

be-OI 2026 Finale - SENIOR Zaterdag 14 maart 2026	Invullen in HOOFDLETTERS en LEESBAAR aub VOORNAAM : NAAM : SCHOOL :	O Gereserveerd
---	--	---

Finale van de Belgische Informatica-olympiade 2026 (duur: 2u)

Instructies die je aandachtig moet lezen voor de proef.

1. **Wacht op het startsignaal** voordat je de bladen uit het plastic mapje haalt.
2. Je moet de finale afleggen in dezelfde taal als die waarin je de kwalificatie hebt afgelegd.
Controleer of deze pagina in de juiste taal is. Als dat niet het geval is, meld het of verander van plaats.
3. Controleer of je de juiste **reeks vragen** hebt gekregen die hierboven in de hoofding vermeld staat.
 - Voor leerlingen tot en met het tweede jaar middelbaar: categorie **belofte, roze** bladen.
 - Voor leerlingen in het derde of vierde jaar middelbaar: categorie **junior, groene** bladen.
 - Voor leerlingen vanaf het vijfde jaar middelbaar: categorie **senior, gele** bladen.
4. Wanneer het startsignaal gegeven wordt, haal je de bladen uit het mapje, maar **verwijder de nietjes niet**.
5. Vul je naam, voornaam en school in, **zeer leesbaar in HOOFDLETTERS en enkel op deze pagina**.
6. Vul je **antwoorden in op de gekleurde bladen** die met deze pagina geniet zijn.
Schrijf **goed leesbaar** met een blauwe of zwarte **bic of pen**.
7. Gebruik de witte bladen als kladwerk.
Vergeet niet je oplossingen over te schrijven op de gekleurde bladen.
8. De proef duurt 2 uur en je moet op je plaats blijven tot het einde.
Als je vroeger klaar bent, roep je een toezichthouder om je antwoorden in te dienen.
9. Wanneer het eindsignaal gegeven wordt, moet je onmiddellijk je oplossingen inleveren.
 - Stop de bladen niet terug in het plastic mapje.
 - **Leg de gekleurde bladen, nog steeds geniet,** in een van de daarvoor voorziene dozen.
 - **Geef het plastic mapje terug** door het in een daarvoor voorziene doos te leggen.
 - Houd de witte bladen bij.
10. Je mag alleen schrijfgerief bij je hebben. Rekentoestellen, smartphones, ... zijn **verboden**.
11. Alle codefragmenten in de opgaven zijn in **pseudo-code**. Op de volgende pagina's vind je een **beschrijving** van de pseudo-code die we gebruiken. Als je in code moet antwoorden, mag je **pseudo-code** of een **courante programmeertaal** gebruiken (Java, C, C++, Pascal, Python, ...). Syntaxfouten worden niet in rekening gebracht bij de beoordeling.

De Belgische Informatica-olympiade wordt mogelijk gemaakt dankzij de steun van onze leden:



Geheugensteun pseudo-code

Gegevens worden opgeslagen in variabelen, elk geïdentificeerd door een naam.

De naam geeft toegang tot de variabele om gegevens op te slaan of op te halen.

We gebruiken $\leftarrow$ als toekenningoperator om een waarde in een variabele op te slaan.

Voorbeeld: $n \leftarrow 100$ plaatst het geheel getal 100 in de variabele genaamd n .

Variabelen en eenvoudige gegevens	Voorbeelden
geheel getal	$n \leftarrow 100$ $m \leftarrow -1$
reëel getal	$pi \leftarrow 3.1415$ $z \leftarrow 0.0$
logische waarde (booleaans)	$t \leftarrow \mathbf{true}$ (waar) $t \leftarrow \mathbf{false}$ (onwaar)

Je kan rekenkundige bewerkingen uitvoeren met getallen en variabelen.

optelling en aftrekking met $+$ en $-$ (voorbeeld: $a+3$)
vermenigvuldiging met $*$ of $\times$ (voorbeeld: $2*a*b$)
gehele deling: quotiënt met $/$ of $//$ en rest met $\%$ (voorbeeld: $14//3$ is gelijk aan 4 en $14\%3$ is gelijk aan 2)
niet-gehele deling $:$: weinig gebruikt, we geven het aan als dat het geval is (voorbeeld: $3.6/4.0$ is gelijk aan 0.9)
macht of exponent met $^$ (voorbeeld: x^3 is gelijk aan $x*x*x$)

We gebruiken vaak variabelen om een resultaat te berekenen

en het vervolgens in een variabele te plaatsen.

Soms wordt het resultaat in een van de variabelen geplaatst die in de berekening worden gebruikt.

Het kader rechts bevat code die dit illustreert.

Na het uitvoeren van deze code geldt $x=25$, $a=4$ en $b=10$.

```

a ← 3
b ← 5
x ← a*b + 10
a ← a + 1
b ← b*2

```

```

if (a < b)
    { p ← p+5 }
else
    { p ← p-2 }
c ← c-1

```

De instructie **if** voert code alleen uit als een voorwaarde waar is.

Je kan eventueel de instructie **else** toevoegen om andere code uit te voeren alleen als de voorwaarde onwaar is.

In het voorbeeld links, als het getal in variabele a kleiner is dan dat in variabele b , dan wordt 5 opgeteld bij variabele p , anders wordt 2 afgetrokken van variabele p .

Daarna wordt in alle gevallen 1 afgetrokken van variabele c .

De instructies die in de verschillende gevallen moeten worden uitgevoerd, worden duidelijk aangegeven door ze tussen accolades $\{ \}$ te plaatsen, of door ze in te springen.

Heel vaak doen we beide: tussen accolades plaatsen en inspringen, zoals hierboven.

Hier is de lijst van de meest gebruikte vergelijkingsoperatoren.

$=$ of $==$ is gelijk aan	$<$ kleiner dan	$<=$ of $\leq$ kleiner dan of gelijk	$>$ groter dan	$>=$ of $\geq$ groter dan of gelijk	$!=$ of $\neq$ is verschillend van
------------------------------	--------------------	---	-------------------	--	---------------------------------------

Je kan meerdere voorwaarden testen door ze te combineren met de logische operatoren **and**, **or**, **not**.

De voorwaarde (p and q) is waar als de 2 voorwaarden p en q beide waar zijn.
De voorwaarde (p or q) is waar als minstens een van de 2 voorwaarden waar is.
De voorwaarde not (p) is waar als p onwaar is en onwaar als p waar is.
Je kan de logische waarde van een voorwaarde opslaan in een booleaanse variabele om later te gebruiken.
Voorbeeld: $f \leftarrow ((a==5) \text{ and } (b>=0)) \text{ or } ((a<0) \text{ and } (b!=10))$
De voorwaarde van een if is soms gewoon een booleaanse variabele.
Voorbeeld: if (f) {a←10} else {b←10}

Soms moeten meerdere gegevens in een enkele gestructureerde variabele worden opgeslagen.

Variabelen en gestructureerde gegevens	Voorbeelden
tabel, lijst, vector	$seq \leftarrow [7, 11, 0, -4, 9]$
tabel, matrix	$M \leftarrow [[0, 1, 2], [3, -1, 3], [0, 0, 5]]$
koppel	$coord \leftarrow (1, 7)$
tekst	$n \leftarrow \text{"John"}$

De individuele elementen van een gestructureerde variabele worden aangeduid met een index die tussen vierkante haken na de naam van de variabele wordt geschreven.

Het eerste element van een gestructureerde variabele G heeft index 0 en wordt genoteerd als $G[0]$.

Het tweede element heeft index 1 en het laatste heeft index $n-1$ als er in totaal n elementen zijn.

Het aantal elementen van een gestructureerde variabele wordt teruggegeven door de functie $len()$.

Voorbeeld: als $G = [5, 9, 12]$ dan geldt $len(G) = 3$, $G[0] = 5$, $G[1] = 9$ en $G[2] = 12$.

