Exercises for which solution notes are available

Exercise 1
Write a specification which is true if and only if the following program terminates.

```
WHILE X>1 DO IF ODD(X) THEN X := (3×X)+1 ELSE X := X DIV 2
```

Exercise 2
Let C be the following command

R:=X;
Q:=0;
WHILE Y≤R DO (R:=R-Y; Q:=Q+1)

Find a condition P such that $[P]$ C $[\mathtt{R} < \mathtt{Y} \wedge \mathtt{X} = \mathtt{R} + (\mathtt{Y} \times \mathtt{Q})]$ is true.

Exercise 3
When is $[\mathtt{T}]$ C $[\mathtt{T}]$ true?

Exercise 4
Write a partial correctness specification which is true if and only if the command C has the effect of multiplying the values of $\mathtt{X}$ and $\mathtt{Y}$ and storing the result in $\mathtt{X}$.

Exercise 5
Write a specification which is true if the execution of C always halts when execution is started in a state satisfying P.

Exercise 6
Find the flaw in the 'proof' of $1 = -1$ below:

1.	$\sqrt{-1 \times -1}$	$= \sqrt{-1 \times -1}$	Reflexivity of =.
2.	$\sqrt{-1 \times -1}$	$= (\sqrt{-1}) \times (\sqrt{-1})$	Distributive law of $\sqrt{\ }$ over $\times$.
3.	$\sqrt{-1 \times -1}$	$= (\sqrt{-1})^2$	Definition of $()^2$.
4.	$\sqrt{-1 \times -1}$	$= -1$	definition of $\sqrt{\ }$.
5.	$\sqrt{1}$	$= -1$	As $-1 \times -1 = 1$.
6.	1	$= -1$	As $\sqrt{1} = 1$.

Exercise 7
Is the following specification true?

$\vdash$ {X=x ∧ Y=y} X:=X+Y; Y:=X-Y; X:=X-Y {Y=x ∧ X=y}

If so, prove it. If not, give the circumstances in which it fails.

Exercise 8
Show in detail that $\vdash$ {X=R+(Y×Q)} R:=R-Y; Q:=Q+1 {X=R+(Y×Q)}

Exercise 9
Give a detailed formal proof that

$\vdash$ {T} IF X≥Y THEN MAX:=X ELSE MAX:=Y {MAX=max(X,Y)}

follows from $\vdash$ X≥Y ⇒ max(X,Y)=X and $\vdash$ Y≥X ⇒ max(X,Y)=Y.

Exercise 10
Suppose we add to our little programming language commands of the form:

CASE E OF BEGIN C_1; ... ; C_n END

These are evaluated as follows:

(i) First E is evaluated to get a value x.

(ii) If x is not a number between 1 and n, then the CASE-command has no effect.

(iii) If $x = i$ where $1 \leq i \leq n$, then command C_i is executed.

Why is the following rule for CASE-commands wrong?

$$\frac{\vdash\ \{P\ \wedge\ E=1\}\ C_1\ \{Q\},\ \ \ldots\ \ ,\vdash\ \{P\ \wedge\ E=n\}\ C_n\ \{Q\}}{\vdash\ \{P\}\ \texttt{CASE}\ E\ \texttt{OF BEGIN}\ C_1;\ \ldots\ ;\ C_n\ \texttt{END}\ \{Q\}}$$

Hint: Consider the case when P is '$X = 0$', E is 'X', C_1 is 'Y:=0' and Q is '$Y = 0$'.

Exercise 11
Devise a proof rule for the CASE-commands in the previous exercise and use it to show:

$\vdash$ {1≤X ∧ X≤3} CASE X OF BEGIN Y:=X-1; Y:=X-2; Y:=X-3 END {Y=0}

Exercise 12
Devise a proof rule for a command

REPEAT *command* UNTIL *statement*

The meaning of REPEAT C UNTIL S is that C is executed and then S is tested; if the result is true, then nothing more is done, otherwise the whole REPEAT command is repeated. Thus REPEAT C UNTIL S is equivalent to C; WHILE ¬S DO C.

Additional exercises without solution notes

Exercise 13
Use your REPEAT rule to deduce:

```
⊢ {S = C+R ∧ R<Y}
   REPEAT (S:=S+1; R:=R+1) UNTIL R=Y
  {S = C+Y}
```

Exercise 14
Use your REPEAT rule to deduce:

```
⊢ {X=x ∧ Y=y}
   S:=0;
   REPEAT
     R:=0;
     REPEAT (S:=S+1; R:=R+1) UNTIL R=Y;
     X:=X-1
   UNTIL X=0
  {S = x×y}
```

Exercise 15
The exponentiation function *exp* satisfies:

$$exp(m,0) = 1$$
$$exp(m,n+1) = m \times exp(m,n)$$

Devise a command C that uses repeated multiplication to achieve the following partial correctness specification:

$$\{\texttt{X=x} \ \wedge \ \texttt{Y=y} \ \wedge \ \texttt{Y} \geq 0\} \ C \ \{\texttt{Z=}exp\texttt{(x,y)} \ \wedge \ \texttt{X=x} \ \wedge \ \texttt{Y=y}\}$$

Prove that your command C meets this specification.

