ТАКМИЧЕЊА ИЗ ИНФОРМАТИКЕ ЗА УЧЕНИКЕ СРЕДЊИХ ШКОЛА

Никола Милосављевић, Драган Урошевић, Душан Здравковић,

Срђан Божовић

Апстракт: Предавање је организовано кроз две целине. У првој, већој целини председник и чланови комисије ће говорити о систему, програму и организацији такмичења за средње школе. У склопу представљања система такмичења, биће речи о његовом историјату и његовом тренутном облику. Наставницима ће бити представљени и детаљи новог правилника. Посебна пажња биће посвећена програму такмичења и областима које су заступљене на разним нивоима такмичења. Биће објашњен начин формулисања задатака и њиховог бодовања. На семинару ће бити речи о томе како решавати конкретне проблеме (биће детаљно анализирани задаци са окружног и државног такмичења 2015/2016. године). У другој, краћој целини, биће представљен систем BubbleBee, који садржи велику, стално растућу колекцију наставних материјала за разне нивое учења програмирања.

Историјат

Такмичења из инфорамтике за ученике срењих школа се организују од школске 1990/1991. године. Мотив за покретање ових такмичења је био покретање Међународне олимпијаде за ученике срењих школа (односно до старосне доби не више од 20 година на крају календарске године). Прва Међународна олимпијада је одржана у Пловдиву 1989. Године. Како су се у то време већ почеле дешавати драстичне промене у радашњој Југославији, на савезном такмичењу су учествовали такмичари из Србије, Црне Горе, Македоније и Босне и Херцеговине. Од школске 1991/92. године до школске 2005/2006. године се организује у 4 круга: окружно, реубличко, савезно (траје два дана) и мала олимпијада (избор екипе за међународна такмичења). Школске 2006/2007. је дошло до промене у организацији па постоје три круга: окружно, државно и Српска Информатичка Олимпијада (СИО), која траје два дана и на којој се бира екипа за међународна такмичења. Од школске 2010/2011. је уведен додатни круг квалификације са незнатно другачијим правилима: траје више дана (од 7 до 10), ученици предају решења на сервер на коме се врши тестирање и оцењивање. Сврха овог (или ових) кругова је адаптација на такмичарске услове.

Од 1993. године постоји и Балканска олимпијада из Информатике (Балканијада). У организацији овог такмичења постоје одређени проблеми око прихватања организације од стране појединих земаља, па је у неколико наврата уместо званичне Балканијаде одржано такмичење за ученике земаља учесница у време одржавања зв. Јесењег турнира у Шумену (Бугарска). Наша земља је организовала Балканијаду два пута: 2002. и 2012. године.

Тренутни облик такмичења

На прва два круга (окружно и државно), ученици су подељени у две категорије:

- А категорија у којој се такмиче ученици другог, трећег и четвртог разреда Математичке гимназије и одељења која раде по програму Математичке гимназије, ученици другог, трећег и четвртог разреда Рачунарске гимназије и осталих специјалisованих одељења.
- Б категорија у којој се такмиче ученици првог разреда Математичке гимназије и одељења која раде по програму Математичке гимназије, ученици првог разреда Рачунарске гимназије и ученици осталих средњих школа

Ученици који по горњем припадају Б категорији могу у било ком тренутку у току годишњег циклуса прећи у А категорију, али онда остају у А категорији до краја годишњег циклуса.

На Српској информатичкој олимпијади постоји само једна категорија.

Задаци који се решавају на такмичењима су алгоритамске природе. За већину задатака је потребно осмислити алгоритам за задати проблем. Алгоритам треба искодирати у неком од званичних програмских језика (тренутно су то Pascal, C C++). У сваком проблему (задатку) постоји временско ограничење (за време извршавања програм) и меморијско ограничење (за укупну количину меморије искоришћене за чување података).

Такмичарска решења се оцењују тако што се програм преводи и извршава на серверу чије су карактеристике познате у време такмичења. Ако програм не може да се преведе (тј. има синтаксних грешака), неће бити оцењивано решење. *Нису дозвољене измене програма након предавања решења* (тј. завршетка израде). За сваки задатак Комисија саставља одређен број тест примера и на основу упешности такмичарског решења (програма) на тим тест примерима одређује се укупан број освојених поена.

Такмичарска решења са на свим нивоима такмичења оцењује Комисија. Локални одбори који врше организацију Окружног такмичења прикупљају решења непосредно после такмичења и предају на такмичарски сервер по процедури која се шаље заједно са поставком проблема. Локални одбори такође обезбеђују услове за такмичење: рачунаре на којима су инсталирана радна окружења (за дозвољене програмске језике) компатибилна са оним која се користе (од стране Комисије) за тестирање и оцењивање такмичарских решења.

По објављивању прелиминарних резултата, тест примери који су коришћени за оцењивање такмичарских решења се објављују на сајту такмичења, како би ученици могли сами да тестирају своја решења. Ученици могу уложити жалбу (приговор). Жалбе не могу бити такве да захтевају измене програма (нпр. брисање непотребних исписа и томе слично).

У циљу унапређења квалитета такмичења последњих три године се врши регистрација такамичара. Само ученици који су (успешно) регистровани као такмичари могу наступити на квалификацијама и само они који су скупили на квалификацијама минимални прописани број поена могу наступити на Окружном такмичењу.

За ову годину су планиране још неке измене у администрацији такмичења, о којима ће бити речи у току самог семинара.

О такмичарским задацима

Годишњи циклус је конципиран тако да се тежина задатака повећава са сваким следећим кругом. Тежина је одређена сложеношћу идеје, количином знања и програмерским умећем који су потребни за решавање задатка (писање програма).

Поред познавања неког од програмских језика, очекује се и знање из одређених области. У наставку су дате листе тих области за поједине кругове (нивое) такмичења. Напомињемо да листе нису строго конципиране, већ само оријентационе, тако да може бити мањих одступања.

Окружно такмичење:

- Једноставне структуре (слогови, низови, стрингови, матрице, листе, скупови)
- Једноставни математички поступци (сумирања, прости бројеви, Еуклидов алгоритам, бројни системи)
- Рекурзија, бектрек, комбинаторика, пребрајања
- Операције са великим (целим) бројевима
- Елементарни (једноставни) алгоритми за сортирање (сортирање селекцијом, сортирање уметањем, bubble сортирање, сортирање пребрајањем)
- Представљање основних геометријских објеката (тачка, дуж, права, кружница) и једноставни поступци над њима (одређивање пресека, удаљености, углова), аналитичка геометрија у равни

Државно такмичење:

- Ефикасни алгоритми за сортирање (quick сортирање, merge сортирање), брзо претраживање (бинарно претраживање, одређивање k-тог најмањег или k-тог највећег елемента)
- Представљање стабала и графова, једноставни „графовски алгоритми“ (обилазак у дубину и ширину, одређивање повезаних компоненти (компоненти повезаности), проналажење петљи, тополошко сортирање графа), бинарно стабло претраживања (претраживачко стабло)
- Динамичко програмирање
- Једноставни геометријски алгоритми (површина полигона, утврђивање да ли тачка припада полигону)

Српска информатичка олимпијада:

- Графовски алгоритми (најкраћи путеви, минимално стабло разапињања, јака повезаност, двострука повезаност, артикулационе тачке графа, Ојлерова тура, упаривање у графу, максимални проток у мрежи)
- Структуре података (редови приоритета, хеш табеле, дисјунктни скупови, сегментна

стабла)

- Геометријски алгоритми (конвексни омотач, тријангулација, пресек полигона, пар најближих тачака)
- Напредне технике проналажења узорка у тексту

Задаци, поставка задатака

Поставке задатака на такмичењима из информатике се обично значајно дуже од поставки задатака из математике, физике и сличних области. Састоје се од описа проблема, формата улазних података за проблем, формата у коме се приказују излазни подаци и одређеног броја примера. Поред тога, обично је проблем могуће решити на више различитих начина, који се разликују по ефикасности и сложености. Због разлика у ефикасности, време извршавања појединих решења се може драстично разликовати. Решења која су спорија треба да донесу такмичару одређен број поена. Мање ефикасна решења се могу извршити само за тест примера чија величина није већа од неке вредности. Због тога последњи део поставке проблема треба да садржи информације о томе колико поена може добити такмичар ако је његово решење такво да ће се извршити за тест примере одређене величине. Уз овај документ прилажемо и поставке проблема за Окружно и Државно такмичење за школску 2015/2016. Годину. На семинару ће бити речи о томе како решавати неке од ових проблема.

