

**SEMINARSKA NALOGA**

PROGRAMIRANJE 1

PRAKTIČNA MATEMATIKA

**Avtor naloge:**

Aljaž Marn

Ljubljana, 2024/25

|  |  |  |  |
| --- | --- | --- | --- |
|  | Ime naloge | Aktivna povezava | Število težavnostnih točk |
| 1 del |  |  |  |
|  | 4 toughts | <https://open.kattis.com/problems/4thought> | 3.0T |
|  | A prize no one can win | <https://open.kattis.com/problems/aprizenoonecanwin> | 3.0T |
|  | Adding words | <https://open.kattis.com/problems/addingwords> | 4.1T |

# Izjava

*Aljaž Marn izjavljam, da sem seminarsko nalogo opravil samostojno in da sem njen avtor. To pomeni, da sem v poročilu točno označil vse tiste dele kode, ki sem jih povzel iz virov in vire navedla. Prav tako vem, da izjava o avtorstvu pomeni, da znam razložiti vsako podrobnost kode, ki jo oddajam. Zavedam se, da v primeru, če izjava prvega stavka ni resnična, kršim disciplinska pravila*

**POROČILO 1. del**

# 4 toughts

Naloga je na spletni strani <https://open.kattis.com/problems/4thought> in je vredna 3.0 točke.

## BESEDILO NALOGE

Napišite program, ki bo ob podanem celoštevilskem vhodu n ustvaril matematični izraz, katerega rešitev je n. Rešitev mora uporabiti natanko štiri števke 4 in natanko tri binarne operacije, izbrane iz množice {∗,+,−,/}. Število 4 je EDINO število, ki ga lahko uporabite. Ni dovoljeno združevati števk 4 za ustvarjanje drugih števil, kot sta 44 ali 444.

Na primer, za n=0 je rešitev 4∗4−4∗4=0. Za n=7 je rešitev 4+4−4/4=7. Deljenje se obravnava kot celoštevilsko deljenje z okrajšavo, tako da je 1/4 enako 0 (namesto 0,25). Upoštevajte običajno prednost operacij, tako da je 4+4∗4=20, ne 32. Nimajo vsi celoštevilski vhodi rešitve z uporabo štirih števk 4 z omenjenimi omejitvami (razmislite o n=11).

**Vhodni podatki:**

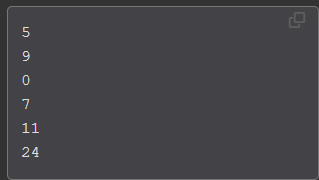
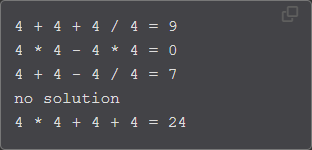
V prvi vrstici je podano celoštevilsko število m. To število je med 1 in 1000. V naslednjih m vrstic, pa je podano celoštevilsko število n. To število pa je med -1000000 in 1000000.

**Izhodni podatki:**

Program izpiše m vrstic. V vsaki je račun, katere rešitev je n. Izhod mora biti formatiran takole (naj znak $ označuje, katero koli izmed dovoljenih operacij):

4 $ 4 $ 4 $ 4 = n

Če tak račun ne obstaja naj program izpiše: no solution.

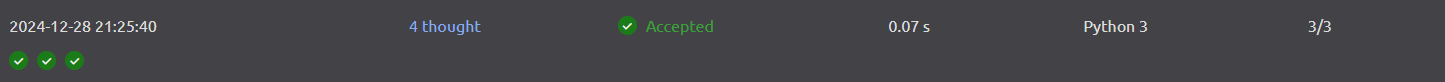
Primer vhodnih podatkov: Primer izhodnih podatkov:

## OPIS PROBLEMA IN IDEJA REŠEVANJA

Prva vrstica vhodnih podatkov nam pove količino testnih primerov. Preostanek pa so rešitve računa katerega moramo sestaviti.

Preberimo najprej prvo vrstico. Potem prebiramo vrstice eno po eno in sproti izpisujemo račune. Pregledujemo vse možne kombinacije operatorjev. Vsak izraz tudi izračunamo in če je izračun enak vhodu, ga izpišemo. Pri računanju moramo paziti na pravilni vrstni red operatorjev, zato najprej izračunamo operatorje, ki imajo »prednost«, tj. najprej množimo in delimo, potem pa seštevamo in odštevamo.

## DOKAZ, DA JE NALOGA REŠENA



# A prize no one can win

Naloga je na spletni strani <https://open.kattis.com/problems/aprizenoonecanwin> in je vredna 3 točke.



## BESEDILO NALOGE

Po prazničnem odprtju vaše nove trgovine, Butična trgovina za alternativno paramedicino in Cwakhsahlvereigh, z razočaranjem ugotovite, da ne dosegate toliko prodaje, kot ste upali. Da bi to popravili, se odločite za posebno ponudbo: označili boste določeno podmnožico od n izdelkov v vaši trgovini kot sodelujoče v ponudbi, in če ljudje kupijo natančno dva od teh izdelkov, pri čemer je strošek teh izdelkov \*strogo\* večji od X evrov, jim boste podarili brezplačen samorogov rog!

Ker ste pred kratkim ugotovili, da so vsi vaši samorogovi rogovi pravzaprav narvaljevi okli, se odločite prelisičiti ponudbo tako, da boste sodelujoče izdelke izbrali na način, da nihče tako ali tako ne bo mogel zaslužiti roga.

Da ne bi nihče postal sumničav, želite označiti čim več izdelkov kot sodelujočih v ponudbi.

