```
%-----------------------------------------------------------------------------
% Daplex examples
%-----------------------------------------------------------------------------

%-----------------------------------------------------------------------------
% Print the code and level of each course.

program p1 is
for each c in course
print(code(c), level(c));

%-----------------------------------------------------------------------------
% Print the names of all fourth year students.

program p2 is
for each s in student such that year(s) = 4
print(forename(s), surname(s));

%-----------------------------------------------------------------------------
% Print the surnames of all teachers, together with the names of the sections
% that they teach, and the code of the course that each section is part of.

program p3 is
for each t in teacher
  for each s in has_lecturer_inv(t)
print(surname(t), section_name(s), code(has_course(s)));

%-----------------------------------------------------------------------------
% As above, but for fourth year courses only.

program p4 is
for each t in teacher
  for each s in has_lecturer_inv(t)
    for each c in has_course(s) such that level(c) = 4
print(surname(t), code(c));

%-----------------------------------------------------------------------------
% Print the names of sections taught by Peter Gray.

program p5 is
for each t in teacher such that forename(t) = "Peter" and surname(t) = "Gray"
  for each s in has_lecturer_inv(t)
print(section_name(s));

program p5a is
for each t in teacher such that forename(t) = "Derek" or surname(t) = "Watson"
  for each s in has_lecturer_inv(t)
print(forename(t), surname(t), section_name(s));

%-----------------------------------------------------------------------------
% Print information about sections of third year courses taught by Peter Gray.

program p6 is
for each c in course such that level(c) = 3
  for each s in has_course_inv(c)
    for each t in has_lecturer(s) such that
      forename(t) = "Peter" and surname(t) = "Gray"
print(section_name(s), code(c));

%-----------------------------------------------------------------------------
% Print the codes of all courses between levels 1 and 4.

program p7 is
for each c in course such that level(c) > 1 and level(c) < 4
print(code(c));

%-----------------------------------------------------------------------------
% Print the result of a simple arithmetic expression.

program p8 is
```

```
print(3 * 3 + 6);

%-----------------------------------------------------------------------------
% Print the names of all undergraduates that take a course above level 3.

program p9 is
for each u in undergrad such that
  some c in takes(u) has level(c) > 3
print(forename(u), surname(u));

%-----------------------------------------------------------------------------
% Print the names of all undergraduates that take at least 2 courses at level
% 3.

program p10 is
for each u in undergrad such that
  at least 2 c in takes(u) have level(c) = 3
print(forename(u), surname(u));

%-----------------------------------------------------------------------------
% Print the names of all undergraduates that take only one course at level 3.

program p11 is
for each u in undergrad such that
  exactly 1 c in takes(u) has level(c) = 3
print(forename(u), surname(u));

%-----------------------------------------------------------------------------
% Definition of a multi-valued function that returns all courses with sections
% taught by the given teacher.

define teaches(t in teacher) ->> course in unidb
  has_course(has_lecturer_inv(t));

%-----------------------------------------------------------------------------
% Print the fornames of all teachers, together with the codes of the courses
% in which they teach sections.

program p13 is
for each t in teacher
  for each c in teaches(t)
print(forename(t), code(c));

%-----------------------------------------------------------------------------
% Definition of the factorial function.

define fac(i in integer) -> integer in unidb
  if i = 1 then 1
  else i * fac(i - 1);

%-----------------------------------------------------------------------------
% Example of use of the factorial function.

program p14 is
print(fac(4) / 4);

%-----------------------------------------------------------------------------
% Print the result of a conditional expression.

program p15 is
print(if 4 = 5 then "true" else "false");

%-----------------------------------------------------------------------------
% Definition of a single-valued function that returns the forename of fourth
% year students and the surnames of all other students.

define one_name(s in student) -> string in unidb
  if year(s) = 4 then forename(s) else surname(s);

%-----------------------------------------------------------------------------
```

```
% Example of the use of the above function.

program p16 is
for each s in student
print(one_name(s), year(s));

%------------------------------------------------------------------------------
% Print all of the values in the given set.

program p17 is
for each i in {3, 4, 5 * 5, 6}
print(i);

%------------------------------------------------------------------------------
% Definition of a multi-valued function that returns two values derived from
% a student’s year.

define gen(s in student) ->> integer in unidb
  {year(s), 2 * year(s)};

%------------------------------------------------------------------------------
% Example of the use of the above function.

program p18 is
for each s in student
  for each i in gen(s)
print(forename(s), i);

%------------------------------------------------------------------------------
% Print the forenames of all students, and integers up to their year of study.

program p19 is
for each u in undergrad
  for each i in {0, 1 to year(u)}
print(forename(u), i);

%------------------------------------------------------------------------------
% Print Graham Kemp’s year of study.

program p20 is
for the s in student such that forename(s) = "Graham" and surname(s) = "Kemp"
print(year(s));

%------------------------------------------------------------------------------
% As above.

program p21 is
print(year(the s in student such that
  forename(s) = "Graham" and surname(s) = "Kemp"));

%------------------------------------------------------------------------------
% As above.

program p22 is
print(year(the p in person such that
  forename(p) = "Graham" and surname(p) = "Kemp" as student));

%------------------------------------------------------------------------------
% Print the year of study of each student (iterating over all people).

program p23 is
for each p in person
print(year(p as student));

%------------------------------------------------------------------------------
% Print the largest value in a set of integers.

program p24 is
print(maximum({1, 5, 7, 9, 8, 6, 4}));
```

```
%------------------------------------------------------------------------------
% Print the total number of members of staff.

program p25 is
print(count(member_of_staff));

%------------------------------------------------------------------------------
% Definition of a single-valued function that returns the number of students
% taking the given course.

define size(c in course) -> integer in unidb
  count(takes_inv(c));

%------------------------------------------------------------------------------
% Print the average number of students taking a course.

program p26 is
print(average(over c in course of size(c)));

%------------------------------------------------------------------------------
% For each level from 1 to 4, print the level and the total number of units in
% courses taught at that level.

program p27 is
for each l in {1 to 4}
print(l, total(over c in course such that level(c) = l of units(c)));

%------------------------------------------------------------------------------
% Print the names of all people who are both students and members of staff.

program p28 is
for each p in person such that
  p is a member_of_staff and p is a student
print(forename(p), surname(p));

%------------------------------------------------------------------------------
% Definition of a single-valued function that returns the duration of the
% longest project supervised by the given teacher.

define longest_duration(t in teacher) -> integer in unidb
  maximum(duration(supervises(t)));

%------------------------------------------------------------------------------
% Definition of a single-valued function that returns the longest project
% supervised by the given teacher.

define longest_project(t in teacher) -> project in unidb
  the p in supervises(t) such that duration(p) = longest_duration(t);

%------------------------------------------------------------------------------
% Definition of an action to print information about a member of staff.

define print_staff(m in member_of_staff) in unidb
  print(forename(m), surname(m), position(m), room(m));

%------------------------------------------------------------------------------
% Definition of an action to print information about a teacher, and the names
% of the sections taught.

define print_teacher(t in teacher) in unidb
  print_staff(t), print("    ", section_name(has_lecturer_inv(t)));

%------------------------------------------------------------------------------
% Print information about all teachers.

program p29 is
print_teacher(teacher);
```