Систем BubbleBee

Систем *BubbleBee* (http://bubblebee.rs) је портал за учење програмирања који садржи материјале за учење програмирања на различитим нивоима и који се може користити како у редовној настави, тако и у припреми одређених нивоа такмичења. Систем омогућава ученицима да пошаљу своје изворне кодове који се онда тестирају и њихова тачност се проверава кроз низ тест примера.

На порталу постоји неколико основних курсева. *Hour of Code* је курс за најмлађе и све оне који желе да на једноставан и сликовит начин уђу у свет програмирања. *C++ за почетнике* је курс програмирања намењен пре свега апсолутним почетницима, али и свима који желе да обнове или употпуне своје разумевање основних појмова у програмирању. *Збирка задатака* обухвата почетне нивое учења програмирања и почетне нивое припреме за такмичарско програмирање. Циљна група збирке су ученици прве две године средњих школа у којима се програмирање изучава по специјализованим плановима и програмима, такмичари основци, ученици трећег разреда гимназија и сви почетници заинтересовани за алгоритамско програмирање. Курс *Увод у алгоритме* покрива широк спектар алгоритама и техника програмирања. Циљ курса је упознавање са најважнијим алгоритмима који се користе за решавање практичних проблема који се јављају у рачунарским наукама.

На порталу такође постоје колекције задатака са претходних такмичења из програмирања ученика основних и ученика средњих школа у Републици Србији, као и са такмичења BubbleCup које је организовао Microsoft развојни центар у Србији.

На крају, на порталу постоје и предлози пројеката за тимску или самосталну израду који су одабрани тако да осликавају реалне софтверске проблеме присутне у данашњој ИТ индустрији.

УПУТСТВО

Пред вама је 6 задатака чији је преглед дат у табели на следећој страни. **Ученици Б категорије решавају само прва 3 задатка а ученици А категорије само последња 3 задатка**. Уколико ученик који по пропозицијама припада Б категорији **преда бар један задатак из А категорије, сматра се да је одлучио да пређе у А категорију**. Задатке морате радити самостално и **коришћење интернета није дозвољено**. Израда задатака траје 4 сата.

Фолдер у коме памтите решења задатака (и који ће бити преузет од стране ваших професора) **мора имати име идентично вашем корисничком имену** под којим сте радили овогодишње квалификације. У том фолдеру се памте искључиво **.pas**, **.c** или **.cpp** изворни (source) кодови **чија имена морају бити као у датој табели**. Оставите себи времена како бисте проверили да ли сте урадили све како треба.

Тестирање задатака се обавља под оперативним системом Linux, **на истој машини и систему као и за овогодишње квалификације**, уз коришћење званичних компајлера **FreePascal 2.4.4-3.1** и **GCC 4.6.3**. Приликом компајлирања, користе се следеће команде:

- За Pascal фајлове: *fpc -dEVAL -XS -O2 -o[ime_fajla] [ime_fajla].pas*
- За C фајлове: *gcc -DEVAL -static -O2 -o [ime_fajla] [ime_fajla].c -lm*
- За C++ фајлове: *g++ -DEVAL -static -O2 -o [ime_fajla] [ime_fajla].cpp*

Меморијска ограничења у задацима се односе на **укупну меморију** коју ваш програм користи у било ком тренутку – *HEAP* + *STACK* + величина самог програма. Осим ових ограничења, **не постоји** посебно ограничење за *STACK* меморију. Максимална дозвољена величина изворног кода је **100 KB**.

НАПОМЕНЕ

- Подаци се читају/исписују преко стандардног улаза и излаза - **немојте користити фајлове!**
- Излазни подаци морају бити **тачно** у облику датим у опису задатка. **Немојте исписивати додатне ствари** попут "Тражени број је...".
- На крају програма **обавезно уклонити** "readln;" и "system('pause');" наредбе!
- Уколико је потребно користити 64-битне бројеве, користите **int64** у Pascal-у, односно **long long** у C/C++-у; обратите пажњу да *long* у C/C++-у не мора увек бити 64-битни тип. Уколико за учитавање/испис 64-битних бројева у C/C++-у користите функције *scanf*/*printf*, потребно је употребити спецификатор **%lld**.
- Ознака за почетак нове линије може бити један карактер '\n' **ИЛИ** два карактера "\r\n". Ваш програм мора исправно обрађивати **оба случаја** приликом учитавања података. Приликом учитавања препоручују се функције *read*/*readln* (Pascal) и *scanf* (C/C++).
- У C/C++ кодовима, користити <iostream> а не <iostream.h>. Такође, морате **експлицитно include-овати** све бибилиотеке чије функције користите (нпр. <cstring>, <cstdlib>, <algorithm>). У неким окружењима (DevC++) ваш код ће радити и без тога **али не и на званичном систему!**
- У C/C++ кодовима функција **main** мора бити декларисана као "**int main()**" а не као "void main()"/"main()". Такође, ова функција мора враћати вредност, тј. морате имати "**return 0;**"

ЗАДАТАК	ДАВИС ЗОНА	(НОВИ) ВАВИЛОН	АБАКУС	ТАЈНА ПРОСТОРИЈА	ФАРАОН	ЗИД
категорија	Б	Б	Б	А	А	А
назив *source* кода	davis.pas davis.c davis.cpp	vavilon.pas vavilon.c vavilon.cpp	abakus.pas abakus.c abakus.cpp	tajna.pas tajna.c tajna.cpp	faraon.pas faraon.c faraon.cpp	zid.pas zid.c zid.cpp
улаз	**СТАНДАРДНИ УЛАЗ (stdin)**					
излаз	**СТАНДАРДНИ ИЗЛАЗ (stdout)**					
временско ограничење	1.0 sec	1.0 sec	1.0 sec	1.0 sec	2.0 sec	0.1 sec
меморијско ограничење	64 MB	64 MB	64 MB	64 MB	64 MB	64 MB
број поена	100	100	100	100	100	100

Назив проблема: Давис зона

Временско ограничење: 1 s
Меморијско ограничење: 64 MB

У старим временима, монголски владар Петар Кан је владао Тајном Комисијом са чврстом песницом. Предани узурпатор Давис је по сваку цену решен да га збаци са трона. Опште је познато да Давис добија надљудске моћи када уђе у његову зону—зато је Петар решен да по сваку цену спречи Дависа да уђе у зону.

