

NI1. SPIONI

80 p

Elevii unei școli de spionaj descoperă pe parcursul unei misiuni de pregătire o serie de documente secrete codificate. La o prima analiză a acestora, elevii observă că toate documentele sunt formate din numere de aceeași lungime (cu același număr de cifre). Într-o etapă ulterioară a analizei documentelor ei identifică o anumită cifră imprimată întotdeauna diferit față de celelalte. Presupunerea lor logică este aceea că doar numerele care conțin acea cifră au o anumită semnificație pentru interpretarea finală. După ce extrag numerele respective, sesizează că o anumită cifră nu se găsește în nici unul din aceste numere. La pasul următor, elevii trebuie să afle câte astfel de numere posibile există, pentru a aprecia corect puterea de calcul necesară interpretării aceluia cod.

Cerință:

Scrieți un program care citește dintr-un fișier de intrare numărul n și cifrele a și b , și determină numărul de numere de n cifre care conțin cifra a și care nu conțin cifra b .

Date de intrare:

Fișierul de intrare **spioni.in** conține pe o singură linie numărul n de cifre precum și cifrele a și b , separate între ele prin câte un spațiu.

Date de ieșire:

Fișierul de ieșire **spioni.out** conține numărul de numere de n cifre care conțin cifra a și care nu conțin cifra b .

Restricții:

$1 < n \leq 1000$

a, b – cifre distincte în sistemul zecimal

Exemplu:

| spioni.in | spioni.out | Explicație |
|-----------|------------|---|
| 2 3 4 | 16 | Există 16 numere de 2 cifre care conțin cifra 3 și nu conțin cifra 4: 13, 23, 30, 31, 32, 33, 35, 36, 37, 38, 39, 53, 63, 73, 83, 93 |

Timp maxim de execuție/test: 1 secundă.

GC1. Maximizare

10 P

Se dau mulțimile $A = \{a_1, a_2, \dots, a_m\}$ și $B = \{b_1, b_2, \dots, b_n\}$ având elemente numere întregi nenule. Se știe că $n \geq m$. Să se determine o submulțime $B' = \{x_1, x_2, \dots, x_m\}$ a lui B , astfel încât valoarea expresiei $E = a_1 * x_1 + a_2 * x_2 + \dots + a_m * x_m$ să fie maximă și să se calculeze expresia E .

Date de intrare

Prima linie a fișierului **MAXIM.IN** conține numerele m și n , despărțite printr-un spațiu. A doua și a treia linie conțin elementele numere întregi ale mulțimii A , respectiv B . Numerele sunt separate prin câte un spațiu.

Date de ieșire

Fișierul **MAXIM.OUT** va conține o linie pe care se va afla un număr întreg, care reprezintă valoarea expresiei E .

Restricții și precizări

$1 \leq m, n \leq 100$

Exemplu

| MAXIM.IN | MAXIM.OUT |
|-----------------------------|-----------|
| 3 5 2 4 3 5 -3 8 -1 2 | 51 |

GC2. RĂZBOI

30 P

O căpetenie s-a infiltrat ca spion în armata romană pentru a decide în ce zi să declanșeze un atac decisiv. Căpetenia trebuie să-i transmită lui Decebal un cuvânt cheie ce reprezintă ziua respectivă. Mesajul e transmis codificat, printr-o propoziție inofensivă.

Algoritmul de codificare este:

- fiecare cuvânt se scrie pe verticală.
- cuvintele sunt formate doar din litere mari
- cuvintele vor fi scrise succesiv de la stânga la dreapta

- tot ce înseamnă separatori se ignoră.

Decodificarea, alegerea cuvântului cheie:

- fiecare secvență în care pe o anumită linie a „tabelului de litere” construit, o literă apare de două sau mai multe ori consecutiv pe coloane diferite, identifică o literă a cuvântului cheie.
- ordinea de extragere a literelor cheie: pe fiecare linie de la stânga la dreapta, iar pe linii de sus în jos.