Exercise 16
Assume gcd(X,Y) satisfies:

```
⊢ (X>Y) ⇒ gcd(X,Y)=gcd(X-Y,Y)
⊢ gcd(X,Y)=gcd(Y,X)
⊢ gcd(X,X)=X
```

Prove:

```
⊢ {(A>0) ∧ (B>0) ∧ (gcd(A,B)=gcd(X,Y))}
   WHILE A>B DO A:=A-B;
   WHILE B>A DO B:=B-A
  {(0<B) ∧ (B≤A) ∧ (gcd(A,B)=gcd(X,Y))}
```

Hence, or otherwise, use your rule for `REPEAT` commands to prove:

```
⊢ {A=a ∧ B=b}
   REPEAT
    WHILE A>B DO A:=A-B;
    WHILE B>A DO B:=B-A
   UNTIL A=B
  {A=B ∧ A=gcd(a,b)}
```

Exercise 17

Deduce:

```
⊢ {S = (x×y)-(X×Y)}
   WHILE ¬ODD(X) DO (Y:=2×Y; X:=X DIV 2)
  {S = (x×y)-(X×Y) ∧ ODD(X)}
```

Exercise 18

Deduce:

```
⊢ {S = (x×y)-(X×Y)}
   WHILE ¬(X=0) DO
     WHILE ¬ODD(X) DO (Y:=2×Y; X:=X DIV 2);
     S:=S+Y;
     X:=X-1
  {S = x×y}
```

Exercise 19

Deduce:

```
⊢ {X=x ∧ Y=y}
   S:=0;
   WHILE ¬(X=0) DO
    (WHILE ¬ODD(X) DO (Y:=2×Y; X:=X DIV 2);
     S:=S+Y;
     X:=X-1)
  {S = x×y}
```

Exercise 20

Using $\mathtt{P}\times\mathtt{X}^{\mathtt{N}}=x^n$ as an invariant, deduce:

```
⊢ {X=x ∧ N=n}
   P:=1;
   WHILE ¬(N=0) DO
    (IF ODD(N) THEN P:=P×X else P:=P;
     N:=N DIV 2;
     X:=X×X)
  {P = x^n}
```

Exercise 21
Prove that the command

```
Z:=0;
WHILE ¬(X=0) DO
 (IF ODD(X) THEN Z:=Z+Y ELSE Z:=Z;
  Y:=Y×2;
  X:=X DIV 2)
```

computes the product of the initial values of X and Y and leaves the result in Z.

Exercise 22
Prove that the command

```
Z:=1;
WHILE N>0 DO
 (IF ODD(N) THEN Z:=Z×X else Z:=Z;
  N:=N DIV 2;
  X:=X×X)
```

assigns x^n to Z, where x and n are the initial values of X and N respectively and we assume $n \geq 0$.

Exercise 23
What are the verification conditions for the following specification?

```
{T} IF X≥Y THEN MAX:=X ELSE MAX:=Y {MAX=max(X,Y)}
```

Are they true?

Exercise 24
What are the verification conditions for the following specification?

```
{X = R+(Y×Q)} R:=R-Y; Q:=Q+1 {X = R+(Y×Q)}
```

Are they true?

Exercise 25
What are the verification conditions generated by the following annotated specification. Are they true?

```
{X=n}
 BEGIN
   Y:=1; {Y = 1 ∧ X = n}
   WHILE X≠0 DO {Y×X! = n!}
    (Y:=Y×X; X:=X-1)
 END
{X=0 ∧ Y=n!}
```

Exercise 26

Why are the verification conditions for the annotated specification

`{T} WHILE F DO {F} X:=0 {T}`

not provable, even though $\vdash$ `{T} WHILE F DO X:=0 {T}`.

Exercise 27

Prove by induction on the structure of C that if no variable occurring in P is assigned to in C, then $\vdash \{P\}\ C\{P\}$.

Exercise 28

Devise verification conditions for commands of the form `REPEAT` C `UNTIL` S (see Exercise 12).

Exercise 29

Consider the following alternative scheme for generating VCs from annotated `WHILE`-commands (due to Silas Brown).

`WHILE`-commands

Alternative verification conditions generated from

$$\{P\}\ \texttt{WHILE}\ S\ \texttt{DO}\ \{R\}\ C\ \{Q\}$$

are

(i) $P \wedge S \Rightarrow R$

(ii) $P \wedge \neg S \Rightarrow Q$

(iii) the verification conditions generated by
$\{R\}\ C\{(Q \wedge \neg S) \vee (R \wedge S)\}$

Either justify these VCs, or find a counterexample.