Дависова зона се налази унутар тајне просторије (која је потпуно окружена зидовима) облика правоугаоника страница a и b. Просторија је подељена на $a \cdot b$ квадратних поља димензија 1×1. Поља су нумерисана од $(1, 1)$ (горње-лево) до (a, b) (доње-десно). Зона је правоугаоник димензија c и d, са горњим левим пољем (x, y) (овај правоугаоник је потпуно садржан у просторији).

Давис креће са поља (d_x, d_y), и у сваком кораку ће се померити за Δx поља надоле, односно Δy поља удесно. Уколико би Давис тим померањем ударио у зид, одбиће се под истим углом под којим је ударио у њега. Ако Давис удари у зид више од k пута без да уђе у зону, одустаће овога пута.

Петра сада занима да ли ће Давис икада бити у зони, и ако хоће, занима га поље на коме ће бити када први пут буде у зони, да би знао да на њега постави стражаре.

Улаз. У првом реду стандардног улаза налазе се два природна броја a и b, који представљају број редова и колона просторије, редом. У другом реду налазе се два природна броја c и d, који представљају број редова и колона Дависове зоне, редом. У трећем реду налазе се два природна броја x и y, који представљају ред и колону горњег левог поља зоне, редом. У четвртом реду налазе се два природна броја d_x и d_y, који представљају ред и колону поља са кога Давис креће. У петом реду налазе се три цела броја Δx, Δy и k, који представљају број поља надоле и удесно за колико ће се Давис померити у сваком кораку, као и ограничење за број удара у зид.

Излаз. У првом и једином реду стандардног излаза потребно је исписати два природна броја h_x и h_y, који представљају ред и колону поља на коме ће Давис бити први пут када се нађе у зони, редом. Уколико Давис не успе да уђе у зону, потребно је исписати "`NE`" (без наводника).

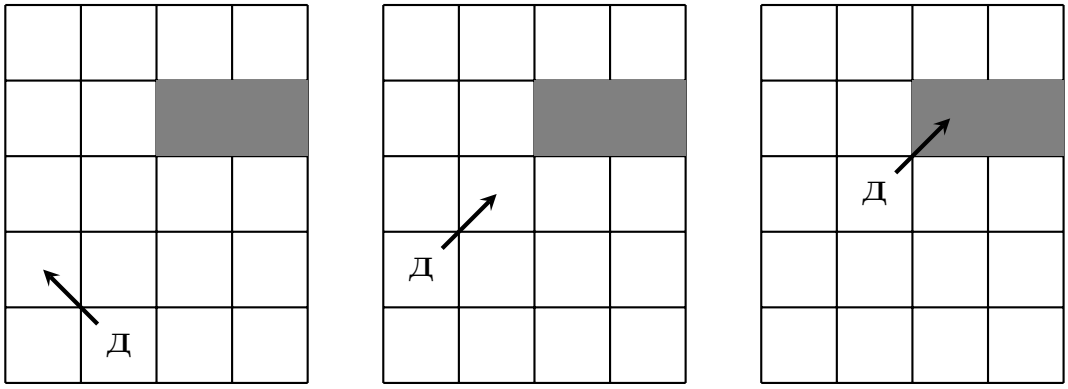
Пример 1.

Улаз	Излаз

`5 4`
`1 2`
`2 3`
`5 2`
`-1 -1 1`
	`2 3`

Објашњење. На сликама испод налази се опис тест примера; на свакој слици Дависова позиција означена је словом **Д**, поља зоне су затамњена, а стрелица означава поље на које ће Давис отићи у следећем кораку.

Давис најпре удара у зид у пољу (4, 1); два корака након тога успешно улази у зону, и то у пољу (2, 3). Притом није ударио у зид више од једном, тако да је ово коначно решење које треба исписати.



Ограничења.

- $1 \leq a, b, c, d \leq 1000$;
- $1 \leq c, x, d_x \leq a$;
- $1 \leq d, y, d_y \leq b$;
- $0 \leq k \leq 1000$;
- $\Delta x, \Delta y \in \{-1, 0, 1\}$;
- Дависова зона ће увек бити обухваћена просторијом.

Напомена. Тест примери су подељени у три дисјунктне групе:

- У тест примерима вредним 20 поена важиће $\Delta x = 0$ или $\Delta y = 0$.
- У тест примерима вредним 20 поена важиће $k = 0$.
- У тест примерима вредним 60 поена нема додатних ограничења.

Назив проблема: (Нови) Вавилон

Временско ограничење: 1 s
Меморијско ограничење: 64 MB

Непосредно пре 626. године п.н.е. на подручју Месопотамије је избио велики грађански сукоб. Побуњени грађанин Напобаласар је желео да збаци са власти тадашње владаре Асирце и да оснује Халдејско царство, односно Нови Вавилон. Интересантна чињеница је да је сваки од грађана тадашње Месопотамије имао јединствени матични број грађана који је цео број из интервала $[1, m]$.

Напобаласар је такође био високи свештеник вуду религије Авалон која много верује у округле столове и веровао је да постићи победу ако и само ако успе да грађане распореди на столу у облику осмице који је састављен од два кружна стола капацитета n, тако да он седи на месту које спаја те столове и да важи да је најмањи заједнички садржалац јединствених матичних бројева грађана на једном столу једнак највећем заједничком делиоцу јединствених матичних бројева грађана на другом столу и да је та вредност управо једнака вредности његовог јединственог матичног броја грађана. Напобаласар тренутно нема времена да израчуна на колико начина може распоредити грађане на оваквом столу, па вас моли за помоћ како би саградио Нови Вавилон. Пошто овај број може бити веома велик, ваш задатак је да испишете остатак при дељењу овог броја бројем $10^9 + 7$.

Улаз. У првој и јединој линији стандардног улаза се налазе три цела броја n, m и id која редом представљају капацитет сваког од кружних столова, број људи који живе у Месопотамији и Напобаласаров јединствени матични број грађана.

Излаз. У првој и јединој линији стандардног излаза треба исписати остатак при дељењу броја начина на који Напобаласар може изабрати грађане по описаном веровању, бројем $10^9 + 7$.

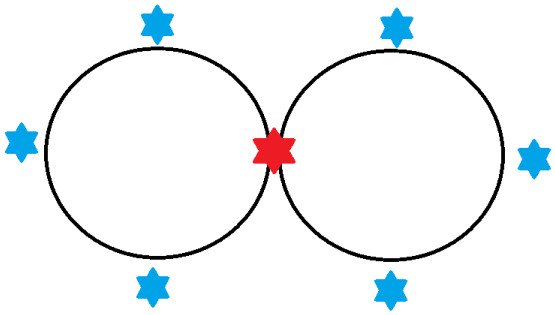
Пример 1.

Улаз	Излаз
`4 48 12`	`720`

Пример 2.

Улаз	Излаз
`4 38 12`	`0`

Објашњење.



На слици изнад су илустровани дати пример, где црвено место представља Напобаласара. У првом тест примеру, Напобаласар може за једним столом да распореди људе са јединственим матичним бројевима грађана $1, 2, 3, 4, 6$ тако да њихов најмањи заједнички садржалац износи 12 и он то може

урадити на $5 \cdot 4 \cdot 3 = 60$ начина, а за другим столом може да распореди људе са матичним бројевима $24, 36, 48$ тако да је њихов највећи заједнични делилац једнак 12 и он то може урадити на $3 \cdot 2 \cdot 1 = 6$ начина. Коначан резултат је $(60 \cdot 6) \cdot 2$ пошто прву групу људи може ставити или на леви или на десни сто.
У другом тест примеру, Напобаласар никако не може за другим столом да распореди 3 грађана тако да њихов највећи заједнички делилац износи 12, па је број начина 0.

