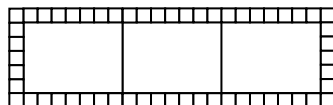


Општинско такмичење програмирања за ученике основних школа

27. фебруар 2010

II категорија (7. и 8. разред)

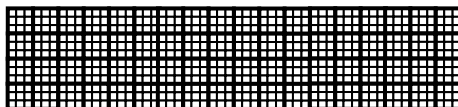
- Кнез Љубинко је решио да у свом родном месту направи мост преко реке. Сакупио је најбоље мајсторе који су почели изградњу каменог моста. У каменоломима су секли огромне камене плоче једнаких димензија које су постављали на стубове и то им је представљало основу моста. Ширина плоче је увек одређивала ширину моста. Да се ивице ових плоча не би временом оштетиле, након што поређају плоче у низ они су око ових великих плоча постављали мање плоче квадратне основе димензије 1 метар. Написати програм **MOST** у коме се уносе ширина плоче **S** и дужина плоче **D**, задате у метрима и број плоча **N** који је потребан да се направи основа једног моста, а затим се израчунава број мањих квадратних плоча које су потребне да се поставе по ивицама моста. Вредности променљивих **S**, **D** и **N** су позитивни цели бројеви, $0 < S, D, N \leq 100$.



Пример. Улаз: $S = 5$ $D = 7$ $N = 3$

Излаз: 56

- У кнежевини Кнеза Љубинка при прављењу пута, када се направи камена основа, користе се дрвене коцке које се ређају преко камена да би камена основа што дуже трајала. Након неколико година дрвене коцке пропадају и потребно их је заменити. Радници су од дрвета правили коцке само у две величине и пут су прекривали само једном врстом од ове две. Које ће коцке да користе за неки део пута зависило је од тога које коцке се боље уклапају, тј. којом врстом ће бити покривен пут, а да при томе нема празнина ни вишка. Ако са обе врсте могу да поплучају на одговарајући начин, бирају оне коцке којих им треба у мањем броју. Написати програм **КОЦКЕ** којим се уносе ширина **S** и дужина **D** пута, задате у метрима, које треба прекрити дрвеним коцкама, а затим се уносе дужине ивица коцки **K1** и **K2**. Програм треба да испише број дрвених коцки које ће бити искоришћене за покривање пута, ако се поштују правила, или да испише коментар **NE MOZE**, ако ни једном врстом коцки прекривање није могуће. Вредности променљивих **S**, **D**, **K1** и **K2** су позитивни цели бројеви, $0 < S, D, K1, K2 \leq 100$.



Пример. Улаз: $S = 12$ $D = 54$ $K1 = 2$ $K2 = 3$

Излаз: 72

Улаз: $S = 12$ $D = 55$ $K1 = 5$ $K2 = 4$

Излаз: NE MOZE

- На свим великим градилиштима у кнежевини кнеза Љубинка увек се организује исхрана радника. Кнез Љубинко тај посао повери неком великом газди из оближњег места. У зависности од броја радника, дужине посла и посете порезника, посао исхране радника на градилишту изабраном газди некада донесе добит, а некада трошак. Принцип организовања исхране је увек исти: Газда је у обавези да на одређеном месту на почетку постави **P** столова, сваки сто му за **K** дана донесе зараду од 1 златника и тих **P** златника се смешта у посебан сеф. Ако у неком тренутку у сефу има најмање **S** златника, газда даје **S** златника из сефа, а дотадашњи број столова увећа 2 пута. Порезници долазе на градилиште сваких **M** дана и тада за сваки сто наплаћују порез од 2 златника. Порез се плаћа из сефа у коме се чува зарада. Уколико у сефу нема довољно златника, газда порез плаћа од свог новца, што рачуна као негативну вредност у сефу, а када стигне нова зарада он прво надокнади свој новац остављајући евентуални остатак у сеф. Порезници увек стижу касно увече, тј. након што је газда, евентуално, додао зараду и удвостручио број столова, ако је имао довољно златника, и порез се плаћа за затечену ситуацију. Након завршетка радова који трају **D** дана газда своди рачун о својој заради. Написати програм **SEF** који за унете вредности **P**, **K**, **S**, **M** и **D** одређује зараду коју је газда остварио на овом градилишту. Вредности променљивих **P**, **K**, **S**, **M** и **D** су позитивни цели бројеви, а вредности за број столова и укупну зараду ће у сваком тренутку бити мањи или једнаки 2 000 000 000.

Пример. Улаз: $P = 2$ $K = 3$ $S = 10$ $M = 10$ $D = 15$

Излаз: 6

Улаз: $P = 2$ $K = 3$ $S = 10$ $M = 5$ $D = 15$

Излаз: -2

Израда задатака траје 120 минута

- задатак – 30 поена
- задатак – 35 поена
- задатак – 35 поена

Општинско такмичење из програмирања за ученике основних школа

12. март 2011

II категорија (7. и 8. разред)

1. Ресторан „36“ је због добре хране и одличне забаве увек пун гостију, због чега се запослени у њему труде да направе што више места за седење. Ресторан располаже столовима за шест, четири и две особе, од којих прва врста заузима површину од $7m^2$, друга $5m^2$, а најмања $2m^2$. Поред столова у ресторану се мора наћи место и за бенд који забавља госте и који сваке вечери наступа са различитим бројем извођача, при чему је по извођачу потребно обезбедити $1m^2$. Обзиром да број извођача варира, столови се сваке вечери изнова распоређују.

