

# ПЪРВИ ТУРНИР ПО ПРОГРАМИРАНЕ ЗА КУПАТА НА ДЕКАНА НА ФМИ, СУ – София, 5.12.2004

## Задача А. Антитероризъм

В днешни дни тероризмът се превръща в световна заплаха и борбата с него става все по-важна. Но за ефективна борба е необходимо да се предприемат както превантивни действия така и да има сериозна информационна система, с помощта на която в тежки кризисни ситуации да се реагира бързо и точно.

Нека разгледаме една възможна ситуация. В град X са разположени N ( $1 < N \leq 1000$ ) взривни устройства, които антитерористичната група означила с числата от 1 до N. Известно е, че терористите подготвят сериозен удар и след кратко време ще са готови да взривят всичките устройства едновременно. Антитерористичната група обаче е добре подготвена и има пълна агентурна информация за плана на врага. А разполага и с добри специалисти по обезвреждане на бомби, както и с хеликоптер. Поради големите разстояния, както и заради натовареното движение в града, този хеликоптер се оказва единственото средство за придвижване и само с негова помощ трябва да се посетят последователно всички взривни устройства и да се обезвредят, като полетите от място с бомба до друго такова трябва да са по права линия. Терористите очакват подобно поведение и за целта са създали свои постове, от които да издебнат, нападнат и унищожат борците срещу тероризма. Антитерористите не могат да си позволят да прелетят над пост, нито пък да минава през едно и също място с взривно устройство повече от веднъж. За тяхно щастие е установена следната тактика на врага – постове винаги са разположени на пътя между две взривни устройства, един пост се грижи само за едната посока на пътя между две взривни устройства и нито един път не се пази и в двете посоки. Напишете програма, с която да помогнете за борбата срещу световната заплаха.

На първия ред на текстовия файл **a.inp** е записано едно число K – броя на градовете, които трябва да се спасят. Следват K описания на оперативната обстановка във всеки от градовете. Всяко описание започва с един ред съдържащ 2 числа – N и M, съответно броя на взривните устройства в града и броя на терористичните постове. Следват M реда, съдържащи по две числа I и J, които описват пост на терористите и означават че поста пази въздушния път от I към J (а пътят от J към I не се пази). Вашата задача е да определите в какъв ред да се обезвредят бомбите така, че антитерористичната група да не бъде нападната от вражески пост. За всеки град трябва да се напишат по 2 реда на стандартния изход – първия от тях е `City #X: ,` където X е номер на града започвайки от 1. Вторият ред трябва да се състои от N числа и да представлява последователността, в която да се обезвредят взривните устройства. Няма ограничения за това кое устройство да се обезвреди първо. Ако има няколко възможни отговора, трябва да отпечатате произволен верен отговор като приемате, че такъв винаги има. За разделител на числата и във входа и в изхода се използва един интервал.

Примерен вход:	Възможен отговор:
2	City #1:
5 7	2 1 5 4 3
1 2	City #2:
3 5	8 2 3 4 5 1 7 6 9
3 2	
3 4	
4 5	
4 2	
2 5	
9 1	
9 8	

# ПЪРВИ ТУРНИР ПО ПРОГРАМИРАНЕ ЗА КУПАТА НА ДЕКАНА НА ФМИ, СУ – София, 5.12.2004

## Задача В. Боулинг

Една игра на боулинг се състои от десет фрейма. Целта във всеки фрейм е да се хвърля топка за боулинг към 10 кегли и да се съборят колкото се може повече от тях. Кеглите са подредени в правилен триъгълник.

Играчът има право на най-много две хвърляния за всеки фрейм. Когато събори всички кегли от първото хвърляне, фреймът се отчита като страйк. Ако след първото хвърляне останат несъборени кегли, играчът има право на второ хвърляне за да събори оставащите кегли. Ако той успее да събори оставащите кегли с втория си опит, фреймът се отчита като спеър.

Резултатът от една игра е сумата от резултатите за отделните фреймове. Резултатът за фрейм е броят на съборените кегли + бонуси за страйк и спеър. Ако играчът отбележи страйк в някой от фреймовете, той печели 10 точки + броя на съборените кегли от следващите две хвърляния. Ако играчът отбележи спеър в някой от фреймовете, той печели 10 точки + броя на съборените кегли от следващото хвърляне. Играч отбелязал страйк в последния фрейм получава правото на две допълнителни хвърляния, а играч отбелязал спеър в последния фрейм получава правото на едно допълнително хвърляне. Съборените в допълнителните хвърляния кегли се броят само като бонус за предходния страйк или спеър. Така максималният възможен резултат за една игра (когато играчът отбележи страйк във всички фреймове + 2 допълнителни хвърляния за десетия фрейм) е 300.

