

Faculteit Wiskunde en Informatica

Toets Inleiding Programmeren

Vrije Universiteit

27 oktober 1997

tijdsduur : 1:30 uur

Opgave 1.

- a) Machinetaal (machine language) en assembleertaal (assembly language) zijn programmeertalen met de gemeenschappelijke eigenschap dat deze maar voor 1 processor (processing unit) ontworpen zijn.

Wat is het verschil in notatie van de opdrachten (statements) in deze programmeertalen?

- b) Wanneer wordt een programmeertaal een hogere programmeertaal (high-level programming language) genoemd?
- c) Wat is het verschil tussen de manier waarop een vertaler (compiler) en een interpretator (interpreter) een programma geschreven in een hogere programmeertaal (high-level language) uitvoeren?

Opgave 2.

a) Geef van de onderstaande arithmetische expressies de uitkomst

- | | | | |
|----|---------------|-----|---------------|
| 1) | $2 + 3 * 4$ | 2) | $2 * 3 + 4$ |
| 3) | $(2 + 3) * 4$ | 4) | $2 + (3 * 4)$ |
| 5) | $3 / 2$ | 6) | $3 \% 2$ |
| 7) | $3. / 2$ | 8) | $3. / 2.$ |
| 9) | $8 / 2 / 2$ | 10) | $8 / (2 / 2)$ |

b) Als de volgende declaraties van een variabele i gegeven is:

`int i = 2;`

geef dan van de onderstaande expressies de uitkomst

- | | | | |
|----|--------------------------------|-----|------------------------------------|
| 1) | <code>i == 3</code> | 2) | <code>i = 3</code> |
| 3) | <code>2 * 2 != 2 + 2</code> | 4) | <code>2 * 2 >= 2 + 2</code> |
| 5) | <code>2 * 2 <= 2 + 2</code> | 6) | <code>2 * 2 == 2 + 2</code> |
| 7) | <code>"tekst"</code> | 8) | <code>"tekst" + "voorbeeld"</code> |
| 9) | <code>(double) 3</code> | 10) | <code>(double) 3.0</code> |

c) Voor een hier niet nader gespecificeerd probleem is het noodzakelijk om een in een class te beschikken over de constante pi ($\pi = 3.14$), de variabele straal (heeft als waarde een reeel getal) en de variabele eenheidscirkel (geeft aan of de cirkel waarvan de straal onthouden wordt wel of niet de eenheidscirkel is).

Declareerd pi, straal en eenheidscirkel. Kies bij alldrie het juiste primitieve type.

Opgave 3.

- a) Gegeven zijn de volgende declaraties:

```
Input in = new Input();  
int i = in.readInt();
```

Schrijf een algoritme dat i met 1 ophoogt als i even is.

(HINT: een getal is even als het deelbaar is door 2)

♀

- b) Gegeven zijn de volgende declaraties:

```
Input in = new Input();  
int i = in.readInt();
```

Schrijf een algoritme dat de tekst "negatief", "nul" of "positief"

afdrukt, afhankelijk van het feit of de waarde van i respectievelijk negatief, nul of positief is.

- c) Gegeven zijn de volgende declaraties:

```
1997-10-27 Tentamen.txt
Input in = new Input();

int n = in.readInt(),          // n >= 0

    faculteit;
```

Schrijf een algoritme voor de berekening van de kleinste faculteit die groter of gelijk is aan n.

$(y! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot (y-1) \cdot y$ voor $y > 0$ en $0! = 1$)

Het resultaat moet aan de variabele faculteit geassigneerd worden.

- d) Een Syracuse rij is een rij natuurlijke getallen, die met een willekeurig natuurlijk getal kan beginnen, en waarin ieder getal, behalve het eerste getal, uit het voorgaande getal berekend wordt. Als het n-de getal ($n \geq 1$) uit een Syracuse rij even is, is het (n+1)-ste getal de helft van het n-de getal. Als het n-de getal oneven is, is het (n+1)-ste getal gelijk aan 3 maal het n-de getal plus 1. Een voorbeeld:

Een stuk van de Syracuse rij die met 3 begint:

3, 10, 5, 16, 8, 4, 2, 1, 4, 2, 1, 4, 2, 1, 4, 2, 1,

Het is gebruikelijk bij Syracuse rijen om te stoppen met het noteren van getallen nadat het getal 1 genoteerd is, omdat na het getal 1 de drie getallen 4, 2, 1 eindeloos herhaald worden. Het vermoeden bestaat,

maar dit vermoeden is tot nu toe niet bewezen, dat iedere Syracuse rij, die met een natuurlijk getal ongelijk aan nul begint, uiteindelijk op 1 zal eindigen.

Gegeven zijn de volgende declaraties:

```
Input in = new Input();
int start = in.readInt(),
```

Schrijf een algoritme dat de de getallen van de Syracuse rij die met start begint afdruckt t/m de eerste keer dat er een 1 in de rij voorkomt.

Waardering

Opgave	a	b	c	d	totaal
1.	3	3	3		9
2.	4	4	4	12	
3.	3	4	4	15	
					-- +
					36

Het eindcijfer E volgt uit het puntentotaal T als volgt :

$$E = T / 4 + 1$$