Ограничења.

- $1 \leq n \leq 10^6$
- $1 \leq m \leq 10^9$
- $1 \leq id \leq m$

Напомена. Сваки од грађана има **јединствен** матични број. Ниједан сто не сме имати празно место. Тест примери су подељени у три дисјунктне групе:

- У тест примерима вредним 20 поена важиће $n = 2$.
- У тест примерима вредним 20 поена важиће $m \leq 10^3$.
- У тест примерима вредним 60 поена нема додатних ограничења.

Назив проблема: Абакус

Временско ограничење: 1.0 s
Меморијско ограничење: 64 MB

Стари Вавилонци су користили једну варијанту направе Абакус за компликована математичка израчунавања. Абакус се састоји од n куглица нумерисаних бројевима од 1 до n. Свака од куглица се може налазити на једној од 10 позиција (нумерисаних бројевима од 0 до 9), а те позиције говоре о вредности коју има одговарајућа куглица (то су вредности од 0 до 9). Куглице су подељене у m дисјунктних група, нумерисаних бројевима од 1 до m (поједине групе могу бити празне). Корисник има на располагању две врсте операција:

- **Постави BrG Poz**, овом командом постављају се све куглице из групе BrG у позицију Poz (тј. поставља вредност свих куглица из те групе на Poz). Ова операција има цену $C1$.
- **Промени BrK Poz**, овом командом се премешта само куглицу са редним бројем BrK на позицију Poz (тј. мења се њена вредност на Poz). За ову операцију корисник плаћа цену једнаку "кружном растојању" између досадашње позиције те куглице и нове позиције. "Кружно растојање" је најкраће растојање између позиција, ако би оне биле поређане на кругу у складу са нумерацијом. Нпр. кружно растојање између 0 и 7 је 3, између 1 и 9 је 2, а имеђу 7 и 3 је 4.

Добили сте опис Абакуса (број куглица, цену операције **Постави** (C1) и поделу куглица по групама), почетно стање куглица (тј. позицију у којој се налази свака од куглица на почетку) и крајње стање куглица (тј. позиције на којима треба на крају да се налазе). Потребно је да одредите минималну цену да се од задатог почетног стања куглица (тј. почетних позиција куглица) дође до крајњег стања куглица.

Улаз. У првом реду стандардног улаза се налази број куглица (n) и цена операције Постави (C1). У другом реду се налази n природних бројева, а i-ти од њих представља редни број групе у којој се налази i-та куглица. У трећем реду се налази n бројева, а i-ти од тих бројева представља почетну позицију (тј. почетну вредност) i-те куглице. У четвртом реду се налази n бројева, а i-ти од тих бројева представља крајњу позицију i-те куглице.

Излаз. На стандардни излаз исписати само један број - минималну збирну цену операција које треба извести да би се куглице довеле на крајње позиције.

Пример 1.

Улаз	Излаз
8 3 6 5 6 5 5 6 1 2 5 1 5 2 3 5 8 0 0 1 0 2 3 8 4 8	10

Објашњење. Куглице из групе 5 не морамо уопште премештати. Куглице из групе 6 пребацујемо командом **Постави 6 0** у позицију 0, а затим командом **Промени 6 8** и плаћамо цену 5 (3+2). Куглицу из групе 1 пребацујемо командом **Постави 1 4** и плаћамо цену 3 (ако би користили команду **Промени 7 4** платили би 4). Куглицу из групе 2 пребацујемо командом **Промени 8 8** и плаћамо цену 2 (командом **Постави 2 8** платили би 3).

Ограничења.

- $1 \leq n \leq 10^6$
- $1 \leq m \leq 1000$
- $1 \leq C1 \leq 10^6$

Напомена. Тест примери су подељени у три дисјунктне групе

- у тест примерима који вреде 30 поена важиће да се све куглице налазе у једној групи (тј. $m = 1$).
- у тест примерима који вреде 30 поена сваку од група чини блок узастопних куглица (тј. блок куглица чије су ознаке узастопни природни бројеви).
- у тест примерима који вреде 40 поена нема додатних ограничења.

Назив проблема: Тајна просторија

Временско ограничење: 1 s
Меморијско ограничење: 64 MB

Египћани су веома познати по њиховим пирамидама које и дан данас представљају праву енигму за историчаре. Међутим, историчари верују да су на прагу великог открића, али им је потребна ваша помоћ. Наиме, проучавајући унутрашњост једне од пирамида, они су наишли на зид на коме су у једном реду исписани разни бројеви. Добро позната теорија о једној од најпознатијих краљица Египта, свемоћној Тамари, гласи да је Тамара била опчињена низовима бројева где сви бројеви у низу имају бар једну исту цифру. Историчари верују да је низ бројева написан на зиду који су нашли баш дело краљице Тамаре и да представља загонетку чије решење води до Тајне просторије где се налази скривено сво краљичино благо.

Историчари су убеђени да је решење загонетке најдужи подниз узастопних елемената датог низа бројева за који важи да постоји бар једна цифра која се налази у свим бројевима тог подниза. Пошто они нису успели да нађу овај подниз, они су вас замолили за помоћ.

Улаз. У првом реду стандардног улаза се налази цео број N који представља дужину низа бројева написаног на зиду. У другом реду се налази N целих бројева који представљају бројеве низа.

Излаз. У први и једини ред стандардног излаза исписати један цео број који представља дужину најдужег подниза узастопних бројева датог низа за који важи да постоји бар једна цифра која се налази у свим бројевима тог подниза.

Пример 1.

Улаз	Излаз
5 123 222 334 443 538	3

Пример 2.

Улаз	Излаз
3 13 456 3	1

Објашњење. У првом тест примеру подниз $[334, 443, 538]$ је дужине 3 и сви његови елементи садрже цифру 3. Не постоји подниз дужине 4 чији бројеви имају бар једну исту цифру.
У другом тест примеру никоја два суседна броја немају заједничку цифру, па је решење за тај тест пример 1.

Ограничења.

- $1 \leq N \leq 10^6$.
- Сви бројеви у датом низу бројева су између 1 и 10^9

Напомена. Тест примери су подељени у четири дисјунктне групе:

- У тест примерима вредним 20 поена важи $1 \leq N \leq 100$.
- У тест примерима вредним 30 поена важи $1 \leq N \leq 1000$.
- У тест примерима вредним 20 поена важи да су сви бројеви у датом низу једноцифрени.
- У тест примерима вредним 30 поена нема додатних ограничења.

Назив проблема: Фараон

Временско ограничење: 2 s
Меморијско ограничење: 64 MB

Према историчарима, у старој египатској држави, фараони су морали да докажу представницима племена да су довољно снажни да воде државу тако што су пред њима морали да се боре са биком и да му ишчупају реп. Међутим, мање познато је да су фараони морали да докажу и да су довољно паметни решавајући древни математички задатак.

У древном математичком задатку су постављена два броја p и q. Фараон је морао да одговори у колико најмање специјалних математичких операција може од броја p да се добије број q. Специјалне математичке операције су следеће:

- Број t може да се помножи било којим простим бројем r, како би се добио број $t \cdot r$
- Број t може да се подели било којим простим бројем r који дели t, како би се добио број t/r

Наравно, фараон је знао да су прости бројеви они природни бројеви већи од 1 који су дељиви само са 1 и самим собом.

