

Deze opdrachten zijn bedoeld om je vertrouwd te maken met het gebruik van *Lazarus*. Ze gaan er vanuit dat je beschikt over

- [1] een laptop met Lazarus 0.9.24 of hoger
- [2] ‘Getting Started with Lazarus at TU/e’
- [3] ‘Samenvatting elementair (Object) Pascal’
- [4] ‘Programmeren in Pascal: Voorbeelden’
- [5] de mappen met bestanden uit `Reeks1.zip`

1. (a) In map `GetalKlutser` staat Lazarus project `GetalKlutser.lpi`. Open dit door dubbelklikken. Lazarus start op.
 - (b) Executeer de `GetalKlutser` en probeer wat invoerwaarden. Wat valt er zoal op? Wat zou het klutsresultaat van 300 zijn als die niet ‘geblokkeerd’ was?
 - (c) Zoek in [4] de betekenis van de operaties **mod** en **div** op.
 - (d) Bepaal de waarde van de uitdrukkingen
 - i. $6 \bmod 2$
 - ii. $6 \operatorname{div} 2$
 - iii. $13 \bmod 2$
 - iv. $13 \operatorname{div} 2$
 Vul ze in het gegeven sjabloon `GetalKlutser.txt` in.
 - (e) Introduceer een integer hulpvariabele m en vervang de opdracht

$$r := G * r + (h \bmod G)$$
 door

$$m := h \bmod G$$

$$; r := G * r + m$$
 opdat $h \bmod G$ inspecteerbaar wordt.
 - (f) Tabelleer het waardenverloop van de variabelen n , h , m en r wanneer de `GetalKlutser` invoerwaarde 13 verwerkt. Maak één kolom per variabele en één rij per programmastap waarin

een variabele verandert. Noteer bij onveranderde waarde een puntje. Gebruik hiervoor weer het gegeven sjabloon `GetalKlutser.txt`. De tabel begint met

n	h	m	r
?	?	?	?
13	.	.	.
.	13	.	.
.	.	.	0
.	.	1	.

Benut hierbij de Lazarus traceervoorzieningen (Breakpoint direct na `readln(n)`, plaats variabelen n, h, m, r in de Watch List, en ga verder met Step Over; zie ook [2]).

- (g) Experimenteer ook met **const** $G = 10$. Wat is hiervan het effect?
 - (h) Beschrijf kort *wat* de `GetalKlutser` bewerkstelligt, d.w.z. wat het verband is tussen in- en uitvoer, zonder te vervalten in de details van *hoe* dat gebeurt.
 - (i) Registreer uzelf bij `peach3` op <https://peach.win.tue.nl/> en koppel uzelf aan de juiste groep.
 - (j) Stuur je antwoorden (in tekstbestand `GetalKlutser.txt`) voor onderdelen (d), (f), (g) en optioneel (h) in bij `peach3` voor de opgave `GetalKlutser`.
-
2. (a) Lees de opgave `Snoep` in `peach3`.
 - (b) Open project `Snoep.lpi` in map `Snoep`. Lees de programmatekst door. Er zitten een aantal fouten in.
 - (c) Zend `Snoep.lpr` in bij `peach3` en verbeter de gevonden fouten. Zend het dan opnieuw in. Herhaal dit tot het programma (redelijk) goed werkt.
 - (d) Pas het programma zó aan dat het een eigen oplossing voor opgave `Snoep` vormt, zonder gebruik van **while**. Dit is een

drastische wijziging. Lees eerst de opgave nog eens goed door. Denk aan de operatoren **mod** (rest na deling) en **div** (gehele deling zonder rest). Vul als commentaar ook auteur, id.nummer, datum en een korte omschrijving in.

- (e) Zend je eigen goede oplossing `Snoep.lpr` in bij `peach3`.

3. De **huiswerkopgave Mediaan** staat in `peach3` vermeld. Deze wordt nagekeken en beoordeeld. Maak een nieuwe map met een nieuw project:

Open `TUEConsoleApp.lpi` in TUE Console Application en creëer via 'Project>Save Project As ...', in een nieuwe map (bijv. genaamd Reeks 1 Mediaan), een nieuw project genaamd `Mediaan.lpi`.

Zend van je oplossing het bestand `Mediaan.lpr` bij `peach3` in binnen de afgesproken termijn.

Samenvatting van de opgave: Bepaal van drie verschillende getallen a, b, c de middelste in grootte.

Hier zijn twee ontwerpideeën:

- (a) Bepaal voor ieder mogelijk antwoord ($x = a, b, c$) de conditie waaronder het optreedt en gebruik die in een (of meer) **if**-opdracht(en) met `writeln (x)`.
- (b) Merk op dat met een hulpvariabele h de volgende opdrachten de waardes van a en b verwisselen:

```

h := a
; a := b
; b := h

```

Sorteer hiermee de waardes a, b, c en schrijf b .

Geef duidelijk in het commentaar aan welk ontwerp je volgt.

Laat je eventueel ook inspireren door het programma `Maximum.dpr` in de map `Maximum`.

Let op dat je programma alleen drie getallen leest, één getal schrijft, en geen andere in- of uitvoer doet. Een slotregel met 'Tik <return> ...' (als in de TUE Console Application) wordt wel geaccepteerd.

Algemene Spelregels

1. Je vraagt tijdig assistentie als je vastloopt.
(Dus niet eindeloos zelf blijven modderen.)
2. Je gebruikt alleen behandelde Pascal taalconstructies.
(Dus nog geen 'loops', nog geen arrays, etc.)
3. Je zorgt voor (deze lijst wordt nog uitgebreid):
 - Functionele correctheid
 - Systematische opmaak (inspringen, etc.)
 - Adequaat commentaar, m.n. in aanhef (vermeld auteur, id.nummer, datum en doel van het programma), bij declaratie van variabelen en bij belangrijke stappen in het programma
4. Je controleert *zelf* je werk vóór inzenden.
(Dus niet klakkeloos inzenden en `peach3` als 'ontwikkelomgeving' misbruiken.)
De enige uitzondering hierop vormt opgave 2 (Snoep).
5. Je zendt je *eigen* werk in.
(Dus niet van een ander kopiëren.)