De tabel heeft grootte 3, maar de hoogste index is 2.

Om code te herhalen, bijvoorbeeld om de elementen van een tabel te doorlopen, kan je een **for**-lus gebruiken.

```
sum ← 0
n ← len(T)
for (i ← 0 to n - 1 step 1)
{
    sum ← sum + T[i]
}
```

De notatie **for** (i←a **to** b **step** k)

staat voor een lus

- waarin i begint met de waarde a,
- die herhaald wordt zolang $i \leq b$,
- die i met k verhoogt aan het einde van elke stap.

Het voorbeeld links berekent de som van de elementen van de tabel T. De elementen worden een voor een opgeteld in de variabele sum.

De instructies die in een lus moeten worden uitgevoerd, worden duidelijk aangegeven door ze tussen accolades { } te plaatsen,

of door ze in te springen. Heel vaak doen we beide: tussen accolades plaatsen en inspringen, zoals hierboven.

Je kan ook een lus schrijven met de instructie **while**, die code herhaalt zolang de voorwaarde waar is.

In het voorbeeld rechts delen we een positief geheel getal n door 2,
dan door 3, vervolgens door 4 ...
totdat het nog maar uit één cijfer bestaat
(dat wil zeggen totdat $n < 10$).

```
d ← 2
while (n ≥ 10) {
    n ← n/d
    d ← d+1
}
```

Vraag 1 – Het verloren paleis aflevering 1: tellen.

Archeologen hebben een oud paleis met vele zalen ontdekt.

Maar er zijn vallen die ze niet kunnen vermijden.

Ze hebben professor Zarbi te hulp geroepen.

De professor begreep al snel dat je op bepaalde gegraveerde tegels moet stappen om de vallen uit te schakelen.

In de zalen zijn de gegraveerde tegels altijd in meerdere horizontale rijen geplaatst.

De eerste rij bestaat uit slechts één gegraveerde tegel, de zogenaamde *top*.

De professor heeft ontdekt dat je vooruit moet gaan volgens bepaalde regels, anders gaat er onvermijdelijk een val af.

1. Je moet starten vanaf de top (bovenaan in de voorbeelden) en enkel op gegraveerde tegels stappen.
2. Bij het vooruitgaan moet je altijd van een gegraveerde tegel naar een andere gaan die deze raakt (niet enkel via een hoek).
3. Je moet precies één gegraveerde tegel per rij betreden.
4. Je moet de laatste rij gegraveerde tegels bereiken (helemaal onderaan in de voorbeelden).

Vanaf nu betekent het woord **pad** een doorkruising die aan deze 4 regels voldoet.

Vanaf nu zijn alle genoemde of getekende **tegels** gegraveerde tegels.

De tegels zijn vaak in een piramide geplaatst, zoals in alle voorbeelden op deze pagina.

We noteren **P_n** voor een piramide bestaande uit n rijen tegels.

Voorbeeld 1

Hier zijn piramides **P₃**, **P₄** en **P₅**.

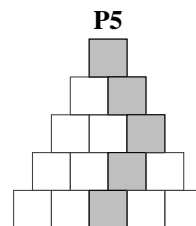
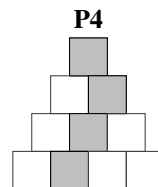
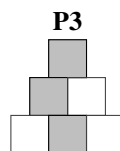
In elk is een voorbeeld van een pad grijs gekleurd.

De top is grijs (regel 1).

De grijze tegels van aangrenzende rijen raken elkaar (regel 2).

Er is slechts één grijze tegel per rij (regel 3).

Er is een grijze tegel in de laatste rij (regel 4).



Professor Zarbi wil tellen hoeveel verschillende paden er zijn om piramides te doorkruisen.

Voorbeeld 2

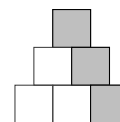
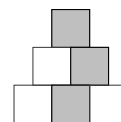
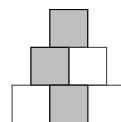
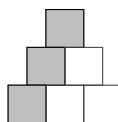
Hier zijn alle paden die **P₃** doorkruisen.

1 pad bereikt de linkertegel van de laatste rij.

2 paden bereiken de middelste tegel van de laatste rij.

1 pad bereikt de rechtertegel van de laatste rij.

In totaal doorkruisen 4 verschillende paden **P₃**.



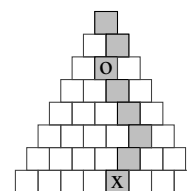
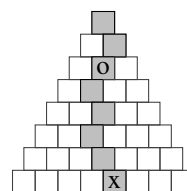
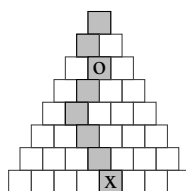
Soms wil hij enkel de paden tellen die door bepaalde tegels gaan.

Voorbeeld 3

Hier zijn 3 paden die **P₈** doorkruisen.

Ze gaan allemaal door de tweede tegel van de derde rij (tegel gemarkeerd met een o).

Ze bereiken allemaal de vijfde tegel van de laatste rij (tegel gemarkeerd met een x).



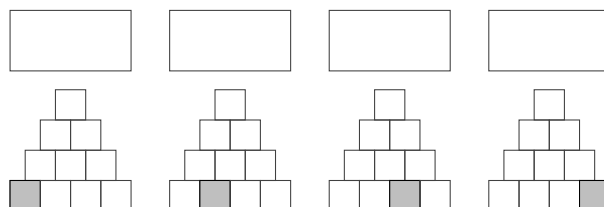
Opmerking: op een van de laatste pagina's van deze vragenlijst staan lege piramides tot je beschikking voor je kladwerk.

De paden tellen die een bepaalde tegel bereiken.

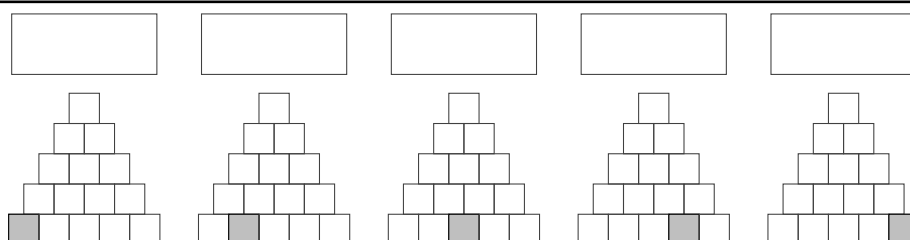
In de volgende vragen zijn piramides getekend met 1 grijze tegel in de laatste rij.

Hoeveel verschillende paden bereiken de grijze tegel? Schrijf de antwoorden in de rechthoekjes.

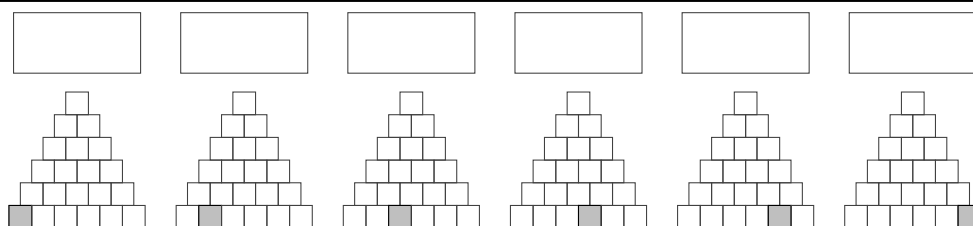
Q1(a) /4	Hoeveel paden bereiken de grijze tegel in P4?
-----------------	--



Q1(b) /5	Hoeveel paden bereiken de grijze tegel in P5?
-----------------	--



Q1(c) /6	Hoeveel paden bereiken de grijze tegel in P6?
-----------------	--

**Alle paden tellen.**

Q1(d) /1	In totaal, hoeveel paden doorkruisen P2?
-----------------	---



Q1(e) /1	In totaal, hoeveel paden doorkruisen P4?
-----------------	---



Q1(f) /1	In totaal, hoeveel paden doorkruisen P5?
-----------------	---

Q1(g) /1	In totaal, hoeveel paden doorkruisen P6?
-----------------	---

Q1(h) /1	In totaal, hoeveel paden doorkruisen Pn (waarbij n een positief geheel getal is)?
-----------------	--

De paden tellen die door bepaalde tegels gaan.