Како би му отежали посао, представник сваког племена му задаје нове бројеве p и q, и фараон мора тачно да одговори на сваки задати пар бројева.

Историчари су пронашли стари запис на којем су записани постављени бројеви фараону и знајући да је фараон доказао да је паметан (одговорио је тачно на сва питања) ваш задатак је да пронађете фараонове одговоре за сваки пар постављених бројева p и q.

Улаз. У првом реду стандардног улаза ће се налазити цео број N који означава број племена (број постављених питања). У наредних N редова ће се налазити по два цела броја p и q.

Излаз. На стандардни излаз потребно је исписати N редова где се у k-том реду налази одговор фараона на k-то питање из улаза.

Пример 1.

Улаз	Излаз
5	0
1 1	3
16 2	2
2 3	1
3 9	2
6 10	

Ограничења.

- $1 \leq N \leq 10^6$
- $1 \leq p, q \leq 10^6$

Напомена. Тест примери су подељени у три дисјунктне групе:

- У тест примерима вредним 20 поена важи $N \leq 10^4$
- У тест примерима вредним 40 поена важи $p, q \leq 10^5$
- У тест примерима вредним 40 поена нема додатних ограничења.

Назив проблема: Зид

Временско ограничење: 0.1 s
Меморијско ограничење: 64 MB

Налазимо се у години 80. пре нове ере. Цела Галија је под влашћу Римљана... Осим N села, која се још увек успешно одупиру. После пуно покушаја, Римљани су одустали од освајања свих села. Одлучили су да ће направити велики зид, који ће одвојити нека села од њихове територије, а затим освојити преостала.

За свако село је позната његова позиција (X_i, Y_i) (у километрима) и цена опремања војске (у златницима) довољне да га освоји W_i. Римска војна тактика је толико добра да при овоме не очекују губитке – за освајање више села цена опремања војске једнака је највећој цени за појединачна села. На пример, уколико ван зида остану три села, са $W_i = 50, 20, 30$, неопходна војска ће коштати 50 златника.

Зид мора бити правоугаоног облика, са страницама паралелним координатним осама и центром у $(0, 0)$ (нико није рекао зашто, али сигурно постоји добар разлог). Један километар зида кошта C златника. Могуће је направити зид који је по једној или обе димензије веома кратак и ограђује само села на одговарајућој оси – у том случају, сматра се да је дужина "кратке" странице зида 0 (нпр. да би се оградила села на $(0, 1)$ и $(0, -1)$, довољан је зид са страницама дужине 2 и "занемарљиво мало", са укупном дужином 4).

Ваше задужење је да одредите најмању цену за коју је могуће остварити овај план (саградити зид и освојити сва села ван зида).

Улаз. У првом реду стандардног улаза ће се налазити два цела броја N и C, који означавају редом број села и цену километра зида.

У наредних N редова ће се налазити по три цела броја X_i, Y_i, W_i, који означавају редом координате и цену освајања села (у реду $i + 1$ ће се налазити подаци за i-то село).

Излаз. На стандардни излаз исписати један број – минималан број златника које је неопходно потрошити да би се план остварио.

Пример 1.

Улаз	Излаз
4 2 2 -1 12 3 2 60 -5 -5 6 5 6 2	46

Пример 2.

Улаз	Излаз
2 10 0 0 5 1 1 1	1

Објашњење. У првом примеру, оптимално је саградити зид са теменима у $(\pm 3, \pm 2)$. Укупна дужина зида је 20, тако да кошта 40 златника. Ван зида остају треће и четврто село, са ценама освајања 6 и 2, тако да је војску потребно платити 6 златника.

У другом примеру, могуће је оградити прво село зидом са две занемарљиво кратке странице, који

кошта нула златника, и платити један златник за војску.

Ограничења.

- $1 \leq N \leq 10^6$
- $1 \leq C \leq 200$
- $-10^6 \leq X_i, Y_i \leq 10^6$
- $0 \leq W_i \leq 10^6$

Напомена. Тест примери су подељени у четири дисјунктне групе:

- У тест примерима вредним 20 поена важи $N \leq 100$, $-100 \leq X_i, Y_i \leq 100$.
- У тест примерима вредним 20 поена важи $N \leq 10^3$ и сва села су на x-оси ($Y_i = 0$).
- У тест примерима вредним 30 поена важи $N \leq 10^3$.
- У тест примерима вредним 30 поена нема додатних ограничења.

ЗАДАТАК	МАЈЕ	КРАЉ АРТУР	ТИХО БРДО	ВАТРОМЕТ	МАГИЈА	ПОДЗЕМНЕ ВОДЕ
категорија	Б	Б	Б	А	А	А
назив *source* кода	maje.pas maje.c maje.cpp	artur.pas artur.c artur.cpp	tiho.pas tiho.c tiho.cpp	vatromet.pas vatromet.c vatromet.cpp	magija.pas magija.c magija.cpp	podzemne.pas podzemne.c podzemne.cpp
улаз	**СТАНДАРДНИ УЛАЗ (stdin)**					
излаз	**СТАНДАРДНИ ИЗЛАЗ (stdout)**					
временско ограничење	0.4 sec	0.1 sec	0.1 sec	0.5 sec	0.1 sec	1 sec
меморијско ограничење	64 MB	64 MB	64 MB	64 MB	64 MB	64 MB
број поена	100	100	100	100	100	100

Назив проблема: Маје

Временско ограничење: 1 s
Меморијско ограничење: 64 MB

Цивилизација маја је једна од најпознатијих древних цивилизација. Ова цивилизација је, између осталог, позната по њиховој, за тај период, веома напредној математици. Маје су биле опчињене неопадајућим низовима бројева, тј. низовима који задовољавају да је сваки елемент низа, сем првог, већи или једнак од претходног у низу.

Маје су веровале да се неке битне особине низа могу открити тако што се низ трансформише у неопадајући низ бројева. Међутим, пошто су веровале да се редослед елемената у низу не сме мењати, маје су користиле специјалну операцију над елементима низа како би га трансформисале. Дозвољена операција узима један од бројева из низа и редом пребацује неколико цифара (могуће њих 0) са почетка броја на његов крај.

Нпр. примењујући ову операцију на број 12345, могуће је добити један од бројева (усправном чртом је означена група цифара која се пребацује на крај броја):

- $|12345 \to 12345$
- $1|2345 \to 23451$
- $12|345 \to 34512$
- $123|45 \to 45123$
- $1234|5 \to 51234$

Ваш задатак је да за дати низ бројева одредите да ли је могуће да се примењујући описану операцију на **сваки елемент низа тачно једном**, добије неопадајући низ елемената.

Улаз. У првом реду стандардног улаза се налази природан број N који представља дужину низа. У другом реду се налази N природних бројева који представљају низ.

Излаз. У првом и јединном реду стандардног излаза ваш програм треба да испише "DA" уколико је могуће трансформисати дати низ у неопадајући, примењујући описану операцију на сваки елемент низа тачно једном. У супротном, у првом реду стандардног излаза ваш програм треба да испише "NE".

Пример 1.

Улаз	Излаз
`5` `3302 142 214 115 600`	`DA`

Пример 2.

Улаз	Излаз
`3` `24 15 33`	`NE`

Објашњење. У првом тест примеру, један од начина да се низ трансформише у неопадајући низ бројева је да се одраде следеће операције над бројевима:

$$33|02 \to 0233,\ 1|42 \to 421,\ 21|4 \to 421,\ 11|5 \to 511,\ |600 \to 600$$

Овим операцијама се добија низ бројева: 233 421 421 511 600, који је неопадајући.