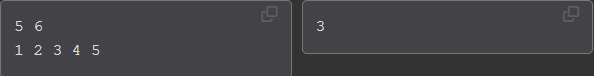
Vhodni podatki:

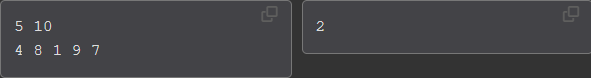
V prvi vrstici, sta dve števili. Prva označuje število izdelkov (pozitivno število manjše od 100000), druga pa ceno, ki jo mora kupec preseči, da dobi samorogov rog (pozitivno število manjše od bilijona).

V naslednji vrstici pa so cene izdelkov. To so pozitivna števila, ki so manjša od bilijona.

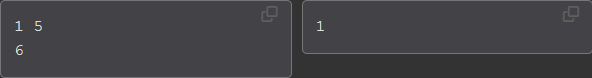
Izhodni podatki:

Število izdelkov, ki jih lahko označimo, da sodelujejo v ponudbi, s tem, da nihče ne more dobiti roga.

Primer vhodnih podatkov 1: Primer izhodnih podatkov 1: 

Primer vhodnih podatkov 2: Primer izhodnih podatkov 2:

Primer vhodnih podatkov 3: Primer izhodnih podatkov 3:

Primer vhodnih podatkov 4: Primer izhodnih podatkov 4: 

## OPIS PROBLEMA IN IDEJA REŠEVANJA

Vhodne podatke sestavljata dve vrstici. V prvi vrstici sta dve število, število izdelkov in cena, ki jo mora kupec preseči. V drugi vrstici pa so cene izdelkov, ki se prodajajo v trgovini.

Preberemo število n in izdelkov in ceno x. Potem preberemo cene izdelkov in jih za lažje obravnavanje uredimo po velikosti. Potem pogledamo dva zaporedna elementa in ju seštejemo. Če je seštevek manjši od x, vemo, da ta element lahko označimo za ponudbo.

## DOKAZ, DA JE NALOGA REŠENA



# Adding Words

Naloga je na spletni strani <https://open.kattis.com/problems/addingwords> in je vredna 4.1 točke.



## BESEDILO NALOGE

Psihologi na Univerzi Wassamatta verjamejo, da ljudje lažje upravljajo z besedami kot s številkami. Zato so zasnovali eksperimente, da bi ugotovili, ali to drži. V zanimivem preobratu se eden od njihovih eksperimentov ukvarja z uporabo besed, ki predstavljajo številke. Namesto seštevanja številk želijo seštevati besede. Vi ste raziskovalni programer na projektu in vaša naloga je napisati program, ki prikazuje to sposobnost.

Vhodni podatki:  
Vhod je zaporedje do 2000 ukazov. Vsak ukaz je v svoji vrstici. Ukazi se zaključijo, ko se prebere celotna datoteka. Vsak ukaz je ali definicija ali račun ali pa izbris. Vsi elementi znotraj ukaza so ločeni z enojnim presledkom.

Definicija ima obliko *def x y,* kjer je x ime spremenljivke, y pa celo število, ki je večje od -1000 in manjše od 1000. Dve spremenljivki, ne moreta biti hkrati definirani z isto številčno vrednostjo. Če je bila x že prej definirana, nova definicija izbriše njeno staro definicijo. Imena spremenljivk so lahko dolga do 30 malih črk, vsaka črka med a in z.

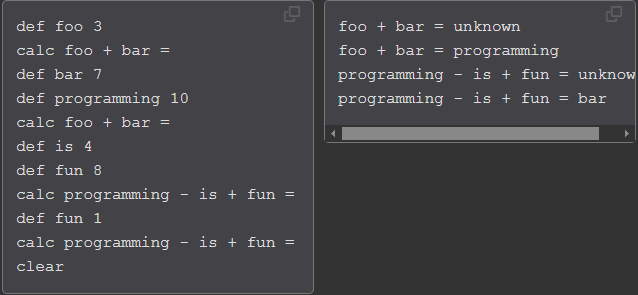
Ukaz za račun se začne z besedo *calc* sledijo pa ji ena do 15 spremenljivk, ločenih z operatorji za seštevanje in odštevanje. Konec ukaza, je enačaj. Na primer: calc foo + bar – car +

Ukaz *clear* naroči programu naj pozabi vse svoje definicije.

Izhodni podatki:

Vaš program ne sme proizvesti nobenega izpisa za definicije, za izračune pa mora proizvesti vrednost izračuna. Če za rezultat ne obstaja beseda ali če katera od besed v izračunu ni bila definirana, mora biti rezultat izračuna »unknown«. Beseda »unknown« se nikoli ne uporablja kot spremenljivka v vhodu.

Primer vhodnih podatkov: Primer izhodnih podatkov:



## OPIS PROBLEMA IN IDEJA REŠEVANJA

Vhodni podatek je vrstica, ki lahko ali definira novo spremenljivko, računa, ali pa pobriše vse obstoječe definicije. Vrstice je treba brati dokler ne pridemo do konca datoteke.

Če je prva beseda v ukazu »*def*«, si ime spremenljivke in njeno vrednost nekam zapomnimo.

Če je prva beseda v ukazu »*calc*«, vse besede po njihovih definicijah pretvorimo v številke, če kakšna beseda ni definirano izpišemo ukaz brez »*calc*« in na koncu še »*unknown*«. Te številke potem shranjujemo v seznam. Če moramo številko odšteti jo pomnožimo z -1. Na koncu potem to seštejemo in pogledamo ali kakšna izmed spremenljivk ustreza našemu rezultatu, če ne izpišemo ukaz brez »*calc*« in na koncu še »*unknown*«. Če pa rezultat ustreza spremenljivki, izpišemo ukaz brez »*calc*« ter to spremenljivko.

Če je prva in edina beseda v vrstici »*clear«* izbrišemo vse definicije.

## DOKAZ, DA JE NALOGA REŠENA