Date de intrare RAZBOI.IN

- fraza de lungime maximă 255 caractere aflată pe prima linie

Date de ieșire RAZBOI.OUT

- cuvântul cheie situat pe prima linie a fișierului

| | | | | | | | | | | |
|-------------------|--|---|---|---|---|---|---|---|---|---|
| RĂZBOI.IN | M-AM MIRAT CIND AM AFLAT, ASA DIN SENIN, TAINE GRELE | | | | | | | | | |
| | M | M | C | A | A | A | D | S | T | G |
| | A | I | I | M | F | S | I | E | A | R |
| | M | R | N | | L | A | N | N | I | E |
| | | A | D | | A | | | I | N | L |
| | | T | | | T | | | N | E | E |
| RĂZBOI.OUT | MAINE | | | | | | | | | |

GC3. 3PRIME

40P

Fiind dat un număr natural n de cel puțin două cifre, realizați toate descompunerile sale în sumă de 3 numere prime distincte.

Date de intrare 3PRIM.IN

- un număr natural $n \in [10, 1000000]$

Date de ieșire 3PRIM.OUT

- toate descompunerile în ordine crescătoare, fiecare descompunere fiind pe o linie nouă

| 3PRIM.IN | 3PRIM.OUT |
|----------|------------------|
| 10 | 2 3 5 |
| 25 | 3 5 17 5 7 13 |

GC4. Insule

100 P

Pe o hartă s-au găsit mai multe insule. Acestea sunt numerotate de la 1 la n . Insulele sunt legate între ele prin poduri. Determinați care sunt grupurile care au cele mai multe insule legate între ele și matricea de adiacență a ultimei comunități.

Date de intrare

Prima linie a fișierului de intrare **INSULE.IN** conține numerele n și m , reprezentând numărul de insule și numărul de poduri existente, iar pe următoarele m linii se află câte 2 numere întregi (x și y), separate prin câte un spațiu; aceste numere ne arată existența unui pod între insula x și insula y .

Date de ieșire

Fișierul de ieșire **INSULE.OUT** va conține detalierea tuturor comunităților cu numărul cel mai mare de insule, matricea de adiacență a ultimei comunități

Restricții și precizări

- $1 \leq n \leq 30$
- $1 \leq m \leq 1000$
- În cazul în care sunt mai multe comunități cu același număr de insule se vor afișa toate.
- Insulele se vor afișa în cadrul comunității în ordinea DF (parcursarea în adâncime a grafului pornind de la insula cea mai mică), iar comunitățile se ordonează după prima insulă din fiecare comunitate, la matricea de adiacență se ordonează insulele crescător.

| INSULE.IN | INSULE.OUT |
|-----------|-----------------------------------|
| 9 9 | comunitatea 1 are insulele: 1 8 9 |
| 7 5 | comunitatea 2 are insulele: 2 4 6 |
| 5 3 | comunitatea 3 are insulele: 3 5 7 |
| 4 2 | matricea de adiacenta: |

| | |
|-----|---------|
| 4 6 | # 3 5 7 |
| 2 6 | 3 0 1 1 |
| 1 9 | 5 1 0 1 |
| 9 8 | 7 1 1 0 |
| 8 1 | |
| 3 7 | |

| INSULE.IN | INSULE.OUT |
|-----------|----------------------------------|
| 6 3 | comunitatea 1 are insulele:2 6 5 |
| 1 4 | matricea de adiacenta: |
| 2 6 | # 2 5 6 |
| 6 5 | 2 0 0 1 |
| | 5 0 0 1 |
| | 6 1 1 0 |

CL1. Culori

40 p

Gigel este mare pasionat al jocului Minesweeper. El descoperă că suprafața dreptunghiulară a jocului este împărțit în mai multe arii colorate diferit. Ariile pot fi colorate în 9 culori diferite codificate de la 1 la 9. Pot exista mai multe arii colorate cu aceeași culoare, iar minele întotdeauna se găsesc în aria cu suprafața maximă.

Cerință

Trebuie să ajutați pe Gigel să găsiți din fiecare culoare zona cu aria maximă.

