

# Toetspracticum Imperatief Programmeren

2 november 2010

- Je kunt de opgaven in willekeurige volgorde maken.
- De eerste opgave is 1 punt waard, de overige vier opgaven elk 2 punten. De eerste punt krijg je gratis.
- Er wordt voor de beoordeling uitsluitend gekeken of je programma door Justitia is goedgekeurd. Er wordt dus niet naar je code gekeken, noch wordt er een verslag gevraagd.
- Als je programma voor een test in Justitia faalt, dan staat er een hint bij de test die fout gaat.
- Hopelijk overbodig om te vermelden: het is niet de bedoeling dat er wordt 'samengewerkt'. Als plagiaat geconstateerd wordt zullen zowel de kopiërende partij als de partij die gelegenheid bood tot kopiëren zich moeten verantwoorden.
- Het is toegestaan om tijdens het toetspracticum gebruik te maken van het dictaat, het boek en eventueel het internet (bijv. Nestor). Het is niet toegestaan om gebruik te maken van e-mail. Ook is het niet toegestaan gebruik te maken van laptops en mobiele telefoons.
- Bij iedere opgave is een voorbeeld invoer file `vb.in` gegeven en de bijbehorende uitvoerfile `vb.out`. Deze kun je gebruiken om zelf te testen of de uitvoer van jouw programma overeenkomt met de gevraagde layout, zodat er geen misverstand kan bestaan over spaties in de uitvoer. Voor iedere opgave kun je de voorbeeldfiles vinden bij de betreffende opgave in Justitia.
- Uiteraard mag je er vanuit gaan dat de invoer bij iedere opgave correct is. Bijvoorbeeld, als de invoer een positief getal behoort te zijn, dan hoeft je niet te testen of de invoer misschien toch negatief is.

## Opgave 1: Eindcijfer

Het eindcijfer voor het vak *Imperatief Programmeren* wordt bepaald uit het cijfer voor het tentamen (noem dit cijfer T), het cijfer voor het toetspracticum (noem dit cijfer TP), en het cijfer voor het practicum (noem dit cijfer P). Voor de bepaling van het eindcijfer voor het vak gelden de volgende regels:

- Als het tentamencijfer T lager dan een 5 is (dus ook een 4.99), dan is het eindcijfer gelijk aan de afronding van het tentamencijfer volgens de onderstaande afrondregels.
- Als het tentamencijfer minstens een 5 is, dan wordt het eindcijfer E bepaald volgens de formule  $E = (2 * T + TP + P) / 4$ . Hierbij is op zijn beurt het cijfer P het gemiddelde van de 6 practicumssessies P1, P2, ..., P6 (dus  $P = (P1 + P2 + P3 + P4 + P5 + P6) / 6$ ) en TP het maximum van de resultaten van het toetspracticum en de herkansingstoets. Bij de berekening van P vindt geen afronding plaats.
- Het eindcijfer wordt afgerond op helen en halven (bijv. 4.24 wordt een 4, terwijl 4.25 een 4.5 wordt). Nu is er nog één merkwaardigheid: aan de Faculteit Wiskunde- en Natuurwetenschappen wordt het cijfer 5.5 niet gegeven. Bij dit vak ronden we daarom ieder cijfer tussen de 5 en de 5.5 (dus ook 5.499) af naar een 5, terwijl een cijfer hoger of gelijk aan een 5.5 en lager dan een 6 wordt afgerond naar een 6.

Gevraagd wordt een programma dat het eindcijfer van een student afdruckt, gegeven de deelresultaten. De uitvoer moet precies één decimaal achter de komma hebben (dus 7.0 en niet 7). De invoer bestaat uit drie regels. Op de eerste regel staat het tentamencijfer. Op de volgende regel staan de 6 resultaten voor de practica (P1, ..., P6). Op de laatste regel staan de resultaten voor respectievelijk het toetspracticum en de herkansingstoets.

[Hint: als n van het type float is, dan drukt het statement `printf("%.1f\n", n)`; dit getal met één decimaal achter de komma af.]

### Voorbeeld 1:

**invoer:**

7.3

10 8 7.5 8.5 8 6

4 8

**uitvoer:**

7.5

### Voorbeeld 2:

**invoer:**

5

6 6 6 6 6 6

4 6

**uitvoer:**

6.0

## Opgave 2: Bevriend

De oude Pythagoreërs deden onderzoek naar 'bevriende' getallen. Twee getallen zijn bevriend als het ene gelijk is aan de som van de echte delers van het andere, en omgekeerd. Een getal  $d$  is een echte deler van een getal  $n$  als  $d$  een deler van  $n$  is met  $d < n$ . Bevriende getallen zijn bijvoorbeeld 220 en 284, want de som van de echte delers van 220 is  $1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110 = 284$ , en die van 284 is  $1 + 2 + 4 + 71 + 142 = 220$ .

Gevraagd wordt een programma te maken dat twee positieve gehele getallen inleest en daarna bevriend afdrukt als de twee getallen bevriend zijn, en niet bevriend als de twee getallen niet bevriend zijn.