Зна се да:

- столова за 4 и 6 особа има толико да се сигурно сви могу увек распоредити и они се први постављају
- простор који бенд заузима никада није већи од простора који остаје слободан након што се поставе столови за 4 и 6 особа
- након распоређивања столова за 4 и 6 особа и бенда, преостали простор се попуњава столовима за две особе.

Написати програм **STOLOVI** који након уноса површине ресторана, броја столова за 4 и 6 особа и броја извођача у бенду одређује колико столова за две особе може да се постави.

Улазни подаци. Стандардни улаз садржи четири линије од којих свака садржи по један позитиван цео број и то следећим редим: прва линија – **R** површина ресторана, друга линија – **S1** број столова за четири особе, трећа линија – **S2** број столова за шест особа, четврта линија – **B** број извођача у бенду.

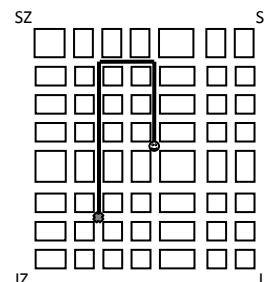
Изразни подаци. Једина линија стандардног излаза садржи ненегативан цео број који представља број столова за две особе који се може поставити.

Пример.

Улаз:
58
2
4
5

Израз:
7

2. Сима је први пут у граду, па је у своју дневну руту за обилазак уписао и посету ресторану „36“. Купио је карту града и успео на њој да одреди на којој позицији се тренутно налази. Уочио је да су сваке две улице или међусобно паралелне или нормалне, да се свака улица протеже у једном од два правца, север-југ и исток-запад, као и да се он налази на раскрсници. Како је карта била мало старија, у њу није био уписан препоручени ресторан, па је Сима замолио пролазнике да му објасне како да стигне до ресторана „36“. Испоставило се да се и сам ресторан налази на раскрсници. Упутство које је добио је било у облику: идеш на север и на трећој раскрсници скренеш на запад, затим на другој раскрсници иди на југ и на петој раскрсници се налази ресторан. Дакле, до ресторана је имао 2 скретања, при чему је за свако сретање дато упутство на коју страну ићи улицом и до које раскрснице. Написати програм **KARTA** који након унетих података из упутства, одређује где се ресторан налази у односу на тренутну позицију Симе, при чему су могући следећи излази:



- **SZ** ако се ресторан налази северо-западно,
- **SI** ако се ресторан налази северо-источно,
- **JZ** ако се ресторан налази југо-западно,
- **JI** ако се ресторан налази југо-источно,
- **ISTA** ако се ресторан налази у истој улици у којој се налази и Сима (имати у виду да се он налази на раскрсници).

Улазни подаци. Стандардни улаз садржи шест линија и свака линија садржи позитиван цео број и то следећим редом: P1, R1, P2, R2, P3, R3. Вредности P1, P2, P3 представљају смер у коме треба Сима да се креће и могу имати вредности од 1 до 4 (1 – север, 2 – југ, 3 – исток, 4 – запад), а R1, R2, R3 представљају редни број раскрснице до које треба ићи.

Изразни подаци. Једина линија стандардног излаза садржи једну од следећих речи: **SZ, SI, JZ, JI, ISTA**.

Пример.

Улаз:
1 3
4 2
2 5

Израз:
JZ

3. Ресторан „36“ има свој паркинг на чијој рампи ради оператер. Рампа може бити отворена за улаз, у ком случају кроз рампу у минути може да прође 3 возила, или за излаз, када кроз рампу у минути пролазе 2 возила. Оператер поставља статус рампе „улаз“ или „излаз“ у зависности од ситуације, тј. у зависности од редова који се формирају. За све време трајања статуса „улаз“ на паркинг улазе возила, осим уколико је паркинг пун када не може да уђе ни једно возило. За све време трајања статуса „излаз“ са паркинга излазе возила, осим уколико је паркинг празан, па са паркинга не може да изађе ни једно возило. Оператер може више пута узастопно да постави исти статус. Написати програм **PARKING** у коме се за унети број возила која се могу паркирати на паркингу, број статуса које је оператер поставио од тренутка када је паркинг био празан до посматраног тренутка и трајање, у минутима, сваког постављеног статуса, одређује колико возила има на паркингу у том тренутку.

Улазни подаци. Прва линија стандардног улаза садржи позитиван цео број **P** који представља максималан број возила која се могу паркирати на паркингу. Друга линија стандардног улаза садржи позитиван цео број **N** који представља број постављених статуса. Наредних **N** линија стандардног улаза садрже по један цео број који представља трајање, у минутима, постављеног статуса, при чему уколико је број позитиван статус је „улаз“, а уколико је број негативан статус је „излаз“.

Излазни подаци. Једина линија стандардног излаза садржи ненегативан цео број који представља број возила на паркингу у посматраном тренутку.

Пример.

Улаз:	Излаз:
100	66
5	
15	
-6	
-20	
18	
4	

Израда задатака траје 120 минута

1. задатак – 30 поена
2. задатак – 35 поена
3. задатак – 35 поена