Date de intrare

Prima linie a fișierului de intrare culori.in conține două numere întregi n și m , separate printr-un spațiu, numărul de linii, respectiv de coloane a matricii. Următoarele n linii conțin câte m numere întregi de la 1 la 9 care reprezintă diferitele culori.

Date de ieșire

În fișierul culori.out veți afișa pe 9 linii numerele care reprezintă maximele pentru zonele colorate începând cu maximul pentru zona codificată cu 1 și terminând cu zona codificată cu 9.

Restricții și precizări

- $2 \leq n, m \leq 100$
- Soluțiile sunt unice
- Totdeauna există cel puțin o zonă colorată

Exemplu

| culori.in | culori.out | Explicatie |
|-----------|------------|--|
| 6 5 | 8 | Există 2 zone codificate cu 1 iar aria maximă este 8. Există 3 zone codificate cu 2 iar aria maximă este 6. Etc. |
| 1 1 1 1 2 | 6 | |
| 2 1 1 2 2 | 5 | |
| 2 2 1 1 3 | 0 | |
| 2 3 3 3 3 | 0 | |
| 1 1 2 2 2 | 1 | |
| 1 2 2 2 6 | 0 | |
| | 0 | |
| | 0 | |

Timp maxim de execuție/test: **1 secunde**

CL2. Primpal

30 puncte

Să se afișeze toate numerele prime și palindrome dint-un interval dat. Un număr este palindrom dacă este egal cu oglinditul lui. (de exemplu 2346432). Un număr este prim dacă nu are alt divizor decât 1 și numărul în sine. (de exemplu 13).

Cerință

Să se afișeze toate numerele prime și palindrome dint-un interval dat.

Date de intrare

Fișierul de intrare **primpal.in** conține pe linia 1 a si b două numere naturale nenule, separate printr-un spațiu.

Date de iesire

Fișierul de ieșire **primpal.out** va conține pe linii separate (câte un singur număr/linie) toate numerele care satisfac condiția din enunț. Pe prima linie se va scrie numărul de astfel de numere din interval.

Restricții

- $3 \leq a, b \leq 10000000$

Exemplu

| primpal.in | primpal.out |
|-------------|--|
| 6 25 | 2 7 11 |
| 19020 31360 | 9 19391 19891 19991 30103 30203 30403 30703 30803 31013 |

Timp maxim de executare/test: 1 secundă

CL3. tractor**100p**

Ion stă în vârful unui deal. El culege mure cu ajutorul unui tractor modificat. Operația este foarte dificilă tractorul putând să deplaseze doar spre sud și vest din cauza terenului. Noroc cu vecinul Gigel care are o hartă exactă a dealului cu toate tufe de mure, precum și cu cantitatea de mure pe care o conțin. Harta este de dimensiunile $n \times m$. Ion intră cu tractorul prin colțul Nord-Vest (poziția (1,1) a matricii) și trebuie să ajungă la poalele dealului în colțul Sud-Est (poziția (n,m) a matricii) pe o rută optimă (adică să culegă cât mai multe mure).

Cerință

Trebuie să îl ajutați pe Gigel să construiți o hartă a dealului cu parcursul tractorului pe care să indicați numărul de deplasări elementare optime, precum și direcțiile de deplasare pe care trebuie să urmeze Ion, precum și cantitatea de mure culese! Tractorul se poate deplasa în două direcții: Est și Sud.

Date de intrare

Prima linie a fișierului de intrare tractor.in conține două numere întregi n și m , separate printr-un spațiu, numărul de linii, respectiv de coloane, ale matricii care reprezintă dealul cu tufe de mure. Următoarele n linii conțin câte m numere întregi care reprezintă cantitatea de mure a fiecărei tufe, separate printr-un singur spațiu.

Date de ieșire

În fișierul tractor.out veți afișa pe prima linie numărul maxim de mure culese max pe a doua linie L caractere din multimea E, S reprezentând deplasările efectuate, iar începând cu cea de-a treia linie harta dealului cu tufe de mure, tufe de pe care recolta a fost culeasă fiind marcate cu *. Harta se tipărește folosind 3 caractere pentru fiecare număr, numerele fiind separate printr-un spațiu.

