

i. List all the Courses taught by the teacher - "PPC".

```
[] course.coursename(σ course.coursename=teaches.coursename(courseX(σ
teacher.teacherid=teaches.teacherid and teacher.teachername="PPC"(teacher
X teaches)))
```

ii. List all students registered in the courses taught by "PPC".

```
[] student.studentname(σ student.studentid=enrolls.studentid(studentX(σ
enrolls.coursename=course.coursename(enrollsX(σ
course.coursename=teaches.coursename(courseX(σ
teacher.teacherid=teaches.teacherid and teacher.teachername="PPC"(teacher
X teaches))))))
```

iii. List the timings of all courses in Class-Room "NC142".

```
[] timings.starttime,timings.endtime,timings.date(σ
timings.timeid=timetable.timeid(timingsX(σ
timetable.coursename=course.coursename(timetableX(σ
course.coursename=willbein.coursename(courseX(σ
willbein.roomnumberclassroom.roomnumberand
classroom.roomnumber="NC142"(classroom X willbein))))))
```

iv. List the name of the students who received the highest marks in the courses taught by "PPC"

```
[] student.studentname (σ max(marks.marks) and
marks.marksid=student.studentid (studentX(σ
have.marksid=marks.marksid(marksX(σ
have.coursename=course.coursename(have X(σ
course.coursename=teaches.coursename(courseX(σ
teacher.teacherid=teaches.teacherid and teacher.teachername="PPC"(teacher
X teaches)))))) group by student.studentid)
```

v. List the students who have received a grade of "EX" in the largest number of courses.

```
(σ max(count(gradecard.grade)) and student.studentid=grade.gradeid
(studentX(σ gradecard.gradeid=performance.gradeid and
count(gradecard.grade="EX") and gradecard.grade="EX" (gradecard X
performance)) group by gradecard.gradeid)group by student.studentid)
```