An Exploration of Teachers' Perspective about the Learning of Iteration-Control Constructs

Emanuele Scapin & Claudio Mirolo

University of Udine, Italy

ISSEP 2019
Larnaca, November 18–20, 2019

Outline

Outline

Students' difficulties

- Students appear to struggle with programming
- Several works report about students' difficulties with basic control flow constructs, and specifically with iteration
- Still little explored for pre-tertiary education
- *Project target: to identify new methodological instruments to improve student learning of iterations*

Students' difficulties

- Students appear to struggle with programming
- Several works report about students' difficulties with basic control flow constructs, and specifically with iteration
- Still little explored for pre-tertiary education
- *Project target: to identify new methodological instruments to improve student learning of iterations*

Students' difficulties

- Students appear to struggle with programming
- Several works report about students' difficulties with basic control flow constructs, and specifically with iteration
- Still little explored for pre-tertiary education
- *Project target: to identify new methodological instruments to improve student learning of iterations*

Students' difficulties

- Students appear to struggle with programming
- Several works report about students' difficulties with basic control flow constructs, and specifically with iteration
- Still little explored for pre-tertiary education
- *Project target: to identify new methodological instruments to improve student learning of iterations*

Project's main steps

1. *Interviewing a pilot sample of instructors about their approach to the teaching of iteration and their perception of students difficulties*
2. *Collecting information about students' perception on the topic through a short survey*
3. Based on the outcome of steps 1 and 2, designing a survey to collect related information and good practices from a larger sample of teachers and students
4. Devising a methodological approach to the teaching of iteration and building a catalogue of significant program examples to support students' learning
5. Experimenting the instructional strategies in classroom to investigate on their effectiveness

Project's main steps – present subject

1. *Interviewing a pilot sample of instructors about their approach to the teaching of iteration and their perception of students difficulties*
2. *Collecting information about students' perception on the topic through a short survey*
3. Based on the outcome of steps 1 and 2, designing a survey to collect related information and good practices from a larger sample of teachers and students
4. Devising a methodological approach to the teaching of iteration and building a catalogue of significant program examples to support students' learning
5. Experimenting the instructional strategies in classroom to investigate on their effectiveness

Outline

The path to our present study: Teachers

- Structure of teacher interviews (and student survey)
  - Pedagogical Content Knowledge (PCK)
  - Content Representation (CoRe)
  - Concept Inventories
- Face-to-face interviews (step 1) of 20 experienced upper secondary teachers of introductory programming in different kinds of high schools in North-East of Italy

The path to our present study: Teachers

- Structure of teacher interviews (and student survey)
  - Pedagogical Content Knowledge (PCK)
  - Content Representation (CoRe)
  - Concept Inventories
- Face-to-face interviews (step 1) of 20 experienced upper secondary teachers of introductory programming in different kinds of high schools in North-East of Italy

The path to our present study: Students

- Survey (step 2) administered to a sample of 164 students from the schools of the interviewed teachers
- About 80% third-year (K11) students of technical institutes
- Both questions on their subjective perception and tiny problems on iteration constructs

The path to our present study: Students

- Survey (step 2) administered to a sample of 164 students from the schools of the interviewed teachers
- About 80% third-year (K11) students of technical institutes
- Both questions on their subjective perception and tiny problems on iteration constructs

The path to our present study: Students

- Survey (step 2) administered to a sample of 164 students from the schools of the interviewed teachers
- About 80% third-year (K11) students of technical institutes
- Both questions on their subjective perception and tiny problems on iteration constructs

Teacher interview protocol

1. Course organization (5 questions)	
programming languages, key programming concepts, related lesson plan, how much time for each concept, extra-computing prerequisites	
2. Introductory programming in general (6 questions)	
2.1. teaching	Are the tasks assigned to students simple variations of those dealt with in class? Or do they cover unfamiliar situations as well? What are your more frequent suggestions to students for improving their programming performance?
2.2. learning	What is the major learning obstacle that students face before being introduced to object-oriented programming?
2.3. assessment	How do you assess a working solution if it is inefficient, or convoluted, or somehow at odds with what you expected? While trying to achieve the assigned tasks, do you expect your students to apply the models introduced in class? Or do you also appreciate "creative" solutions? Are the different solutions by students compared in class? How?