Restricții și precizări

- $2 \leq n, m \leq 150$
- $0 \leq \text{recolta/tufă} \leq 999$
- $0 \leq \text{capacitate tractor} \leq 65535$
- Soluțiile sunt unice

Exemplu

| tractor.in | tractor.out | Explicatie |
|-------------|-------------|---|
| 5 5 | 256 | Drumul parcurgeri optime este marcat cu steluțe. Ion culege 256 de mure și străbate traseu indicat. Fiecare număr este tipărit folosind trei caractere (precedat de maxim două) și urmat de un singur spațiu. |
| 1 2 1 1 12 | ESSESSEE | |
| 1 3 1 1 1 | * * 1 1 12 | |
| 1 4 2 1 3 | 1 * 1 1 1 | |
| 2 1 113 4 2 | 1 * * 1 3 | |

| | | |
|----------------|--------------------------|--|
| 3 111 113 3 15 | 2 1 * 4 2 3 111 * * * | |
|----------------|--------------------------|--|

Timp maxim de execuție/test: **1 secunde**

RM1. Suma de numere consecutive

20 puncte

Se dă un număr natural **n**. Scrieți un program care să determine toate reprezentările numărului natural **n** ca suma de cel puțin două numere naturale consecutive.

Date de intrare

Fișierul de intrare **numar.in** conține numărul natural.

Date de ieșire

Fișierul de ieșire **numar.out** va conține pe fiecare linie câte o reprezentare a numărului natural **n**, precedat de textul „Solutia nr. k”, unde **k** este numărul de ordine al reprezentării curente.

Restricții și precizări

$1 < n < 1000$

Exemplu :

| numar.in | numar.out |
|----------|--|
| 15 | Solutia nr. 1: 7+8 Solutia nr. 2: 4+5+6 Solutia nr. 3: 1+2+3+4+5 |

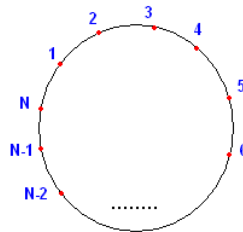
Timp de rulare/test: 1 secundă

PF1. Patinoar

100 puncte

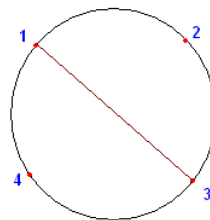
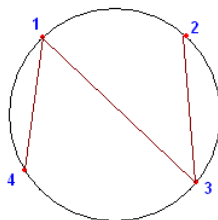
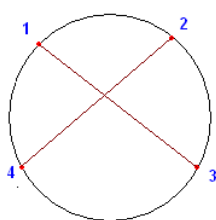
Gigel deține un patinoar circular și deoarece se apropie Crăciunul el dorește să îl orneze cât mai frumos în felul următor:

- există **N** puncte de interes dispuse circular în jurul patinarului
- fiecare punct este identificat prin numărul său de ordine, și sunt ordonate crescător
- punctele sunt la distanțe egale unul de celălalt



- Gigel dorește să unească aceste puncte cu ghirlande care generează un anumit grad de frumusețe. Gradul de frumusețe obținut de unirea a două puncte *i* și *j* este dat într-o matrice în a_{ij}
- o ghirlandă are capete unice și nu se poate intersecta cu o altă ghirlandă
- toate punctele trebuie unite !

Greșit :



Astfel Gigel dorește să obțină cel mai mare grad de frumusețe posibil pentru patinoarul lui.

Restricții:

- N este par
- $1 \leq N \leq 200$
- $1 \leq \text{costul de conectare} \leq 1000$

Input :

Prima linie a fișierului 'patinoar.in' conține N numărul de puncte. Următoarele N linii conțin câte N numere. Pe linia $i + 1$ se află numerele $a[i][1] a[i][2] \dots a[i][j] \dots a[i][N]$ ce semnifică gradul de frumusețe ce îl generează unirea punctelor i și j cu o ghirlandă.

Output:

În fișierul 'patinoar.out' se găsește un singur număr ce reprezintă costul maxim de conectare.