In de volgende vragen zijn piramides getekend met een of meer grijze tegels.

Schrijf in het rechthoekje boven de piramide het aantal verschillende paden die door de grijze tegels gaan. **Vergeet de 4 regels niet!**

Q1(i) /6	Hoeveel paden gaan door de grijze tegels van de piramides P6?

Tegels in diamantvorm.

De tegels zijn vaak in een piramide geplaatst, maar dat is niet altijd het geval.

Er bestaat bijvoorbeeld een diamantvormige opstelling waar de laatste rij slechts één tegel heeft.

In de volgende vragen zijn zalen in diamantvorm getekend met een of meer grijze tegels.

Schrijf in het rechthoekje boven de diamant het aantal verschillende paden die door de grijze tegels gaan.

Q1(j) /8	Hoeveel paden gaan door de grijze tegels van de diamanten?

Vreemde zalen.

De tegels van bepaalde zalen zijn op een ingewikkelde manier opgesteld.

In de volgende vragen moet je alle paden tellen die vreemde zalen doorkruisen.

Schrijf in het rechthoekje boven de zaal het totale aantal paden die deze doorkruisen (tot aan een grijze tegel).

Q1(k) /6	In totaal, hoeveel paden doorkruisen deze vreemde zalen?

Stappen in de grote piramides.

Sommige zalen bevatten zeer grote piramides van tegels en professor Zarbi wil een manier vinden om het tellen van paden te vereenvoedigen.

Hij gebruikt de onderstaande afspraken.

- De n rijen tegels van een piramide P_n zijn genummerd van 0 tot $n-1$ van boven naar beneden.
- In een piramide heeft rij nummer r altijd $r+1$ tegels genummerd van 0 tot r van links naar rechts.
- De coördinaten van een tegel zijn (r, c) waarbij r de rij is en c het nummer in die rij.

Voorbeeld

De figuur hiernaast stelt een piramide P_{10} voor.

De rijnummers staan links weergegeven.

De nummers die in elke rij worden gebruikt staan op de tegels.

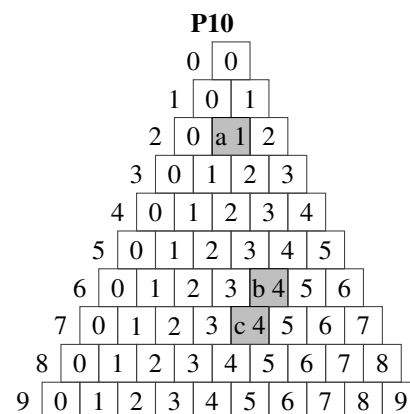
De 3 grijze tegels zijn aangeduid met de letters **a**, **b** en **c**.

Hun coördinaten zijn $(2, 1)$, $(6, 4)$ en $(7, 4)$.

Opmerking

Deze figuur geeft ook de coördinaten van de tegels in de kleinere piramides.

De eerste 4 rijen tonen bijvoorbeeld de coördinaten in een piramide P_4 .



Professor Zarbi telt in meerdere stappen de paden die door grijze tegels van een grote piramide gaan.

Elke stap gebruikt een deelpiramide.

- De eerste deelpiramide heeft dezelfde top als de grote piramide, en haar laatste rij bevat de eerste grijze tegel van de grote piramide.
- Vervolgens worden deelpiramides gebruikt waarvan de top een grijze tegel is, en waarvan de laatste rij de volgende grijze tegel bevat.
- Er is een bijzonder geval als de laatste rij van de grote piramide geen grijze tegel bevat. In dat geval is er geen grijze tegel in de laatste rij van de laatste deelpiramide.

Professor Zarbi gebruikt de volgende notaties.

- $C(r, c)$ is het aantal paden dat tegel (r, c) bereikt (in een piramide met $r+1$ rijen).
- $T(n)$ is het totale aantal paden dat een piramide P_n doorkruist (dus met n rijen).

Let op: voor deze BeOI-finale is het **verboden om $T(1)$ te gebruiken**.

Voorbeeld

Hier is de opsplitsing in 4 stappen van het tellen van de paden in P_{10} hierboven.

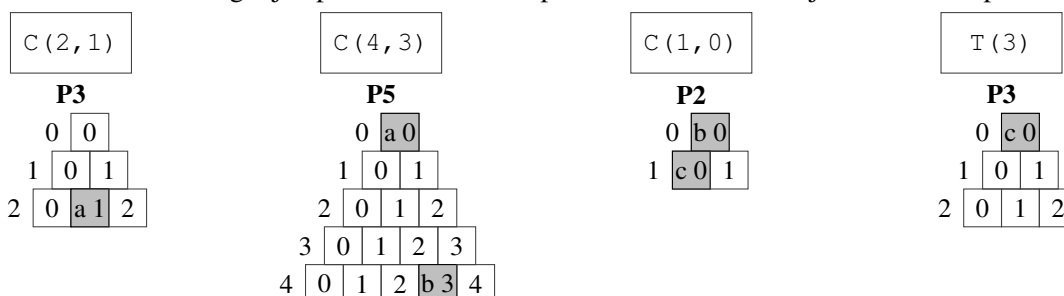
Eerste stap: van de top naar de eerste grijze tegel **a**.

Tweede stap: van de eerste grijze tegel **a** naar de tweede **b**.

Derde stap: van de tweede grijze tegel **b** naar de derde **c**.

Laatste stap: van de laatste grijze tegel **c** naar de laatste rij van P_{10} .

Het aantal mogelijke paden voor elke stap staat in het rechthoekje boven de stap.



Uiteraard moet je de aantallen mogelijke paden van de stappen gebruiken om het aantal paden door de grijze tegels in de grote piramide te bepalen.

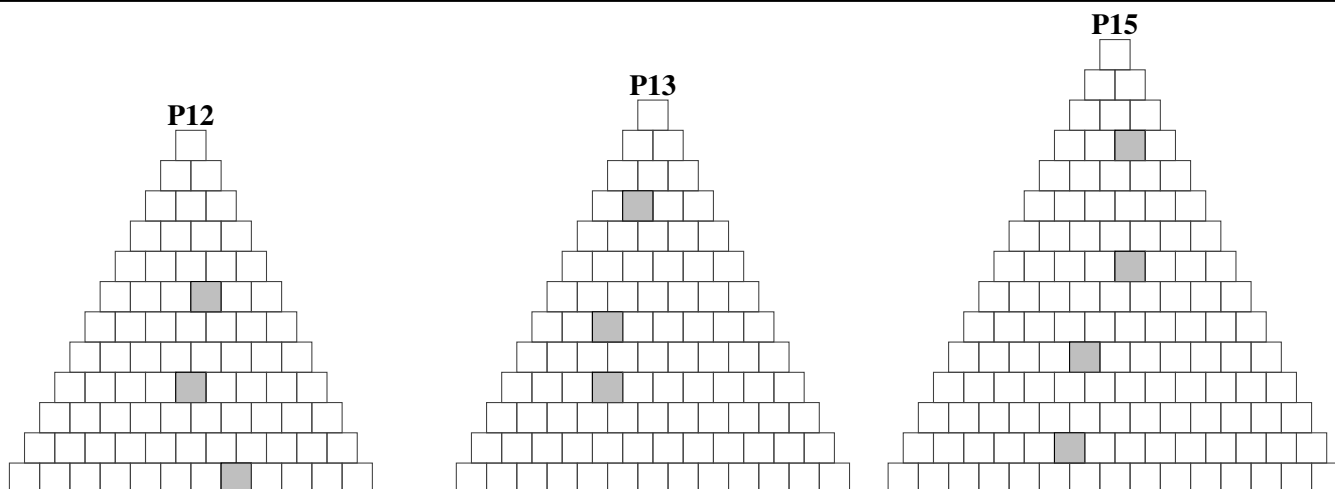
Geef in de volgende vragen een wiskundige uitdrukking met de notaties $C(r, c)$ en $T(n)$ om het aantal paden te bepalen die door de grijze tegels gaan van het voorbeeld **P10** van de vorige pagina en van de piramides **P12**, **P13** en **P15** hieronder.