Входните данни се четат от текстов файл **b.inp**, състоящ се от поредица от резултатите за няколко игри. На всеки ред е записан резултата от една боулинг игра, като резултатите за всяко хвърляне са разделени с един интервал. Резултатът от всяко хвърляне е представен с един знак – цифра отбелязваща броя на съборените кегли, '/' – за спеър или 'X' – за страйк. Краят на входните данни се отбелязва с `Game Over` на отделен ред.

Програмата трябва да изведе на отделен ред на стандартния изход резултата за всяка от боулинг игрите от входния файл. Редът на резултатите трябва да отговаря на редът на игрите във входния файл.

Примерен вход:	Отговор:
1 0 1 4 6 1 2 3 3 1 1 2 4 5 X 3 2 1 8	63
1 / 2 1 1 5 1 4 3 6 1 3 1 5 1 1 1 / 1 / 1	69
3 1 4 5 6 1 2 2 9 0 1 1 4 3 5 1 1 6 7 2	64
X X X X X X X X X X X X	300
Game Over	

# ПЪРВИ ТУРНИР ПО ПРОГРАМИРАНЕ ЗА КУПАТА НА ДЕКАНА НА ФМИ, СУ – София, 5.12.2004

## Задача С. Кръг

Рим е древно селище. Неотдавна археолозите откриха в централната част на града останките на римски стадион. Градските власти искат да построят привлекателен за туристите кръгъл площад около стадиона, но са ограничени от съществуващите сгради и улици. Улиците и лицата на сградите образуват затворен изпъкнал полигон с  $N$  върха, във вътрешността на който трябва да бъде построен площадът. Той може да се допира до страните на полигона, но не трябва да ги пресича. Градските власти биха искали да знаят колко голям може да се получи кръглия площад, ако центърът му е зададена точка във вътрешността на полигона.

Напишете програма, която по зададено  $N$  ( $3 \leq N \leq 2000$ ), изпъкнал затворен полигон с  $N$  върха и точка, избрана за център, намира максималния радиус на кръг, който лежи изцяло във вътрешността на полигона. Всички точки се определят с целочислени координати в правоъгълна координатна система.

На първия ред на текстовия файл **c.inp** е записан броят на тестовите примери. За всеки тестов пример върховете  $N$  и координатите  $X$ ,  $Y$  на центъра ще бъдат на първия ред, разделени с по един интервал. Всеки от следващите  $N$  реда ще съдържа координатите на един от върховете на полигона. Върховете са зададени в реда по които се срещат в полигона при обхождането му (в посока на движението на часовниковата стрелка или в обратната посока). Всички координати са цели числа в интервала  $[-1000, 1000]$ .

Програмата трябва да изведе на стандартния изход максималният възможен радиус за всеки тестов пример. Всеки радиус, число с шест цифри след десетичната точка, трябва да е записан на отделен ред. Редовете трябва да следват наредбата на тестовите примери от входния файл.

Примерен вход:	Отговор:
2	5.000000
4 5 5	1.341641
0 0	
0 10	
10 10	
10 0	
5 0 3	
-10 0	
0 0	
5 10	
5 15	
-1 10	

# ПЪРВИ ТУРНИР ПО ПРОГРАМИРАНЕ ЗА КУПАТА НА ДЕКАНА НА ФМИ, СУ – София, 5.12.2004

## Задача D. Директории

Станчо иска да разработи нова операционна система – Stancho DOS. За нещастие, той не е много добър системен програмист. Затова има нужда от вас, да му напишете програма-команден интерпретатор, която да обработва системните команди. Тъй като Станчо е далеч от поддръжка на дискови устройства, операционната система ще държи файловете в паметта. По този начин потребителите няма да могат да пазят информация след изключване на компютъра, но едва ли в днешно време някой има нужда да изключва компютъра си.

Както повечето файлови системи, и тази има дървовидна структура. Нелистата са директориите (каталозите), а листата - файловете. Всеки файл или директория има *име* – низ от знаци с дължина не надхвърляща 100, различни от бленк (интервал, backspace, tab и т.н.) и '/'. Името не може да бъде '..'. Примерни имена са: stancho.sys, StanCho.Sys) (\*)&)^;Psdlkjfadslkj sdf и fish. *Пътят към файл* представлява низ започващ с '/' и евентуално последван от няколко имена разделени с '/'. Както в повечето операционни системи пътят указва еднозначно къде се намира файлът в дървовидната файлова система, като имената разделени с '/' са имената на директориите по пътя. Примерни пътища са: /sys/stancho.sys и /stancho.sys.

