# Assignment 1 CSSE3100/7100 Reasoning about Programs

A sample solution is provided below. Each red number represents 0.5 marks.

**1.**

```
{u != 0}
{!(u == 0)}                                                              (A.1) [1]
{!(u == 0) && !(u == 0)} [2]
{!(u >= 0 && u <= 0) && !(u <= 0 && u >= 0)} [3]
{!(7*u >= 0 && u >= 7*u)} && !(-7*u >= 0 && u >= -7*u)}                  (A.24) [4]
{(7*u >=0 ==> u < 7*u) && (-7*u >=0 ==> u < -7*u)}                       (A.56) [5]
{(forall y':: (y'==7*u ==> (y'>=0 ==> u < y'))) &&
        (forall y':: (y'==-7*u ==> (y'>=0 ==> u < y')))}                 (A.65) [6]
 {forall y':: (y'==7*u ==> (y'>=0 ==> u < y')) &&  (y'==-7*u ==> (y'>=0 ==> u < y'))}   (A.37) [7]
 {forall y':: ((y'>=0 && y'==7*u) ==> u < y') && ((y'>=0 && y'=-7*u) ==> u < y')}      (A.34) [8]
 {forall y' :: (y'>=0 && y'==7*u) || (y'>=0 && y'=-7*u) ==> u < y'}               (A.9, A.7) [9]
 {true && forall y' :: y' >= 0 && (y' == 7*u || y' == -7 * u) ==> u < y'} [10]
  t := Abs(7*u);
  {u < t}
```

The program is not correct.[11] To make it correct add requires u != 0 [12]

**2.**

(a) ExpK:

```
z := 1;
var i := n;
while (i > 0)
    invariant z * $a^i$ == $a^n$ && i >= 0
{
    {(i%2 != 0 ==> z * $a^i$ == $a^n$ && i >= 1) &&
            (i % 2 == 0 ==> z * z * $a^{i/2}$ == $a^n$ && i/2 >= 0} [1]
    if i % 2 != 0 {
            {z * $a^i$ == $a^n$ && i >= 1}
            {z * $a^i$ == $a^n$ && i - 1 >= 0}               ($a^{n+1}$ == a*$a^n$)
            {z * a * $a^{i-1}$ == $a^n$ && i - 1 >= 0} [2]
            z, i := z * a, i - 1;
            {z * $a^i$ == $a^n$ && i >= 0}
    } else {
            {z * z * $a^{i/2}$ == $a^n$ && i/2 >= 0} [3]
            z, i := z * z, i / 2;
            {z * $a^i$ == $a^n$ && i >= 0}
    }
    {z * $a^i$ == $a^n$ && i >= 0}
 }
 {z * $a^i$ == $a^n$ && i >= 0 && i <= 0}                    (strengthening) [4]
 {z == $a^n$}
```

Incorrect since the invariant and guard z * $a^i$ == $a^n$ && i >= 0 && i > 0 does not imply the calculated predicate i % 2 != 0 ==> z * z * $a^{i/2}$ == $a^n$ && i/2 >= 0. [5]

A counter-example is i == 2 [6] (even number for i), a == 4 and n == 2 [7] (valid counter-example) which for the invariant gives z * 16 == 16, i.e., z == 1 and for the calculated predicate gives z * z * 4 == 16, i.e., z == ±2. [8]

ExpE:

```
{true}
{1 * $a^n$ == $a^n$} [1]
z := 1;
{z * $a^n$ == $a^n$} [2]
var i, b := n, a;
{z * $b^i$ == $a^n$} [3]
while (i != 0)
    invariant z * $b^i$ == $a^n$
{
    {z * $b^i$ == $a^n$ && i != 0}                      (strengthening) [4]
    {z * $b^i$ == $a^n$}                                (A.16, A.28)
    {i% 2 == 0 || i%2 != 0 ==> z * $b^i$ == $a^n$}      (A.34) [5]
    {(i%2 != 0 ==> z * $b^i$ == $a^{n\,\&\&}$) &&       (i is of type nat)
            (i%2 == 0 ==> z * $b^i$ == $a^n$)}
    {(i%2 != 0 ==> z * $b^i$ == $a^{n\,\&\&}$ && i - 1 >= 0) &&
            (i%2 == 0 ==> z * $b^i$ == $a^n$)} [6]
    if i % 2 != 0 {
            {z * $b^i$ == $a^n$ && i - 1 >= 0}          ($b^x$*$b^y$ == $b^{x+y}$) [7]
            {z * b * $b^{i-1}$ == $a^n$ && i - 1 >= 0} [8]
            z, i := z * b, i - 1;
            {z * $b^i$ == $a^n$ && i >= 0}              (i is of type nat)
            {z * $b^i$ == $a^n$}
    } else {
            {z * $b^i$== $a^n$}                         (b*b == $b^2$ and $(b^x)^y$ == $b^{x*y}$) [9]
            {z * $(b*b)^{i/2}$ == $a^n$} [10]
            b, i := b * b, i / 2;
            {z * $b^i$ == $a^n$}
    }
}
{z * $b^i$ == $a^n$ && i == 0}                          (strengthening) [11]
{z == $a^n$}
```

The program is partially correct since the weakest precondition is true, i.e., the program works from any initial state. [12]

(b)

```
while (i != 0)
    invariant z * $b^i$ == $a^n$
    decreases i [1]
{
    {i != 0 && (i % 2 != 0 ==> ... ) && (i%2 == 0 ==> ... )}    (since i is of type nat) [2]
    {i != 0 && (i % 2 != 0 ==> ... && i >= 0) &&
            (i%2 == 0 ==> ... && i >= 0)}                (i != 0 && i >= 0 ==> i > i/2) [3]
    {i != 0 && (i%2 != 0 ==> ... && i > i - 1 && i >= 0) &&
            (i%2 == 0 ==> ... && i > i/2 && i >= 0)}            (strengthening) [4]
    {(i%2 != 0 ==> ... && i > i - 1 && i >= 0) &&
            (i%2 == 0 ==> ... && i > i/2 && i >= 0)} [5]
    ghost var d := i;
     {(i%2 != 0 ==> ... && d > i - 1 && d >= 0) &&
            (i%2 == 0 ==> ... && d > i/2 && d >=0)} [6]
    if i % 2 != 0 {
            {... && d > i - 1 && d >= 0} [7]
            z, i := z * b, i - 1;
            {... && d > i && d >= 0}
```

```
        } else {
                {... && d > i/2 && d >= 0} [8]
                b, i := b * b, i / 2;
                {... && d > i && d >= 0}
        }
    }
```

The program is totally correct since the invariant is unchanged and hence the weakest precondition remains true.