Belangrijke opmerking

Bereken het eindantwoord niet !

Je moet een wiskundige uitdrukking schrijven die er bijvoorbeeld uit ziet als $C(5, 2) * T(3) + C(4, 2)$.

Q1(l) /2	Uitdrukking voor het aantal paden door de grijze tegels van P10 uit het voorbeeld.
Q1(m) /2	Uitdrukking voor het aantal paden door de grijze tegels van P12 .
Q1(n) /2	Uitdrukking voor het aantal paden door de grijze tegels van P13 .
Q1(o) /2	Uitdrukking voor het aantal paden door de grijze tegels van P15 .



Automatisering.

Professor Zarbi heeft een functie `countPyra(n, G)` geschreven om het tellen van het aantal paden door grijze tegels in een grote piramide te automatiseren.

Deze functie gebruikt 2 correct geïnitieerde parameters.

- n : een geheel getal dat het aantal rijen van de grote piramide bevat.
- G : een tabel van koppels (r, c) die de coördinaten van de grijze tegels bevat (in volgorde van boven naar beneden).

Het programma gebruikt ook de functies $C(r, c)$ en $T(n)$ die de eerder uitlegde waarden teruggeven.

Ten slotte gebruikt het programma een lus "**for** (r, c) **in** G " die evenveel keer wordt uitgevoerd als er elementen in G zijn.

Voorbeelden met de piramide P10 die eerder in dit document is getekend.

- $n=10$
- $G = [(2, 1), (6, 4), (7, 4)]$
Dus $G[0] = (2, 1)$, $G[1] = (6, 4)$, $G[2] = (7, 4)$.
- $(r0, c0) \leftarrow (0, 0)$ initialiseert de 2 variabelen tegelijk (dit is gelijkwaardig aan $r0 \leftarrow 0$ en $c0 \leftarrow 0$).
- $C(2, 1)$ geeft 2 terug en $T(3)$ geeft 4 terug.
- De lus "**for** (r, c) **in** G " wordt 3 keer uitgevoerd.
De eerste keer $(r, c) = (2, 1)$, de tweede keer $(r, c) = (6, 4)$, de laatste keer $(r, c) = (7, 4)$.

Professor Zarbi schrijft een eerste versie van het programma die geen enkele controle uitvoert.

Dit eenvoudige programma werkt correct met de 4 grote piramides **P10**, **P12**, **P13** en **P15** op voorwaarde dat n en G correct worden geïnitieerd.

Het programma gebruikt de volgende variabelen.

- nbr : het aantal paden door de grijze tegels van de grote piramide.
- h : het nummer van de laatste rij van de deelpiramide die momenteel wordt verwerkt (die dus $h+1$ rijen heeft).
- k : het nummer van de grijze tegel in de laatste rij van de huidige deelpiramide (met $0 \leq k \leq h$).

De volgende vraag levert tussen 0 en 6 punten op.

Je verliest 2 punten voor elk vakje dat niet correct is ingevuld. Er zijn geen negatieve punten.

Q1(p) /6	Vul de in de functie <code>countPyraV1(n, G)</code> aan.
-----------------	--

```
function countPyraV1(n, G) {
  nbr ← 1
  (r0, c0) ← (0, 0)

  for (r, c) in G {
    h ← 

    k ← 

    nbr ← 

    (r0, c0) ← (r, c)
  }
  if (r0 < ) { nbr ←  }
  return nbr
}
```

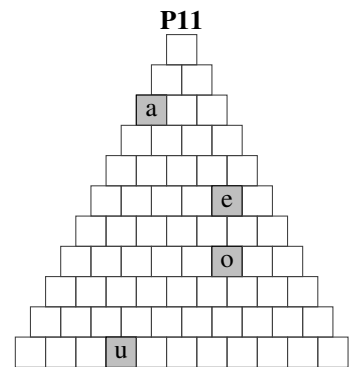
Bepaalde piramides kunnen problemen opleveren.

Voorbeeld

Er zijn geen paden die door alle grijze tegels van de piramide hiernaast gaan.

Het is onmogelijk om van **a** naar **e** te gaan.

Evenzo is er geen deelpiramide met top **o**
waarvan de laatste rij **u** bevat.



Het programma `countPyraV1` houdt geen rekening met deze gevallen en kan foute antwoorden geven of stoppen met een foutmelding!

Het programma `countPyraV2` is een verbetering die in alle gevallen het juiste antwoord geeft.

Professor Zarbi heeft enkel de coderegel `nbr ←` van `countPyraV1` vervangen door een test in `countPyraV2`.

Welke voorwaarden moeten worden gecontroleerd en wat moet er gebeuren naargelang ze waar of onwaar zijn?

Aan jou om de test te schrijven in de volgende vraag.

Deze vraag levert tussen 0 en 6 punten op.

Je verliest 2 punten voor elk vakje dat niet correct is ingevuld. Er zijn geen negatieve punten.

Q1(q) /6	Vul de in de test van <code>countPyraV2</code> die een regel van <code>countPyraV1</code> vervangt.
----------	--

```

if (  and  )
    {nbr ← }

else

    {nbr ← }

```

Vraag 2 – Het verloren paleis aflevering 2: ontcijferen.

*De ontdekkingen van professor Zarbi zijn nog niet voldoende om alle vallen te vermijden.
Maar de archeologen hebben de plattegronden van de zalen ontdekt tussen andere belangrijke documenten.*

Op deze plattegronden is op elke tegel een piramide met kleine schijfjes getekend.

De professor begrijpt dat deze piramides getallen voorstellen en hij slaagt erin ze te ontcijferen.

Opgelet! Verwar ze niet!

In aflevering 1 bestonden de piramides uit tegels.

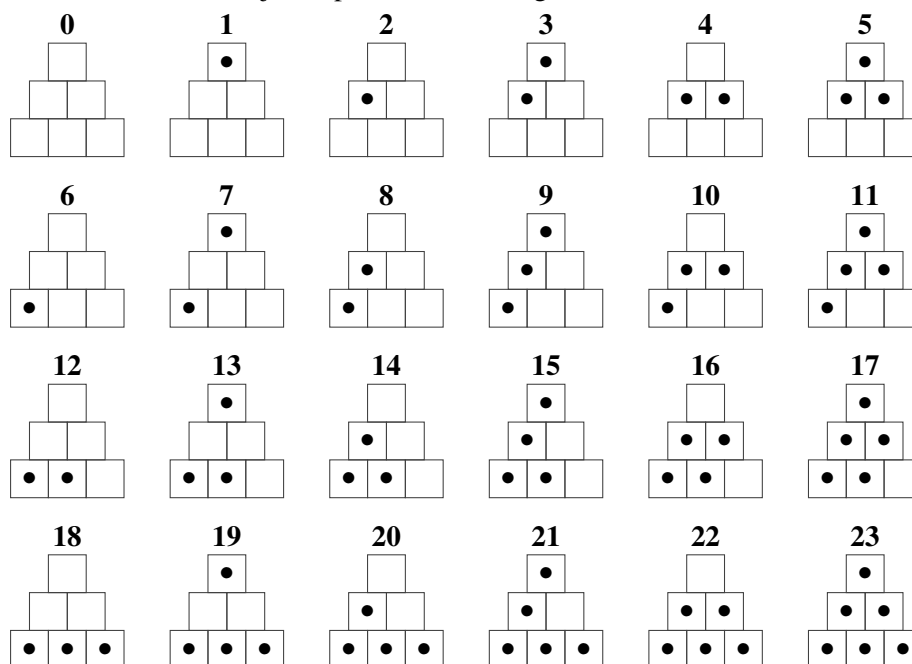
In aflevering 2 zijn de piramides getallen die op de tegels staan.