Exemplu:

Input:

```
4
0 4 7 3
4 0 8 1
7 8 0 9
3 1 9 0
```

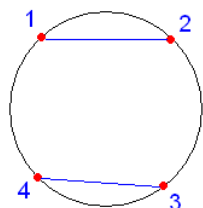
Output:

```
13
```

Explicatie:

Gradul de frumusețe maxim se obține prin unirea punctelor 1 cu 2 care generează un grad de 4 și 3 cu 4 care are un grad de 9. Gradul total de frumusețe este $9 + 4 = 13$.

Respectând condițiile, un grad mai mare de frumusețe nu se poate obține.



SZ1. Parada animalelor.

100 P

În pădurea rotundă, regele animalelor, Leul dorește să organizeze defilarea tuturor animalelor. La invitație au răspuns afirmativ doar patru specii de animale: Zebrele, Maimuțele, Elefanții și Tigrii.

Leul așa dorește ca, fiecare linie din coloana de defilare să fie alcătuită de n animale, iar liniile să difere două câte două privind ordinea animalelor.

Animalele, la rândul lor, au și ele condițiile lor:

- Maimuțele nu vor să fie la extremitatea vreunui rând, și nu vor să aibă ca vecin o altă Maimuță sau Tigru.

- Elefanții și Zebrele considerându-se mai frumoși decât Tigrii, vor ca în momentul onorului la stânga, la dreapta lor, undeva în linie să fie cel puțin un Tigru, și în nici un caz linia să nu înceapă cu un Tigru.

- Pe fiecare linie există cel puțin o Zebră sau un Elefant.

Aranjările de mai jos sunt corecte:

```
Z E M Z T
E M E T T
Z T T T T
```

Aranjările de mai jos nu sunt corecte:

```
Z E Z M T (Maimuță vecin cu Tigru)
Z T Z M Z (două Zebre nu văd spre dreapta nici un Tigru )
M Z E Z T (Maimuța este la extremitate)
```

Cerința problemei: Cunoscând numărul n de animale, care pot fi într-o linie, se cere numărul tuturor liniilor distincte două câte două care respectă condițiile de mai sus.

Date de intrare

Fișierul **PARAD.IN** conține pe o singură linie valoarea n ($1 \leq n \leq 100$).

Date de ieșire

Fișierul **PARAD.OUT** conține pe o singură linie numărul tuturor variantelor distincte (un număr natural cu maxim 60 cifre).

Ex.

| | | | | | | |
|---------------|----------------|--|---------------|-----------------|----------------|--------------------------|
| PARAD.IN 3 | PARAD.OUT 6 | Explicație ZZT ZET ZTT EZT EET ETT | PARAD.IN 4 | PARAD.OUT 22 | PARAD.IN 22 | PARAD.OUT 86502108014 |
|---------------|----------------|--|---------------|-----------------|----------------|--------------------------|

Timp maxim de execuție/test: 0.5 secunde.

SZ2. Casino

100 P

Într-un casino s-a inventat un joc nou. Având la dispoziție un labirint cu mai multe camere așezate matriceal, jucătorul intră pe teren din direcția Vest și trebuie să iasă din labirint pe partea estică. În fiecare cameră îl așteaptă o surpriză plăcută (primește recompensă) sau neplăcută (va fi penalizat), dar el nu poate anticipa ce îl va aștepta în camera următoare. Jucătorul poate să intre într-o nouă cameră aflată într-una din direcțiile Nord, Sud sau Est, dar niciodată nu poate să se întoarcă pe ușa pe care a intrat.

Sumele câștigate sau pierdute se acumulează până ce jucătorul părăsește labirintul spre Est.

Conform regulii jocului, jucătorul poate intra în orice cameră din latura Vest și poate ieși din orice cameră a laturii Est.

Cerința problemei:

Cunoscând schema labirintului, să se calculeze care este suma maximă care se poate câștiga printr-o parcurgere optimă, și pe câte trasee distincte se pot obține aceste sume maxime, respectiv care este suma minimă care se poate câștiga printr-o parcurgere optimă, și pe câte trasee distincte se pot obține aceste sume minime.