### Voorbeeld 1:

invoer:

220 284

uitvoer:

bevriend

### Voorbeeld 2:

invoer:

42 21

uitvoer:

niet bevriend

## Opgave 3: Priemtweelingen

Een *priemtweeling* is een paar natuurlijke getallen  $(p, p + 2)$  waarbij  $p$  en  $p + 2$  priemgetallen zijn. Voorbeelden hiervan zijn  $(3,5)$ ,  $(5,7)$ , en  $(17,19)$ .

In deze opgave krijg je op de invoer een natuurlijk getal  $n$ . Schrijf een programma dat het aantal priemtweelingen  $(p, p+2)$  met  $p+2 \leq n$  afdrukt.

### Voorbeeld 1:

invoer:

5

uitvoer:

1

### Voorbeeld 2:

invoer:

100

uitvoer:

8

## Opgave 4: Stableford

In de golfsport is er een scoresysteem uitgevonden waarbij het mogelijk is om gevorderde spelers en beginnende spelers tegen elkaar te laten spelen zonder dat de gevorderde speler altijd wint. Dit systeem heet *Stableford scoring*.

Een volwaardige golfbaan bestaat uit 18 holes. Van iedere hole is de zogenaamde *par* bekend: het aantal slagen waarin een gemiddelde professionele golfer de hole zou moeten kunnen spelen.

Iedere amateurspeler heeft een zogenaamde *handicap* (een geheel getal). De minimale handicap is 0 en de maximale handicap is 36. Deze handicap geeft het totaal aantal slagen die een amateurspeler extra krijgt ten opzichte van een professional. Deze extra slagen moeten verdeeld worden over de 18 holes waaruit een golfbaan bestaat. De slagen worden toegewezen aan holes in volgorde van moeilijkheidsgraad, de zogenaamde *index* van de hole. De moeilijkste hole heeft index 1, de makkelijkste heeft index 18. Iedere hole heeft een unieke index, m.a.w. er zijn geen twee holes met dezelfde index. Een handicap van 20 betekent dat voor iedere hole een extra slag bij de par mag worden opgeteld. De twee moeilijkste holes met stroke index 1 en 2 krijgen daar bovenop nog een extra slag erbij (immers  $20=18+2$ ). De par van een hole plus het aantal slagen dat door de handicap wordt toegekend aan deze hole, wordt de *persoonlijke par* voor deze hole genoemd.

De Stableford-telling kent de golfer, die evenveel slagen nodig heeft om het balletje in de hole te krijgen als zijn persoonlijke par voorschrijft, twee punten toe. Voor elke slag die hij minder nodig heeft, wordt een extra punt toegekend. Als de speler één slag meer nodig heeft dan zijn persoonlijke par, dan verdient de speler nog 1 punt. Als twee extra slagen of meer boven de persoonlijke par nodig zijn voor een hole, dan kunnen er voor deze hole geen punten meer verdiend worden. De *stablefordscore* van een speler is de optelling van zijn scores voor de 18 holes.

Gevraagd wordt een programma te maken dat de scorekaart van een speler inleest en daarna de stablefordscore van deze speler afdruckt. De invoer bestaat uit 4 regels. Op de eerste regel staan 18 getallen: de par van iedere hole. Op de tweede regel staan ook 18 getallen: de index van iedere hole. Op de derde regel staat de handicap van de speler. Op de laatste regel staan opnieuw 18 getallen: het aantal slagen dat de speler per hole nodig heeft gehad.

**Voorbeeld:**

**invoer:**

```
4 4 4 5 4 3 5 3 4 4 4 4 5 4 3 5 3 4
2 4 11 9 13 18 7 15 5 1 3 12 10 14 17 8 16 6
30
6 8 6 7 6 4 7 5 6 6 6 7 7 6 4 7 5 6
```

**uitvoer:**

29

## Opgave 5: Magisch vierkant

Een *magisch vierkant* is een vierkante tabel van  $n \times n$  velden. In ieder veld is een getal ingevuld. De getallen zijn zodanig ingevuld dat de kolommen, de rijen en de beide diagonalen alle dezelfde som opleveren. Tevens is vereist dat het vierkant alle natuurlijke getallen van 1 tot en met  $n^2$  bevat.

Een voorbeeld van een magisch vierkant met  $n = 3$  is:

8	3	4
1	5	9
6	7	2

Gevraagd wordt een programma te maken dat een vierkant met getallen inleest en bepaalt of de invoer wel of niet een magisch vierkant is. Op de eerste regel van de invoer staat het getal  $n$ . Je mag er vanuit gaan dat  $n < 100$ . Daarna volgen er  $n$  regels met ieder  $n$  getallen: de elementen van het

vierkant. Als de invoer een magisch vierkant is, dan dient het programma de tekst ja af te drukken.  
Zo niet, dan dient de uitvoer nee te zijn.

**Voorbeeld 1:**

**invoer:**

3

8 3 4

1 5 9

6 7 2

**uitvoer:**

ja

**Voorbeeld 2:**

**invoer:**

3

8 3 4

6 7 2

1 5 9

**uitvoer:**

nee