... ...

Teacher interview protocol (continued)

3. Focus on iteration (5 questions)	
3.1. teaching	Can you show some of your favorite examples to make students learn how to apply the iteration constructs?
	In your teaching, do you cover the mappings between different iteration constructs (for, while, do-while/repeat-until)?
3.2. learning	In your experience, to what extent can students master the termination condition of a loop?
	Which features of the iteration constructs are usually understood by (most) students, and which are more difficult to them?
3.3. assessment	How do you usually assess an incorrect termination condition?
	And oversights about the first or last iteration?
4. General educational issues (3 questions)	
strategies to motivate students, manage different learning styles, deal with students' criticisms	
5. Other thoughts (1 question)	
Any other issues you deem important to consider about the teaching/learning of programming?	

Student survey

1. Course organization (3 questions)
favorite programming languages, poor understanding of mathematical/logic prerequisites, accordance of the subject with personal expectations

2. Introductory programming in general (3 questions)
Do you think it would be needed to spend more time on some programming concepts? Which ones? (range of options or open "other" field)
Which kind of errors has been most penalizing for your grading? (open question)
Are you usually successful in solving unfamiliar programming problems? (Likert scale of 4 levels)

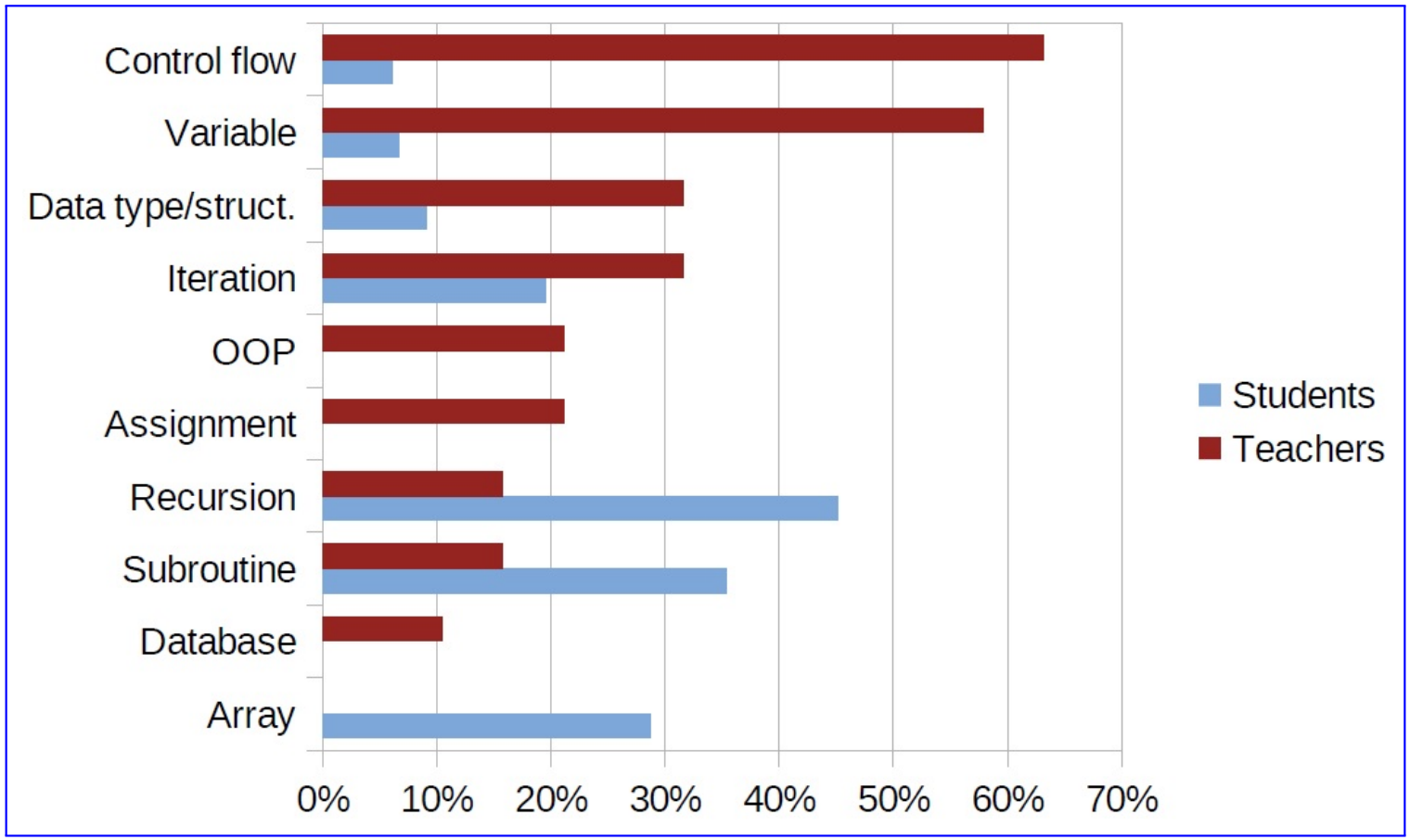
3. Focus on iteration (1 question and 3 tiny problems)
What do you find most difficult when trying to use a loop? (range of options)
Problem 3.1: Given the statement of a simple problem being solved, choose the correct *loop condition* in a flow chart. (4 options available)
Problem 3.2: Given a while loop with a composed condition and a nested conditional, determine the number of iterations for a given input. (6 options)
Problem 3.3: Given 5 code fragments involving nested construct with simple conditions, identify the functionally equivalent ones.