Date de intrare

Fișierul **CASINO.IN** conține o matrice cu m linii și n coloane ($m, n \leq 120$).

linia 1 – valorile lui m și n despărțite cu un spațiu,

liniile 2,..., $m+1$ – conțin câte o linie a matricei cu elemente numere întregi separate cu un spațiu.

(fiecare element $a_{i,j}$ încadrându-se în intervalul $[-255, 255]$).

Date de ieșire

Fișierul **CASINO.OUT** conține patru linii

linia 1 – suma maximă ce se poate câștiga

linia 2 – numărul soluțiilor maxime distincte

linia 3 – suma minimă ce se poate câștiga

linia 4 – numărul soluțiilor minime distincte

Observație

Atât sumele cât și numărul de soluții distincte sunt numere de maxim 9 cifre.

Exemplu

| CASINO.IN | CASINO.OUT | Explicație |
|-----------|------------|--|
| 3 3 | -3 | Suma maximă -3 se poate obține în 3 moduri distincte, iar suma minimă -9 se poate obține în 2 moduri distincte : |
| -1 -1 -1 | 3 | |
| -1 -1 -1 | -9 | |
| -1 -1 -1 | 2 | |
| minim: | | maxim : |
| | | |
| | | |
| | | |

Timp maxim de execuție/test: 2 secunde.

CH1. Depozit

100 puncte

Bulache s-a privatizat și și-a tras un depozit de grane cu n camere, fiecare camera având capacitatea q . Când a cumpărat depozitul, Bulache a găsit cantitățile x_1, x_2, \dots, x_n în camere. Bulache face 2 lucruri :

- vinde cantitatea p din camera i
- încarcă cantitatea r în camera j

Totodată, când bea prea multă bere lui Bulache îi vin idei colorate. Astfel el ar vrea să știe 2 lucruri :

- câte marfa are în camerele cuprinse între indicii i și j ($i \leq j$), inclusiv
- câte marfa mai poate încărca în camerele cuprinse între indicii i și j ($i \leq j$), inclusiv

Date de intrare

Fișierul de intrare **depozit.in** conține pe prima linie numărul n de camere și capacitatea q a fiecărei camere. Pe următoarele linii conține, pe câte o linie, sau o acțiune sau o interogare, în funcție de litera cu care începe rândul. Dacă litera e V sau I, atunci Bulache trebuie să își actualizeze stocul cu cantitatea care urmează după litera, iar dacă litera e S sau L urmată de 2 numere se va răspunde pe o linie din fișierul **depozit.out** la această interogare.

Date de ieșire

Fișierul de ieșire **depozit.out** va conține atâtea linii câte interogări avem în fișierul de intrare.

Restricții și precizări

Numărul de interogări este $\leq 100\,000$.

Exemplu :

| depozit.in | depozit.out |
|---|--------------------------------------|
| 5 10 | 11 (sunt 11 unități în depozit) |
| 2 1 3 4 1 | 39 (se mai pot pune în depozit 39) |
| S 1 5 | 40 (se mai pot pune în depozit 40) |
| L 1 5 | 2 (în camera 4 se mai pot încărca 2) |
| V 4 1 (se vinde din camera 4 cantitatea 1) | 6 (stocul din camerele 1 și 2 e 6) |
| L 1 5 | |
| L 4 4 | |
| I 1 3 (se încarcă în camera 1 cantitatea 3) | |
| S 1 2 | |

Timp de rulare/test: 1 secundă

CH2. Secvente

80 puncte

Fie n numere naturale și m și $p \geq 2$ numere mai mici decât n care îndeplinesc condiția $m \cdot p < n$. Să se determine suma maximă care se poate obține alegând p subsecvențe distincte de maxim m elemente din șirul inițial.

Date de intrare

Fișierul de intrare **secv.in** conține pe prima linie numerele n, m și p .

Pe următoarea linie conține cele n numere.

Date de ieșire

Fișierul de ieșire **secv.out** va conține suma căutată.

Restricții și precizări

Numărul $n \leq 10000$, iar numerele mai mici decât 100.