Het nummersysteem.

De getallen worden voorgesteld door piramides waarvan sommige vakjes een kleine zwarte schijf bevatten.

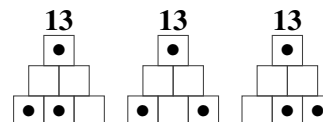
- 0 wordt voorgesteld door een piramide zonder enige schijf.
- Om van een piramide naar de volgende over te gaan, dat wil zeggen van een getal naar het volgende, ga je als volgt te werk.
 1. Vertrekkend vanaf de top zoek je de eerste rij die een leeg vakje bevat.
 2. Je plaatst een kleine schijf in een willekeurig leeg vakje van die rij.
 3. Je maakt alle vakjes van de hogere rijen leeg (die waren gevuld).

Voorbeelden: hier zijn **P3**-piramides die de getallen van 0 tot 23 voorstellen.

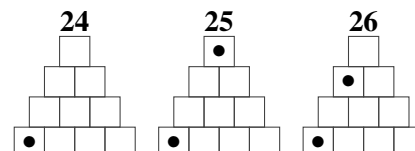
**Getallen met meerdere voorstellingen.**

Meerdere piramides kunnen hetzelfde getal voorstellen omdat je de zwarte schijven overal in hun rij kunt plaatsen (punt 2 hierboven).

Voorbeeld: de 3 piramides hiernaast stellen alle drie het getal 13 voor.

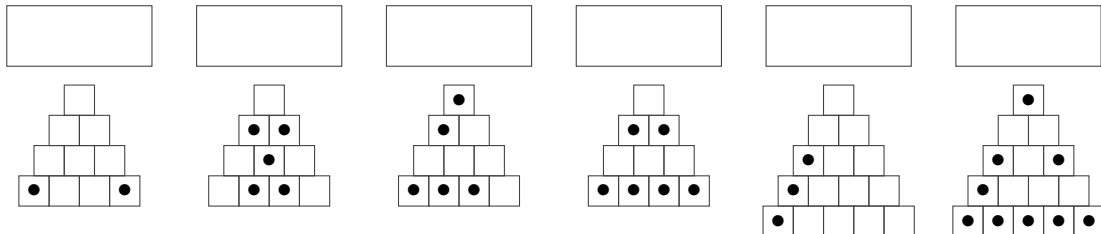
**Voorstellingen van grotere getallen.**

23 is het grootste getal dat je kunt voorstellen met een **P3**-piramide. Om verder te gaan moet je piramides met meer rijen gebruiken.

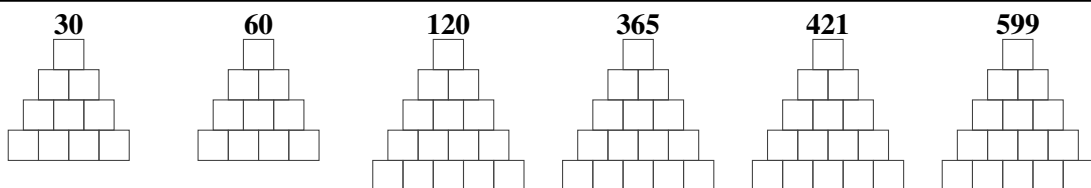


Opmerking: *een piramide tekenen* betekent *kleine zwarte schijven tekenen in bepaalde vakjes ervan*.
 Bijzonder geval: als je geen enkele schijf tekent, *teken je een lege piramide* die het getal 0 voorstelt.

Q2(a) /6	Schrijf in elke rechthoek het getal dat door de getekende piramide wordt voorgesteld.
-----------------	--



Q2(b) /6	Teken een piramide die het aangegeven getal voorstelt.
-----------------	---



Hierboven werd opgemerkt dat meerdere piramides hetzelfde getal kunnen voorstellen en dat 23 het grootste getal is dat je kunt voorstellen met een **P3**-piramide. Dit leidt tot de volgende vragen die je kunt beantwoorden met een getal of een wiskundige uitdrukking.

Q2(c) /1	Hoeveel verschillende P3-piramides kun je tekenen?
-----------------	---

Q2(d) /1	Hoeveel verschillende P4-piramides kun je tekenen?
-----------------	---

Q2(e) /1	Wat is het grootste getal dat je kunt voorstellen met een P4-piramide?
-----------------	---

Q2(f) /1	Hoeveel verschillende P5-piramides kun je tekenen?
-----------------	---

Q2(g) /1	Wat is het grootste getal dat je kunt voorstellen met een P5-piramide?
-----------------	---

Q2(h) /1	Hoeveel verschillende P6-piramides kun je tekenen?
-----------------	---

Q2(i) /1	Wat is het grootste getal dat je kunt voorstellen met een P6-piramide?
-----------------	---

In de volgende vragen moet je alle getallen geven die een bepaalde eigenschap hebben.

Als er meerdere getallen zijn, schrijf ze dan gescheiden door komma's.

Als er geen getal is, antwoord door een kruis te tekenen $\times$.

Q2(j) /1	Welke getallen hebben precies 1 voorstelling in P3?
Q2(k) /1	Welke getallen hebben precies 2 voorstellingen in P3?
Q2(l) /1	Welke getallen hebben precies 3 voorstellingen in P3?
Q2(m) /1	Welke getallen hebben precies 4 voorstellingen in P3?
Q2(n) /1	Welke getallen hebben precies 5 voorstellingen in P3?
Q2(o) /1	Welke getallen hebben precies 6 voorstellingen in P3?
Q2(p) /1	Welke getallen hebben precies 36 voorstellingen in P4?
Q2(q) /1	Wat is het kleinste getal dat precies 10 voorstellingen heeft in P5?
Q2(r) /1	Wat is het grootste getal dat precies 10 voorstellingen heeft in P5?

Nieuwe notatie: lijst van een piramide.

Piramides tekenen is omslachtig en weinig praktisch.

In plaats van een piramide te tekenen geeft professor Zarbi liever de lijst, vertrekkend van de top, van het aantal zwarte schijven in elke rij.

Vanaf nu is het **getal van een piramide** het getal dat door de piramide wordt voorgesteld.

Het is een positief geheel getal, op de gebruikelijke manier geschreven met de cijfers 0 tot 9.

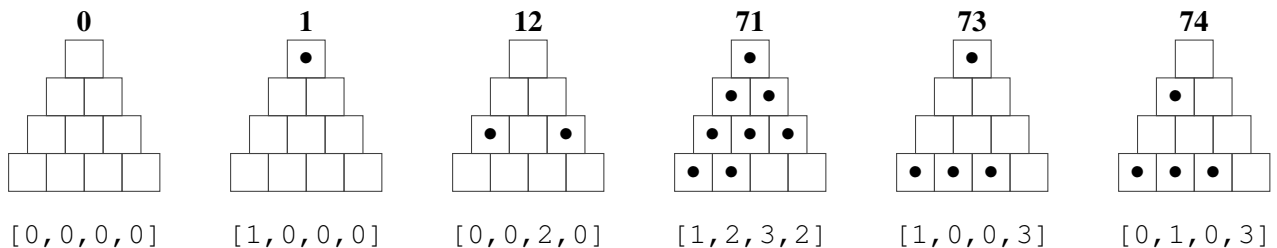
Vanaf nu is de **lijst van een piramide** de lijst van het aantal zwarte schijven in elke rij van de piramide.