Низ дат у другом тест примеру није могуће трансформисати описаном операцијом у неопадајући низ бројева.

Ограничења.

- $n \leq 10^6$
- сви бројеви у датом низу су између 0 и 10^9, укључујући и ове бројеве

Подзадаци и бодовање. Тест примери су подељени у 4 подзадатка, у којима важе следећа додатна ограничења:

- **подзадатак 1 [10 поена]**: сви бројеви у датом низу су мањи од 10
- **подзадатак 2 [20 поена]**: $n = 2$
- **подзадатак 3 [20 поена]**: сви бројеви у датом низу су мањи од 100
- **подзадатак 4 [50 поена]**: нема додатних ограничења

Напомена. Приметити да се, применом операције на одређени број, цифре 0 могу наћи на почетку броја. У овом случају се не посматрају ове нуле, тј. посматра се број без водећих нула. Такође, приметити да у улазу бројеви неће имати водеће нуле.

Назив проблема: Краљ Артур [artur.c/cpp/pas]

Временско ограничење: *TODO* s
Меморијско ограничење: *TODO* MB [64?]

Краљ Артур припрема своје војнике за одлучујућу битку против Саксонаца. Како би искористиле свој пун потенцијал на бојном пољу, неопходно је за сваку од m армија одабрати по једног команданта. Све армије имају исти број војника, и он износи n. Артур зна да сваки командант мора да добро познаје војнике којима командује, па обично за ту улогу бира неког од војника из саме армије. Напредним прорачунима, схватио је да **укупна моћ** (тј. збир вештина свих команданата) не сме да буде превелика, јер тада може доћи до несугласица међу њима и постоји опасност да се армије сукобе међусобно. Истог трена је направио списак свих могућих избора m команданата и за сваки од избора забележио укупну моћ. Након тога је најсавременијим алгоритмом тог доба, „Бозо сорт"-ом, сортирао списак опадајуће по моћи.

Ипак, седам дана није успео да направи правилан избор. Осмог дана му је било доста, и одлучио је да пита свог мудрог чаробњака-саветника Мерлина за савет. Након дужег глађења седе браде, Мерлин је лупио магичан број k и рекао да је врло очигледно да је прави избор онај k-ти са Артуровог списка (при чему Мерлин зна да број могућих избора није мањи од k). Артуру ово свакако није било очигледно, и одмах је отишао да обавести витезове округлог стола о томе. Прво што су га витезови питали било је колика ће у том случају бити укупна моћ армије. Помозите Артуру, који је превише узбуђен да би се сетио тачног броја, да им одговори на питање.

Улаз. У првом реду стандардног улаза налазе се бројеви m, n и k, који представљају број армија, број војника у свакој армији, и Мерлинов магичан број, редом. Свака од наредних m линија садржи n бројева који означавају вештине војника. Бројеви у i-том реду се односе на војнике из i-те армије.

Излаз. У првом и једином реду стандардног излаза потребно је исписати одговор на витешко питање, тј. укупну моћ одабране армије.

Пример 1.

Улаз	Излаз
2 3 6 2 3 1 4 5 6	7

Објашњење. Артур на располагању има две армије, и по три војника у свакој од њих. Кандидати за команданта прве армије су војници са вештинама из скупа {2, 3, 1}, а друге армије војници са вештинама из скупа {4, 5, 6}. Артуров сортиран списак свих могућих укупних моћи је: [9, 8, 8, 7, 7, **7**, 6, 6, 5]. Тражени број је шести број на списку, тј. 7.

Ограничења.

- $1 \leq m \leq 10$.
- $1 \leq n \leq 1000$.
- $1 \leq k \leq 10^{18}$.
- Вештине војника су ненегативни цели бројеви мањи или једнаки 1000.

Подзадаци и бодовање. Тест примери су подељени у 5 подзадатака, у којима важе следећа додатна ограничења:

- **ПОДЗАДАТАК 1 [12 ПОЕНА]**: $m = 1$.
- **ПОДЗАДАТАК 2 [15 ПОЕНА]**: $k \leq 2$.
- **ПОДЗАДАТАК 3 [24 ПОЕНА]**: $m \leq 4, n \leq 30$.
- **ПОДЗАДАТАК 4 [28 ПОЕНА]**: $n \leq 6$.
- **ПОДЗАДАТАК 5 [21 ПОЕНА]**: Нема додатних ограничења.

Назив проблема: Тихо Брдо

Временско ограничење:	0.1 s
Меморијско ограничење:	64 MB

У средњевековном добу, у малом градићу Тихом Брду живело је чудовиште под називом Глутон, које је волело да терорише становнике овог града. Многи витезови су покушавали да убију Глутона, али ни један није успео да преживи тај сусрет.

Једног дана, храбри архитекта Давис је одлучио да нападне Глутона. Сазнао је да се Глутон заправо састоји из гомиле *делова*, који су међусобно спојени на тај начин да сваки део може *контролисати* неке друге делове. Глутон има тачно један део који представља његов *мозак*, и тај део није контролисан од стране иједног другог дела. Такође, Глутон може имати један или више делова који не контролишу ни један други део; ови делови представљају његове *канце*. За сваку канцу постоји **тачно један скуп спојева** којим је мозак (директно или индиректно) контролише. Уколико Давис успешно одсече мозак од контролисања свих канци, онда Глутон остаје потпуно безопасан.

Давис је употребио своје пројектне способности и закључио да је сваком од Глутонових спојева $x \to y$ могуће приписати *снагу* w неопходну да би се тај спој уништио. Дависова жеља је да уложи **минимално снаге** за одсецање мозга од канци, па га занима које спојеве, **слева на десно**, треба да пресече да би то постигао. Уколико постоји више тачних решења, Давис би волео да уложена снага на почетку буде што је мања могућа, тј. тражи се **лексикографски најмање решење** од свих тачних решења.

За дата два низа, a и b, величина n и m, кажемо да је a *лексикографски мањи од* b уколико или постоји нека позиција i тако да је су до те позиције низови исти, а низ a је по i-том члану мањи, тј.

$$(a_i < b_i) \wedge (\forall j < i.\ a_j = b_j)$$

или се низови a и b поклапају на свим компатибилним позицијама, с тим да је низ a краћи, тј.

$$(n < m) \wedge (\forall j \leq n.\ a_j = b_j)$$

Примери: $\{1, 2, 3\} < \{2, 2, 3\}$; $\{1, 2, 3, 4\} < \{1, 2, 4, 4\}$; $\{1, 2, 3\} < \{1, 2, 3, 4\}$.

Заслепљен славом коју би стекао и покличима "Хвала храбри Дависе!" од становника уколико би онеспособио Глутона, Давис нема довољно стрпљења да израчуна решење. Замолио вас је за помоћ.

