$  на брой тестови случая. Ако вашата функция открие правилно параметрите на всички хеш функции, ще получите 5 точки за съответния тест. В противен случай ще получите  $5 * \frac{x-1}{T}$  точки, където  $x$  е номерът на първия сгрешен тестов случай.

## Локално тестване:

Предоставен Ви е файлът `Lgrader.cpp`, който може да компилирате заедно с вашата функция, за да я тествате. При стартиране трябва да въведете числото  $T$ , следвано от  $T$  двойки числа – базата и модула за поредния тестов случай. Програмата ще отпечата комуникацията, която се извършва. Може да модифицирате този файл както искате.