Deze lijst begint met 0 als er geen zwarte schijf aan de top is of met 1 als er wel een is.

Deze lijst bevat n getallen voor een **P_n**-piramide.

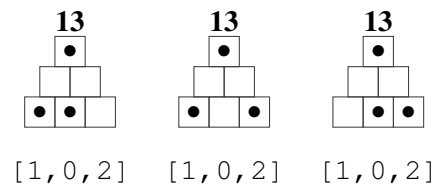
Voorbeelden: hier zijn enkele **P₄**-piramides.

Voor elk ervan staat het **getal** erboven en de **lijst** eronder.

**Opmerking.**

Als meerdere piramides hetzelfde getal voorstellen, dan hebben ze dezelfde lijst.

Voorbeeld: de 3 **P₃**-piramides die het getal 13 voorstellen hebben dezelfde lijst [1, 0, 2].

**Automatisering.**

De professor moet vaak het getal van een piramide berekenen op basis van de lijst.

Het komt ook voor dat hij omgekeerd de lijst van een piramide berekent op basis van het getal.

Hij wil programma's schrijven om zijn berekeningen te automatiseren.

Enkele herinneringen.

- Gehele deling.**

De operator $/$ (of $//$ als je dat liever hebt) geeft het gehele quotiënt van een deling tussen 2 gehele getallen.

De operator $\%$ geeft de rest van een deling tussen 2 gehele getallen.

Voorbeeld: $14/3$ is gelijk aan 4 (net als $14//3$) en $14\%3$ is gelijk aan 2.

- Nummering van de elementen van een lijst.**

De elementen van een lijst worden genummerd vanaf 0.

Voorbeeld: als $LIS = [1, 0, 2]$ dan is $LIS[0] = 1$, $LIS[1] = 0$ en $LIS[2] = 2$.

- Initialisatie van een lijst.**

$LIS \leftarrow [0] * n$ creëert een lijst van n nullen.

Functie PyrLIS (n, dec)

Deze functie ontvangt 2 parameters.

- n: het aantal rijen van de piramide.
- dec: een positief geheel getal dat door een piramide moet worden voorgesteld.
dec is zo genoemd ter herinnering dat het om een **decimaal** getal gaat (dat wil zeggen geschreven met de 10 cijfers van 0 tot 9).

De functie geeft de lijst terug van een **Pn**-piramide die het getal dec voorstelt.

Voorbeelden: PyrLIS (3, 13) geeft [1, 0, 2] terug en PyrLIS (4, 13) geeft [1, 0, 2, 0] terug.

De volgende vraag levert tussen 0 en 6 punten op.

Je verliest 3 punten voor elke zone die niet correct is ingevuld.

Q2(s) /6	Vul de in de functie PyrLIS (dec, n) aan.
-----------------	---

```
function PyrLIS(n, dec) {
  LIS ← [0]*n
  for r ← 0 to n-1 step 1 {
    LIS[r] ← 

    dec ← 
  }
  return (LIS)
}
```

Functie Pyrdec (n, LIS)

Deze functie ontvangt 2 parameters.

- n: het aantal rijen van de piramide.
- LIS: de lijst van een piramide.

De functie geeft het getal dec terug van de **Pn**-piramide waarvan de lijst is gegeven.

Voorbeelden: Pyrdec (3, [1, 0, 2]) en Pyrdec (4, [1, 0, 2, 0]) geven 13 terug.

De volgende vraag levert tussen 0 en 6 punten op.

Je verliest 3 punten voor elke zone die niet correct is ingevuld.

Q2(t) /6	Vul de in de functie Pyrdec (LIS, n) aan.
-----------------	---

```
function Pyrdec(n, LIS) {
  dec ← 0
  v ← 1
  for r ← 0 to n-1 step 1 {
    v ← v * 

    dec ← 
  }
  return (dec)
}
```

Vraag 3 – Het verloren paleis aflevering 3: optellen.

Professor Zarbi heeft alle plattegronden ontcijferd waarop kleine schijfjespiramides op de tegels getekend stonden. Hij heeft de plattegronden overgeschreven door elke kleine piramide te vervangen door het getal dat ze voorstelt. Na lang nadenken en zorgvuldige experimenten heeft hij eindelijk de oplossing gevonden...

Opgelet! Verwar ze niet!

In aflevering 3 bestaan de piramides opnieuw uit tegels, zoals in aflevering 1. Maar in deze aflevering draagt elke tegel een getal (het getal dat in aflevering 2 ontcijferd werd).

Paden met maximale som.

Vanaf nu is een *getallenpiramide* een piramide met een getal op elke tegel.

Ter herinnering, een *pad*

- begint bij de top,
- gaat van een tegel naar een andere die hem raakt,
- gaat over precies één tegel in elke rij,
- moet de laatste rij bereiken.

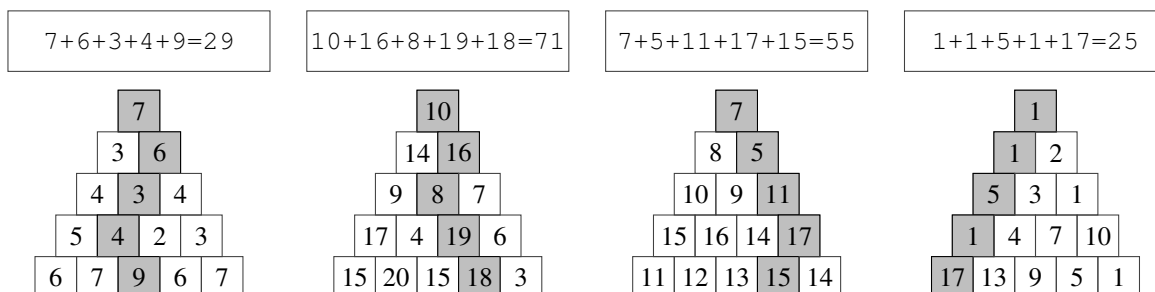
De *score* van een pad is de som van de getallen op de tegels van dat pad.

Professor Zarbi heeft ontdekt dat het pad met de grootste score geen val activeert.

Als meerdere paden dezelfde maximale score hebben, activeert geen van hen een val.

Elk pad met een score lager dan de maximale score activeert een val.

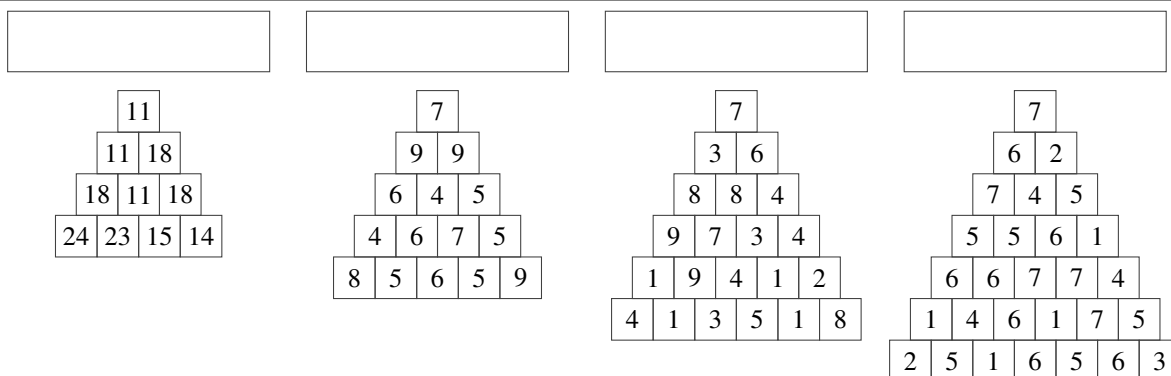
Voorbeelden: hier zijn enkele piramides **P5**.
Voor elk is een pad met maximale score grijs gekleurd.
De score van het pad staat in de rechthoek.