Exemplu :

| secv.in | Secv.out |
|-------------------------------|----------|
| 7 2 3 35 40 50 10 30 45 60 | 240 |

Timp de rulare/test: 1 secundă

Se dau n puncte in plan prin coordonatele x,y. Sa se determine aria poligonului conex de arie minima care contine toate punctele.

Date de intrare

Fişierul de intrare **puncte.in** conţine pe prima linie numarul n iar pe urmatoarele linii perechi de coordonate x si y.

Date de ieşire

Fişierul de ieşire **puncte.out** va conţine aria poligonului scrisa cu 2 zecimale exacte.

Restricţii şi precizări

Numarul n <= 10000.

Exemplu :

| puncte.in | Puncte.out |
|---|------------|
| 6 1 1 2 2 1 3 3 1 3 3 2 3 | 4.00 |

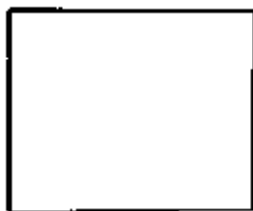
Timp de rulare/test: 1 secunda

Redactati programe care sa deseneze urmatoarele figure recursive. Pentru fiecare figura datele de intrare sunt *nivelul de recursivitate* si *lungimea pasului*. Fiecare problema are 40 p.

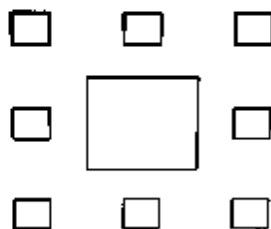
KB1.

Fractal-1

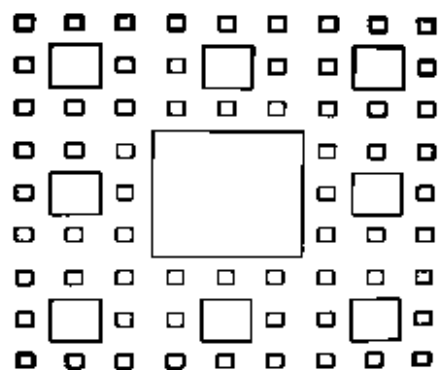
k=1



k=2



k=4



frac1.in

1
100

frac1.in

2
100

frac1.in

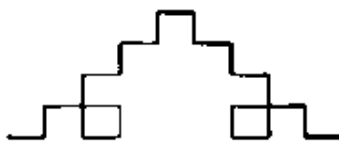
4
100

Fractal-2

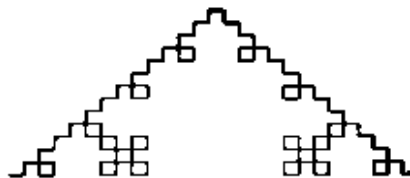
k=1



k=2



k=4



frac2.in

1
95

frac2.in

2
95

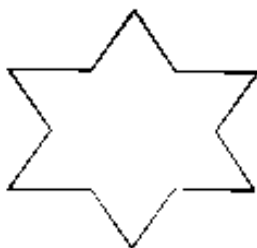
frac2.in

4
95

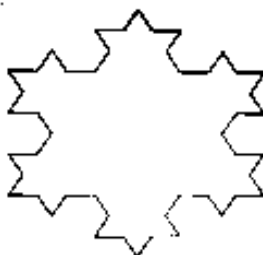
KB3.

Fractal-3

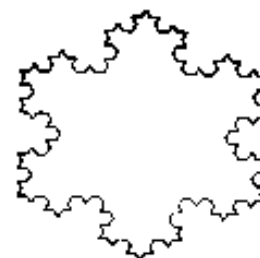
k=1



k=2



k=4



Frac3.in

1
300

Frac3.in

2
300

Frac3.in

4
300

KB4.

Fractal-4

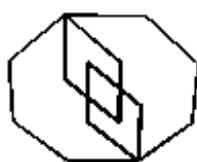
k=1



k=2



k=4



k=6



frac4.in

1
60

frac4.in

2
60

frac4.in

3
60

frac4.in

6
10