4. Other thoughts (1 question)
Do you have any suggestion to make learning informatics more interesting?

Outline

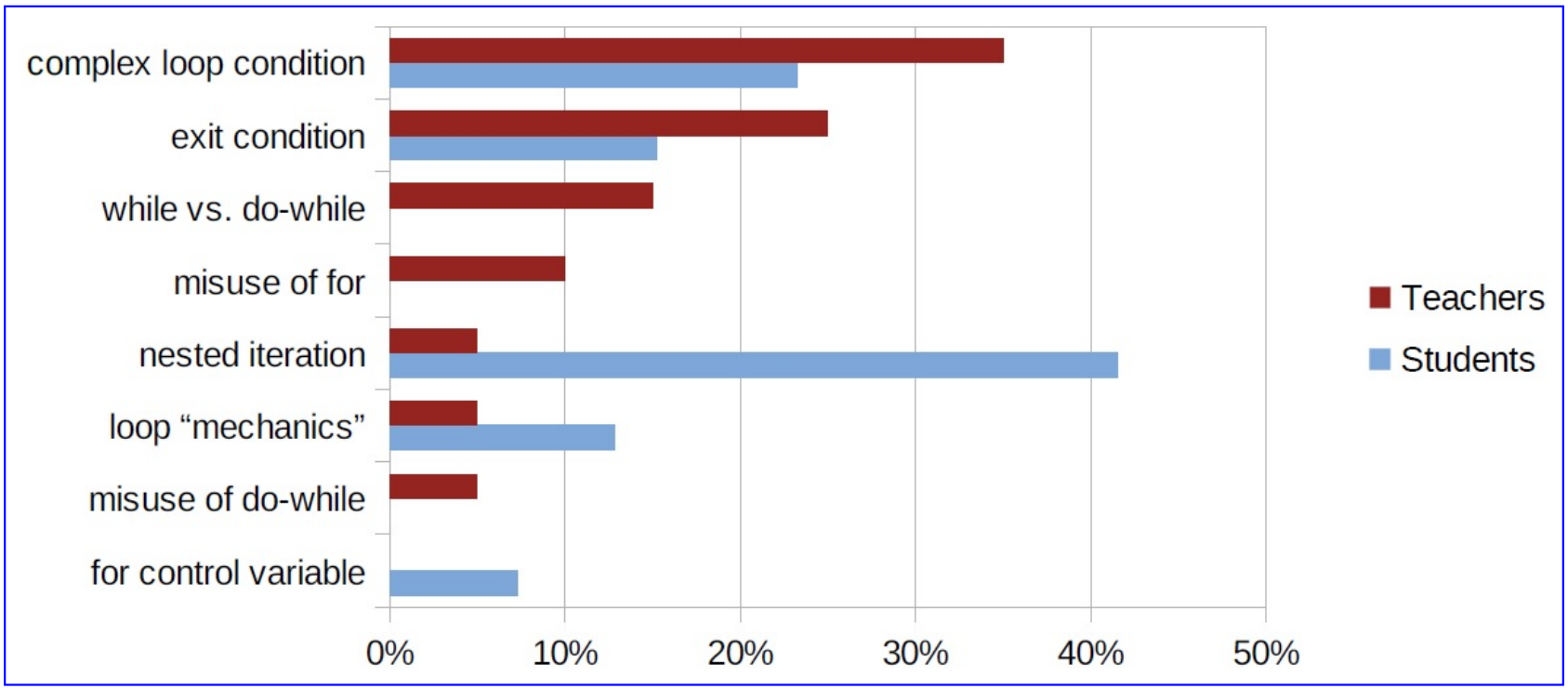
Perception: General issues