Коренът на дървото е директория, която наричаме *главна* директория и означаваме с '/'. Във всеки момент една от директориите е *текуща*, и в нея се извършват операциите. При стартирането на системата главната директория е текуща и съдържа единствен файл с име stancho.sys. Командният интерпретатор трябва да поддържа следните команди:

dir

показва файловете и директориите в текущата директория, сортирани лексикографски.

copy <path>

копира файлът указан с пътя <path> в текущата директория. За да успее командата, трябва указаният файл да съществува и да няма файл със същото име в текущата директория. При успех отпечатва ok, а при неуспех – error в отделен ред на стандартния изход.

ren <from\_filename> <to\_filename>

променя името на елемент (файл или директория) от текущата директория. Командата успява, когато <from\_filename> е име на съществуващ елемент от текущата директория и в нея не съществува елемент с име <to\_filename>. При успех отпечатва ok, а при неуспех – error в отделен ред на стандартния изход.

del <filename>

изтрива елемент от текущата директория. Успява, когато елемент с име <filename> съществува в текущата директория. При успех отпечатва ok, а при неуспех – error в отделен ред на стандартния изход.

cd <dirname>

поддиректорията <dirname> на текущата става нова текуща. Командата успява, когато съществува поддиректория с име <dirname> в текущата директория. При успех отпечатва ok, а при неуспех error в отделен ред на стандартния изход.

cd ..

директорията, която съдържа текущата, става нова текуща. Командата успява, когато

# ПЪРВИ ТУРНИР ПО ПРОГРАМИРАНЕ ЗА КУПАТА НА ДЕКАНА НА ФМИ, СУ – София, 5.12.2004

текущата директория не е '/'. При успех отпечатва `ok`, а при неуспех – `error` в отделен ред на стандартния изход.

`cd /`  
директорията '/' става нова текуща. Винаги успява и изпечатва `ok`.

`md <dirname>`  
Създава празна директория с име `<dirname>` в текущата директория. Успява, когато не съществува елемент с име `<dirname>`

Входните данни се четат от текстов файл с име **d.inp**. Той се състои от много редове с по една команда на тях. Всички команди както и аргументите им са синтактично коректни.

Вашата програма трябва да изведе на стандартният изход резултата от изпълнение на командите.

Примерен вход:	Примерен изход:
<code>dir</code>	<code>stancho.sys</code>
<code>md Fish</code>	<code>ok</code>
<code>md fish</code>	<code>ok</code>
<code>cd Fish</code>	<code>ok</code>
<code>copy /stancho.sys</code>	<code>ok</code>
<code>dir</code>	<code>stancho.sys</code>
<code>cd ..</code>	<code>ok</code>
<code>cd fish</code>	<code>ok</code>
<code>copy /Stancho.sys</code>	<code>error</code>
<code>copy /pancho.sys</code>	<code>error</code>
<code>copy /stancho.sys</code>	<code>ok</code>
<code>dir</code>	<code>stancho.sys</code>
<code>ren stancho.sys pancho.sys</code>	<code>ok</code>
<code>cd /</code>	<code>ok</code>
<code>copy /Fish/pancho.sys</code>	<code>error</code>
<code>copy /fish/pancho.sys</code>	<code>ok</code>
<code>dir</code>	<code>Fish</code>
	<code>fish</code>
	<code>pancho.sys</code>
	<code>stancho.sys</code>

# ПЪРВИ ТУРНИР ПО ПРОГРАМИРАНЕ ЗА КУПАТА НА ДЕКАНА НА ФМИ, СУ – София, 5.12.2004

## Задача Е. Класиране

В състезание участват  $N$  играча. След няколко игри всеки от състезателите печели определен брой точки. В класирането играчите се подреждат в намаляващ ред на точките спечелени от всеки един и от значение е единствено заетото място. Състезателите с равен брой точки заемат едно и също място в класирането. Напишете програма, която по зададено  $N$  намира броя на различните класирания които могат да се получат.

Входният текстов файл **e.inp** съдържа няколко тестови примера. Всеки тестов пример се състои от един ред, на който е записано число  $N$  ( $N < 19$ ) – броят на състезателите.

За всеки тестов пример изведете на стандартния изход по един ред, на който е записан броят на различните класирания. Редовете трябва да следват наредбата на тестовите примери от входния файл.

Примерен вход:	Отговор:
2	3
3	13

# ПЪРВИ ТУРНИР ПО ПРОГРАМИРАНЕ ЗА КУПАТА НА ДЕКАНА НА ФМИ, СУ – София, 5.12.2004

## Задача F. Факториел

Всеки от Вас (би трябвало да) знае какво е факториел и как се изчислява. За всяко положително естествено число  $N$  факториелът на числото е  $N! = N * (N-1) * (N-2) * \dots * 1$ . По дефиниция  $0! = 1$ . Напишете програма, която по зададено  $N$  намира броя на цифрите на  $N!$ .

Входният текстов файл **f.inp** съдържа няколко тестови примера. Всеки тестов пример се състои от един ред, на който е записано цяло неотрицателно число  $N$  ( $N < 200\ 000$ ).

За всеки тестов пример изведете на стандартния изход по един ред, на който е записан броят на цифрите в десетичното представяне на  $N!$ . Редовете трябва да следват наредбата на тестовите примери от входния файл.

Примерен вход:	Отговор:
2	1
5	3
8	5