Назив проблема: Крипто

|  |  |
| --- | --- |
| Аутор: Петар Величковић | Анализа: Петар Величковић |

|  |  |
| --- | --- |
| Тагови: | Велики бројеви, модуларна аритметика, ад-хок |

## Решење и анализа:

Проблем Крипто спада у категорију лакших задатака на другим Квалификацијама. Због једноставности формулације у тексту задатка, није тешко превести проблем у формално математички облик; једна од могућих форми је:

*За дата два цела броја и , одредити цели број тако да важи и .*

Непажљивом читаоцу се ово може чинити као тривијалан задатак, и без обазирања на ограничења променљивих (или можда без разумевања тих ограничења) добијамо следећи C++ код:

int N, M;

scanf("%d%d", &N, &M);

printf("%d\n", N % M);

Овакво решење, међутим, доноси само око 20 бодова; целобројни тип int не подржава бројеве веће од . Пажљивији решаваоци су приметили да је боље користити 64-битни тип long long, који је уједно и највећи целобројни тип који нам програмски језик нуди. Решење се тако трансформише у следећи облик:

long long N, M;

scanf("%lld%lld", &N, &M);

printf("%lld\n", N % M);

Овакво решење доноси 40 бодова; док је овај тип свакако већег капацитета, у задатку можемо да очекујемо бројеве које имају и до цифара, док long long може да подржи целе бројеве само до . Једно могуће решење за овај проблем је креирање сопствене класе за велике бројеве, и затим имплементирање фукнција дељења и множења над тим типовима да би се дошло до операције модуловања; али пажљивим опсервацијама можемо доста поједноставити овај приступ. Најпре приметимо да, пошто је модул увек унутар 64-битне променљиве, да ће и решење стати у 64-битни цео број. Затим би требало број трансформисати у неки облик који ће нам омогућити да решење рачунамо у корацима, а да нам оно никад не искочи из тих граница. Конкретно, уколико је , где је број цифара броја , а његова -та значајна цифра, посматрајмо следећи полином, у две различите форме:

Уколико је , онда је ово очигледно једнако броју . Друга форма нам даје поступан начин за рачунање овог модула; можемо, почевши од нуле, у сваком моменту да множимо број са и да му додамо следећу цифру броја (идући слева надесно), одржавајући све време резултат по модулу . Ово можемо урадити уз помоћ идентитета везаних за модул збира и производа два цела броја:

Коначно решење се онда може изразити у виду рекурентне везе:

где је тражено решење. Ово рачунање се лако обавља следећим кратким C++ кодом:

unsigned long long kripto(string N, unsigned long long M)

{

unsigned long long ret = 0LLU;

for (int i=0;i<N.length();i++)

{

ret \*= (10 % M);

ret %= M;

ret += ((N[i] - '0') % M);

ret %= M;

}

return ret;

}

Временска сложеност овог решења је и оно је довољно за освајање 100 бодова.

**Напомена**: приметите да је у решењу коришћена променљива типа unsigned long long; ово је урађено зато што је могуће искочити из опсега long long променљиве приликом множења са , међутим тест примери нису санкционисали ову грешку; могуће је освојити 100 поена и без коришћења овог типа.