In de volgende vraag zijn piramides getekend met een getal op elke tegel. Teken in elke piramide een pad met maximale score (door de getallen te omcirkelen of de tegels met potlood in te kleuren).

Schrijf de score van het getekende pad in de rechthoek boven de piramide.

Q3(a) /8	Teken een pad met maximale score en noteer de score in de rechthoek.
-----------------	---



Automatisering

Professor Zarbi wil de berekening van de maximale score van een getallenpiramide automatiseren.

In zijn programma's wordt een piramide **P_n** voorgesteld door een lijst **P** van **n** deellijsten genummerd van 0 tot **n-1**.

Elke deellijst bevat de getallen van een rij.

De eerste deellijst **P[0]** bevat het getal aan de top van de piramide.

De laatste deellijst **P[n-1]** bevat de **n** getallen van de laatste rij.

Voorbeeld

De piramide **P₅** hiernaast wordt weergegeven met de rijnummers links.

Voor deze piramide:

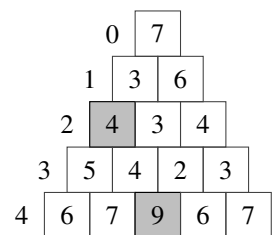
$n=5$

$P = [[7], [3, 6], [4, 3, 4], [5, 4, 2, 3], [6, 7, 9, 6, 7]]$

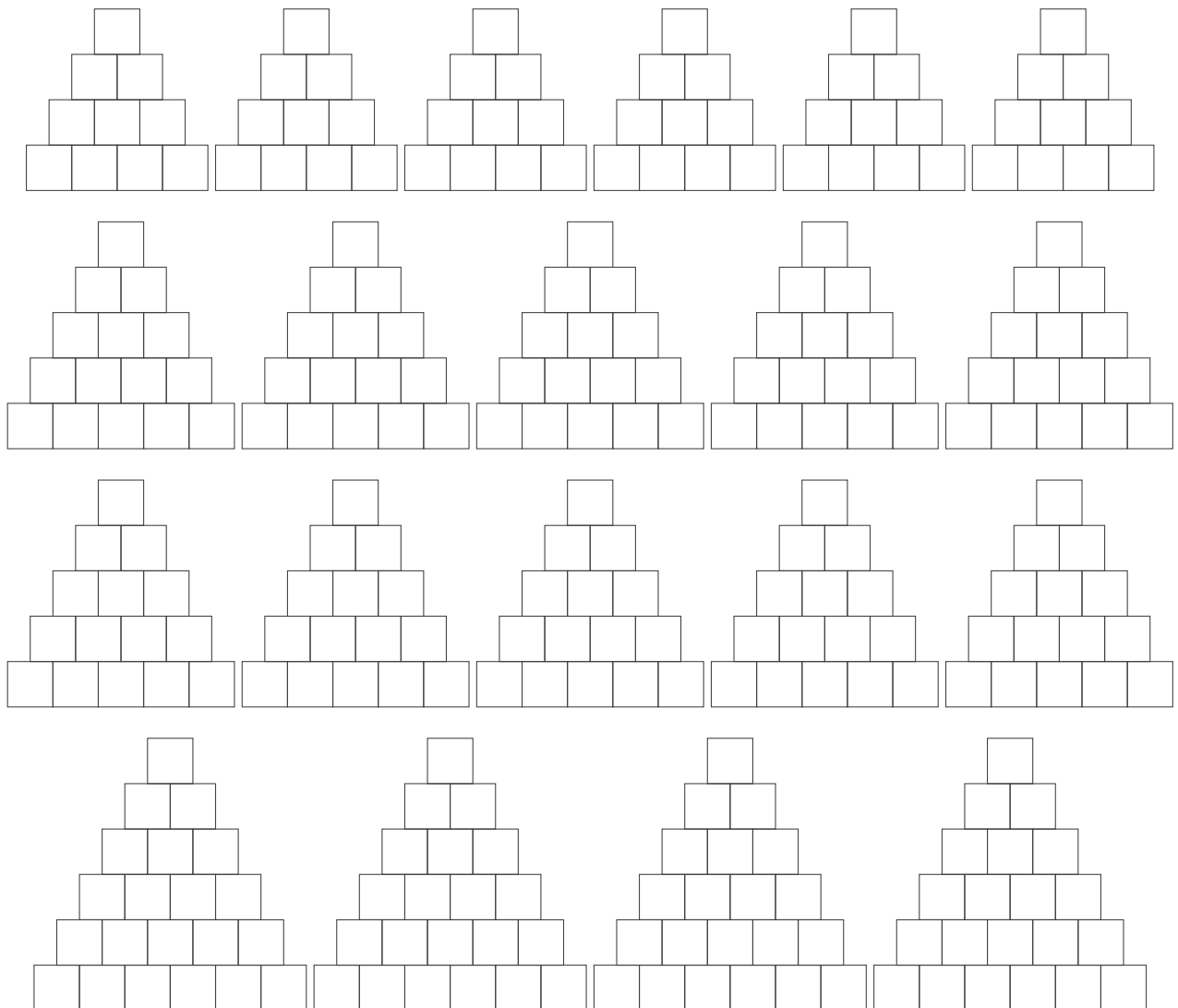
$P[0] = [7]$

$P[4] = [6, 7, 9, 6, 7]$

$P[2][0] = 4$ en $P[4][2] = 9$ zijn grijs gekleurd.



Piramides voor je kladwerk.



Functie MaxScore (n, P)

Deze functie gebruikt 2 parameters.

- n: het aantal rijen van de piramide.
- P: een lijst van lijsten van getallen die de piramide voorstelt, zoals hierboven uitgelegd.

De functie geeft de maximale score terug die mogelijk is voor een pad in deze piramide.

Voorbeeld: MaxScore(5, [[7], [3, 6], [4, 3, 4], [5, 4, 2, 3], [6, 7, 9, 6, 7]]) geeft 29 terug (zie het eerste voorbeeld op de vorige pagina).

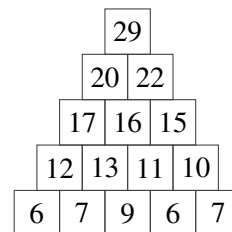
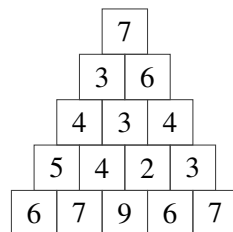
Voor elke tegel berekent het programma een score gelijk aan de maximale score van een pad in de deelpiramide met die tegel als top.

Je kunt de score van een tegel berekenen met behulp van de scores van de tegels die hij raakt in de volgende rij.

De score van een tegel vervangt in P het getal dat op de tegel staat.

Voorbeeld

Tijdens de verwerking van de linkerpiramide
vervangt de functie de waarden in P door die in de rechterpiramide.



Voor de uitvoering: P = [[7], [3, 6], [4, 3, 4], [5, 4, 2, 3], [6, 7, 9, 6, 7]]

Na de uitvoering: P = [[29], [20, 22], [17, 16, 15], [12, 13, 11, 10], [6, 7, 9, 6, 7]]

De volgende vraag levert tussen 0 en 10 punten op.

Je verliest 2 punten voor elke zone die niet correct is ingevuld. Er zijn geen negatieve punten.

Q3(b) /10 Vul de in de functie MaxScore (n, P) aan.

```
function MaxScore(n,P) {
  for r ←  to  step 
  {
    for c ← 0 to r step 1
    {
      if  > 
      {
        {P[r][c] ← }
      }
      else
      {
        {P[r][c] ← }
      }
    }
  }
  return 
}
```

Functie GoodPath (n, P)

Wat er echt toe doet, is een pad met maximale score vinden om geen vallen te activeren.