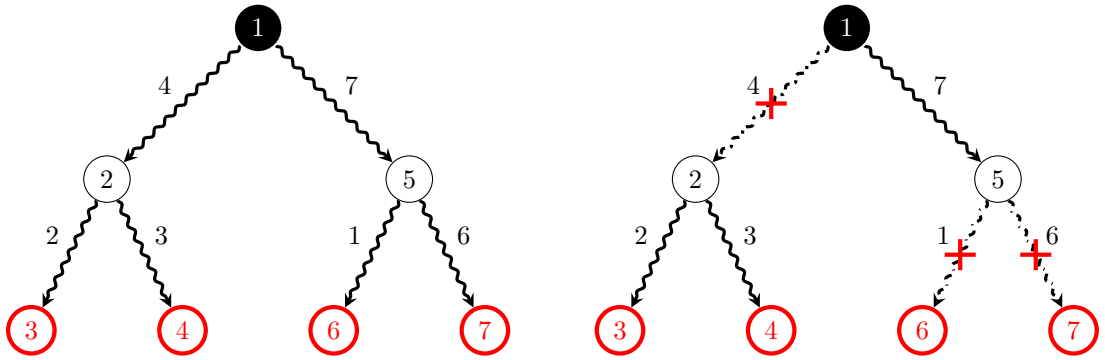
Улаз. У првом реду стандардног улаза налази се природан број n, који представља број Глутонових делова. Део са индексом 1 увек представља мозак. Затим следе описи делова, од мозга до дела са индексом n, редом, на следећи начин: у првом реду се налази цео број m_i, који представља број делова који су под директном контролом текућег дела. Уколико важи $m_i = 0$, онда је текући део канца. У супротном, у другом реду се налази низ a од m_i природних бројева, који представља, **слева на десно**, индексе делова под директном контролом текућег дела. У трећем реду се налази низ w од m_i природних бројева, тако да j-ти члан низа w, w_j, означава снагу неопходну за уништење споја којим текући део контролише j-тог члана низа a, a_j.

Излаз. У првом реду стандардног излаза потребно је исписати минималну укупну снагу неопходну да би Давис онеспособио Глутона. У наредном реду неопходно је исписати *лексикографски најмањи* низ снага које ће требати Давису да би одсекао све неопходне спојеве, слева на десно, редом. Чланове низа треба одвојити размаком.

Пример 1.

Улаз	Излаз
7 2 2 5 4 7 2 3 4 2 3 0 0 2 6 7 1 6 0 0	11 4 1 6

Објашњење. На сликама испод налази се опис тест примера; лева слика представља почетно стање Глутона, а десна представља оптимално решење. Мозак је означен црном позадином, а канче црвеном бојом. Прецртани и испрекидани спојеви представљају оне спојеве које Давис треба да уништи. Постоје *два решења* која захтевају минималну укупну снагу 11; та решења су, слева на десно, $\{4, 7\}$ и $\{4, 1, 6\}$. Од та два решења, бирамо лексикографски мање.



Ограничења.

- $2 \leq n \leq 10^5$;
- $1 \leq w_j \leq 10^9$;
- Укупан број спојева ће увек бити $n - 1$, и увек ће за сваку канџу постојати тачно један скуп спојева који је повезују са мозгом.

Подзадаци и бодовање. Тест примери су подељени у четири подзадатка, у којима важе следећа додатна ограничења:

- **ПОДЗАДАТАК 1 [10 ПОЕНА]**: Сви спојеви ће имати исту потребну снагу за уништење;
- **ПОДЗАДАТАК 2 [10 ПОЕНА]**: Једино мозак може директно контролисати више од једног дела Глутона;
- **ПОДЗАДАТАК 3 [30 ПОЕНА]**: $n \leq 1.000$;
- **ПОДЗАДАТАК 4 [50 ПОЕНА]**: Нема додатних ограничења.

Уколико на свим тест примерима једног подзадатка израчунате тачну укупну снагу, **добијате 60% поена за тај подзадатак**. Уколико коректно одредите лексикографски минимално решење на свим тест примерима једног подзадатка, **добијате 40% поена за тај подзадатак**.

Назив проблема: Ватромет

Временско ограничење:	*TODO* s
Меморијско ограничење:	*TODO* MB

Стари кинески филозоф, који се у слободно време бави прављењем ватромета, сада треба да направи фитиљ (да би могао да их безбедно упали). Он има канап дужине N метара, из ког планира да исече део. Како је прављење фитиља посао који захтева велику прецизност, он жели да сече овај канап тако да мора да мери само дужине које су цели бројеви у метрима – фитиљ мора почети и завршити се на целобројном растојању од почетка канапа. За сваки метар канапа, он зна цео број A_i, који представља његову "поузданост" (метар i обухвата део канапа између позиција на растојањима i и $(i+1)$ метара од почетка канапа).

Постоји Q различитих ватромета које би филозоф могао да користи. За сваки од њих постоје услови које фитиљ мора да испуни – не сме да буде ни превише кратак ни превише дуг, а мора и да буде довољно поуздан. За ватромет i, ови услови су дефинисани целим бројевима L_i, H_i, T_i на следећи начин:

- Дужина фитиља мора бити већа или једнака L_i и мања или једнака H_i,
- Средња вредност поузданости фитиља мора бити барем T_i.

Филозоф жели да одабере један ватромет за који ће направити фитиљ. Због тога, за сваки ватромет га интересује га на колико начина за њега може да направи фитиљ од канапа који му је на располагању. Пошто је заузет и нема много времена, Ваш задатак је да то одредите уместо њега.

Улаз. У првом реду стандардног улаза налази се број N, који представља дужину канапа. У другом реду се налази N целих бројева, где је i-ти број поузданост i-тог метра канапа A_i. У трећем реду се налази Q – број ватромета за које се тражи решење. У наредних Q редова се налазе по три цела броја: у i-том се налазе L_i, H_i и T_i, који представљају горе описане услове за ватромет i.

Излаз. На стандардни излаз исписати Q целих бројева, где је i-ти од исписаних бројева број начина да се исече фитиљ који испуњава услове за ватромет i.

Пример 1.

Улаз	Излаз
5 3 2 3 1 2 2 1 1 2 2 3 2	4 6

Објашњење. За први ватромет, филозоф може да исече било који метар канапа осим четвртог. За други, потребан му је део који је или дужине 2 (довољно поуздани су делови који почињу на растојању 0, 1 и 2 од почетка, са средњом поузданости редом 2.5, 2.5 и 2) или дужине 3 (сва три су довољно поуздана).

Ограничења.

- $1 \leq N \leq 100$
- $1 \leq Q \leq 10^6$
- $1 \leq A_i \leq 10^9$
- $1 \leq L_i \leq H_i \leq N$

- $1 \leq T_i \leq 10^9$

Подзадаци и бодовање. Тест примери су подељени у 3 подзадатка, у којима важе следећа додатна ограничења:

- **ПОДЗАДАТАК 1 [? ПОЕНА]**: $Q \leq 100$
- **ПОДЗАДАТАК 2 [? ПОЕНА]**: $Q \leq 10000$
- **ПОДЗАДАТАК 3 [? ПОЕНА]**: Нема додатних ограничења.

Назив проблема: Магија

Временско ограничење: 0.1 s
Меморијско ограничење: 64 MB

У средњем веку се много причало о злим вештицама које су користиле црну магију да науде људима. Знало се да су оне организоване у хијерархијску структуру, тј. свака зла вештица, осим главне вештице Баба Јага, је имала тачно једну надређену вештицу којој је морала да говори све своје зле планове пре извршавања. Исто тако, свака вештица је знала које су јој вештице подређене.

За разлику од тога што се много знало о злим вештицама, много мање се знало о вештицама које су користиле белу магију да би се бориле против злих вештица и спречиле их у њиховој намери да науде људима. Најпознатија вештица која је користила белу магију је била магична Соња. Она је била једина особа која је знала да изведе магију која одузима сву моћ злој вештици, те је онеспособи у намери да начини неку озбиљну штету. Једино ограничење је било да Соња може искористити ову чин само на злим вештицама које немају ниједну подређену вештицу, јер у супротном би све подређене вештице заједно напале Соњу.