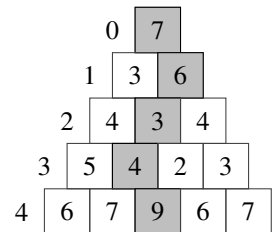
Een pad wordt voorgesteld door een lijst die voor elke rij het nummer bevat van de tegel waarover het pad in die rij loopt.

Voorbeeld

De piramide **P5** hiernaast wordt weergegeven met de rijnummers links.
Het grijs gekleurde pad in de piramide wordt beschreven door de lijst GP van
tegelnnummers in hun rij.

In dit voorbeeld: GP = [0, 1, 1, 1, 2]

Het pad gaat immers over tegel 0 van rij 0,
tegel 1 van rijen 1, 2 en 3
en ten slotte over tegel 2 van rij 4.



De functie GoodPath (n, P) gebruikt dezelfde parameters als de functie MaxScore (n, P).

Ze geeft een lijst terug die een pad met maximale score in P beschrijft.

Het volstaat om de **return** van de functie MaxScore (n, P) te vervangen door enkele regels code.

Deze vraag levert tussen 0 en 6 punten op.

Je verliest 2 punten voor elke zone die niet correct is ingevuld. Er zijn geen negatieve punten.

Ter herinnering: GP ← [0] * n maakt een lijst van n nullen.

Q3(c) /6**Vul de aan in de code van GoodPath die return in MaxScore vervangt.**

```

c ← 0
GP ← [0] * n
for r ← 1 to  step 1
{
    if 

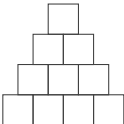
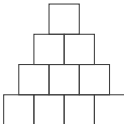
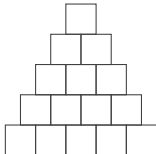
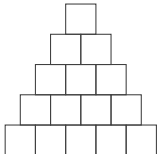
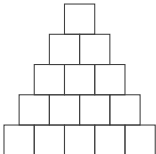
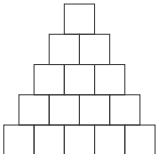
        { c ←  }

    GP [] ← 
}
return GP

```

Vul hier je antwoorden in !

Q1(a) Hoeveel paden bereiken de grijze tegel in P4? 	/4
Q1(b) Hoeveel paden bereiken de grijze tegel in P5? 	/5
Q1(c) Hoeveel paden bereiken de grijze tegel in P6? 	/6
Q1(d) In totaal, hoeveel paden doorkruisen P2?	/1
Q1(e) In totaal, hoeveel paden doorkruisen P4?	/1
Q1(f) In totaal, hoeveel paden doorkruisen P5?	/1
Q1(g) In totaal, hoeveel paden doorkruisen P6?	/1
Q1(h) In totaal, hoeveel paden doorkruisen Pn (waarbij n een positief geheel getal is)?	/1
Q1(i) Hoeveel paden gaan door de grijze tegels van de piramides P6? 	/6
Q1(j) Hoeveel paden gaan door de grijze tegels van de diamanten? 	/8
Q1(k) In totaal, hoeveel paden doorkruisen deze vreemde zalen? 	/6
Q1(l) Uitdrukking voor het aantal paden door de grijze tegels van P10 uit het voorbeeld.	/2
Q1(m) Uitdrukking voor het aantal paden door de grijze tegels van P12.	/2
Q1(n) Uitdrukking voor het aantal paden door de grijze tegels van P13.	/2

Q1(o) Uitdrukking voor het aantal paden door de grijze tegels van P15.	/2
Q1(p) Vul de _____ in de functie countPyraV1 (n, G) aan. <pre> function countPyraV1 (n, G) { nbr ← 1 (r0, c0) ← (0, 0) for (r, c) in G { h ← _____ k ← _____ nbr ← _____ (r0, c0) ← (r, c) } if (r0 < _____) { nbr ← _____ } return nbr }</pre>	/6
Q1(q) Vul de _____ in de test van countPyraV2 die een regel van countPyraV1 vervangt. <pre> if (_____ and _____) {nbr ← _____} else {nbr ← _____}</pre>	/6
Q2(a) Schrijf in elke rechthoek het getal dat door de getekende piramide wordt voorgesteld. 	/6
Q2(b) Teken een piramide die het aangegeven getal voorstelt. 30  60  120  365  421  599 	/6
Q2(c) Hoeveel verschillende P3-piramides kun je tekenen?	/1
Q2(d) Hoeveel verschillende P4-piramides kun je tekenen?	/1

Q2(e) Wat is het grootste getal dat je kunt voorstellen met een P4-piramide?	/1
Q2(f) Hoeveel verschillende P5-piramides kun je tekenen?	/1
Q2(g) Wat is het grootste getal dat je kunt voorstellen met een P5-piramide?	/1
Q2(h) Hoeveel verschillende P6-piramides kun je tekenen?	/1
Q2(i) Wat is het grootste getal dat je kunt voorstellen met een P6-piramide?	/1
Q2(j) Welke getallen hebben precies 1 voorstelling in P3?	/1
Q2(k) Welke getallen hebben precies 2 voorstellingen in P3?	/1
Q2(l) Welke getallen hebben precies 3 voorstellingen in P3?	/1
Q2(m) Welke getallen hebben precies 4 voorstellingen in P3?	/1
Q2(n) Welke getallen hebben precies 5 voorstellingen in P3?	/1
Q2(o) Welke getallen hebben precies 6 voorstellingen in P3?	/1
Q2(p) Welke getallen hebben precies 36 voorstellingen in P4?	/1
Q2(q) Wat is het kleinste getal dat precies 10 voorstellingen heeft in P5?	/1
Q2(r) Wat is het grootste getal dat precies 10 voorstellingen heeft in P5?	/1

Q2(s) Vul de _____ in de functie PyrLIS (dec, n) aan.

/6

```
function PyrLIS(n,dec) {
  LIS ← [0]*n
  for r ← 0 to n-1 step 1 {
    LIS[r] ← _____

    dec ← _____
  }
  return (LIS)
}
```

Q2(t) Vul de _____ in de functie Pyrdec (LIS, n) aan.

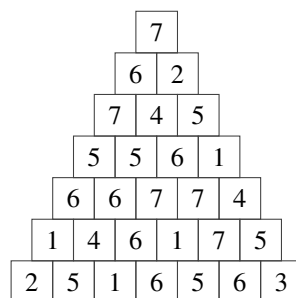
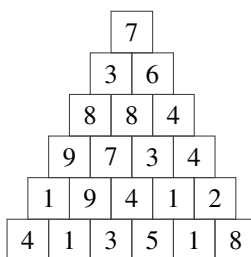
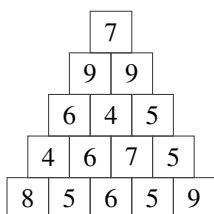
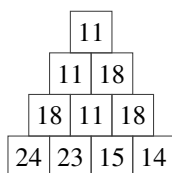
/6

```
function Pyrdec(n,LIS) {
  dec ← 0
  v ← 1
  for r ← 0 to n-1 step 1 {
    v ← v * _____

    dec ← _____
  }
  return (dec)
}
```

Q3(a) Teken een pad met maximale score en noteer de score in de rechthoek.

/8



Q3(b) Vul de _____ in de functie `MaxScore(n,P)` aan.

/10

```
function MaxScore(n,P) {  
  for r ← _____ to _____ step _____  
  {  
    for c ← 0 to r step 1  
    {  
      if _____ > _____  
  
        {P[r][c] ← _____}  
      else  
  
        {P[r][c] ← _____}  
    }  
  }  
  return _____  
}
```

Q3(c) Vul de _____ aan in de code van `GoodPath` die `return` in `MaxScore` vervangt.

/6

```
c ← 0  
GP ← [0]*n  
for r ← 1 to _____ step 1  
{  
  if _____  
  
    { c ← _____ }  
  
  GP[_____] ← _____  
}  
return GP
```