У једном тренутку су се све зле вештице договориле да изврше нападе и свака вештица је одредила којег дана ће искористити своју најразорнију чин. Соњин задатак је да онеспособи све вештице у њиховој намери. Она мора да искористи своју специјалну чин на свакој вештици пре него што та вештица искористи своју чин. Као што је претходно наведено, Соња може да искористи чин само на вештицама које нису никоме надређене. Када онеспособи једну вештицу, та вештица се уклања из хијерархијске структуре злих вештица (те њена надређена вештица има за једну мање подређену вештицу).

Свака зла вештица има свој јединствени редни број, а зна се да Баба Јага има редни број 1. Соња на почетку зна изглед хијерархијске структуре злих вештица и за сваку вештицу зна ког дана ће искористити своју најразорнију чин. Соњина моћ је ограничена, те може да искористи своју специјалну чин само једном дневно. Ваш задатак је да помогнете Соњи и препоручите распоред којим треба да напада вештице, тј. да јој за сваки дан кажете редни број вештице на коју мора да баци своју чин.

Улаз. У првом реду стандардног улаза се налази природан број N. У другом реду стандардног улаза се налази N бројева где i-ти број, t_i, представља редни број дана у коме ће вештица са редним бројем i искористити своју разорну чин. У наредних $N-1$ редова се налази по два природна броја p и q ($1 \leq p, q \leq N$) који означавају да је вештица са редним бројевим p надређена вештици са редним бројем q.

Излаз. Уколико је могуће да Соња онеспособи сваку вештицу пре него што та вештица употреби своју чин, потребно је у једном реду исписати N бројева где i-ти број представља редни број зле вештице коју Соња треба да онеспособи i-тог дана (уколико постоји више таквих редоследа исписати било који). Уколико то није могуће, исписати поруку "Pobedila je crna magija"

Пример 1.

Улаз	Излаз
5 10 7 2 6 2 4 2 1 4 4 5 1 3	5 3 2 4 1

Пример 2.

Улаз	Излаз
5 10 7 9 2 8 4 2 1 4 4 5 1 3	Pobedila je crna magija

Објашњење. У првом тест примеру првог дана се онеспособи вештица са редним бројем 5, те другог дана вештица са редним бројем 3. Приметимо да смо вештицу са редним бројем 3 онеспособили у последњем тренутку, јер да смо покушали да је онеспособимо један дан касније она би већ искористила чин. После вештице са редним бројем 3 редом смо онеспособили вештице са редним бројевима 2, 4 и 1 и ниједна од злих вештица није стигла да искористити своју чин.
Приметимо да је и редослед $3, 5, 2, 4, 1$ валидно решење у овом тест примеру.

У другом тест примеру не можемо стићи да онеспособимо вештицу са редним бројем 4 пре него што она искористи своју чин.

Ограничења.

- $1 \leq n \leq 10^5$
- $1 \leq t_i \leq 10^9$

Подзадаци и бодовање. Тест примери су подељени у 4 подзадатка, у којима важе следећа додатна ограничења:

- **подзадатак 1 [20 поена]**: Свака вештица има највише једну директно подређену вештицу.
- **подзадатак 2 [20 поена]**: Вештица ће увек искористити своју чин тек након што њене подређене вештице искористе своје чини и $n \leq 1000$
- **подзадатак 3 [30 поена]**: n је облика $2^x - 1$ где је x природан број. Свака вештица има или две или ниједну директно подређену вештицу. Узмимо било коју вештицу v и посматрајмо низ $v = f_0, f_1, f_2, \cdots, f_k$ где је f_i надређена вештица од f_{i-1} за све $i \in 1, 2, \cdots, k$. Било који тако описан низ неће бити дужи од $log_2(n+1)$. Другим речима, биће дато комплетно бинарно балансирано стабло, тј. сви чворови на нивоима мањим од x ће имати тачно два сина, а чворови на последњем нивоу немају ниједног сина.
- **подзадатак 4 [30 поена]**: Нема додатних ограничења.

Назив проблема: Подземне воде

Временско ограничење:	0.5 s
Меморијско ограничење:	64 MB

У средњевековном селу Ењоред, налази се n срењовековних плацева поређаних у ред, један до другог, нумерисаних средњевековним бројевима од 1 до n. За сваки плац је позната снага утицаја средњевековних подземних вода над њим; за i-ти плац та вредност је v_i. Једног дана, скупило се k средњевековних вазалних господара и одлучили су да изграде k великих кућа при чему би свака кућа заузимала **тачно t узастопних плацева** и, наравно, **никоје две куће не могу имати заједнички плац**.

У договору са селом, вазални господари су се обавезали да, на име средњевековног пореза, укупно плате по 1 златник за сваки од n плацева који нису део ниједне куће. Испоставило се да су вредности n, k и t такве да када k вазалних господара подједнако поделе порез, **свако од њих треба да плати не више од 10 златника**.

Наравно, најбитнија ствар у овој средњевековној причи је одбрана од средњевековних вештица. У то време, када би људи градили куће, они би постављали анти-вештичне уређаје у **најлевљем плацу** своје куће и отпорност куће на вештице би директно зависила од одговарајуће подземне воде. Нпр. уколико се кућа налази на плацевима $i, i+1, \ldots i+t-1$, отпорност те куће је v_i. Вазални господари су разумни људи који верују у вештице и **договорили су се да сагради куће тако да је сума отпорности кућа на вештице највећа могућа**. Помозите им у томе!

Улаз. У првом реду стандардног улаза налазе се три природна броја n, k и t који представљају, редом, број плацева, број кућа које треба изградити и дужину сваке куће у плацевима. У наредном реду налазе се n природних бројева v_i, где i-ти број представља снагу утицаја подземне воде на i-ти плац.

Излаз. У првом и једином реду стандардног излаза исписати највећу могућу вредност суме отпорности свих кућа. **Гарантује се да ће решење (тј. бар један распоред кућа) увек постојати.**

Пример 1.

Улаз	Излаз
`8 3 2` `4 5 1 4 8 10 7 3`	`20`

Објашњење. Имамо 8 плацева и потребно је изградити 3 куће од којих свака заузима по 2 узастопна плаца. Уколико куће изградимо на следећи начин: 4 $\boxed{5\ 1}$ 4 $\boxed{8\ 10}$ $\boxed{7\ 3}$, тада је укупна отпорност кућа једнака $v_2 + v_5 + v_7 = 20$. Може се показати да не постоји распоред кућа који даје отпорност већу од 20.

Ограничења.

- $n \leq 800\ 000$
- $1 \leq t, k \leq 3\ 000$
- $1 \leq v_i \leq 10^9$, за свако $i = \overline{1, n}$
- За бројеве n, k и t важе сва додатна огранчења из текста задатака.

Подзадаци и бодовање. Тест примери су подељени у 4 подзадатка, у којима важе следећа додатна ограничења:

- **ПОДЗАДАТАК 1 [11 ПОЕНА]**: $n \leq 20$, $k, t \leq 5$
- **ПОДЗАДАТАК 2 [12 ПОЕНА]**: После изградње, остаће тачно један слободан плац
- **ПОДЗАДАТАК 3 [30 ПОЕНА]**: $n \leq 30\ 000$
- **ПОДЗАДАТАК 4 [47 ПОЕНА]**: Нема додатних ограничења.

Напомена. Обратити пажњу да је за испис решења неопходан 64-битни тип података.