Multiple options could be selected. . .

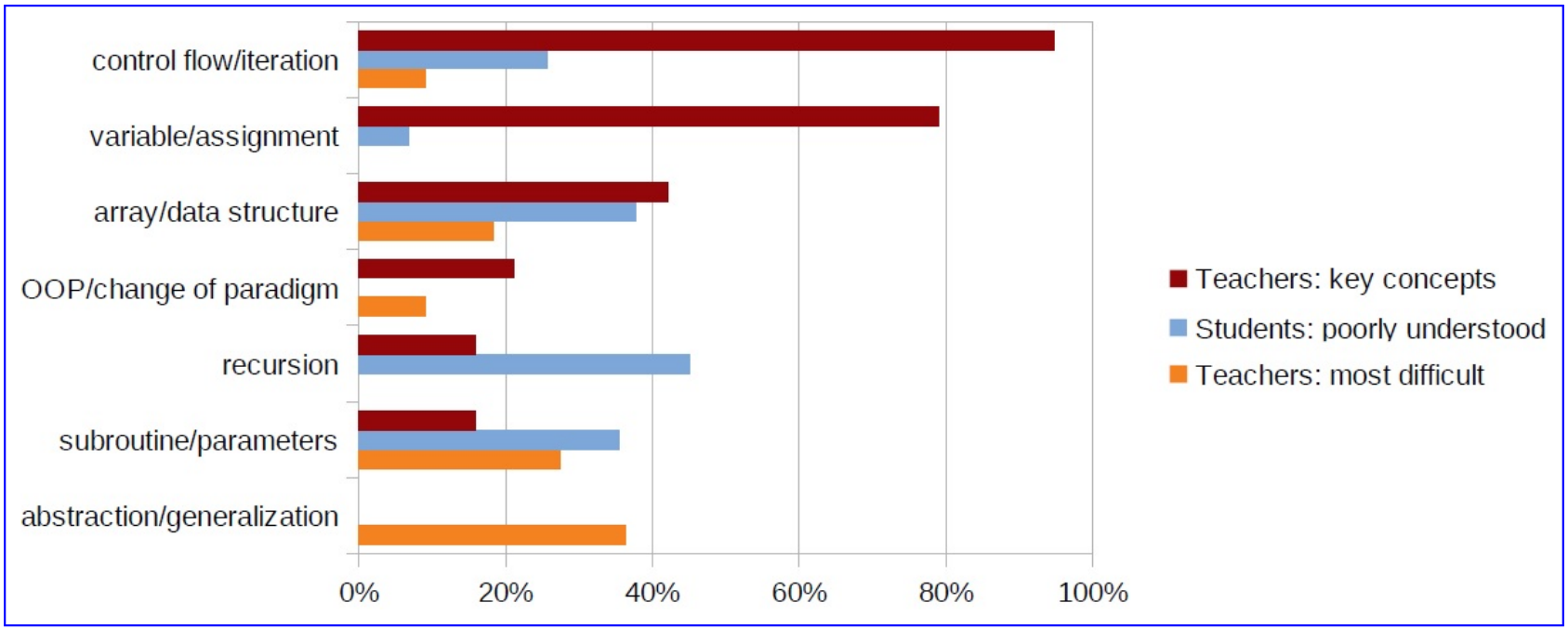


Perception: Iteration issues

Multiple options could be selected. . .



Perception: Issues vs. concept relevance

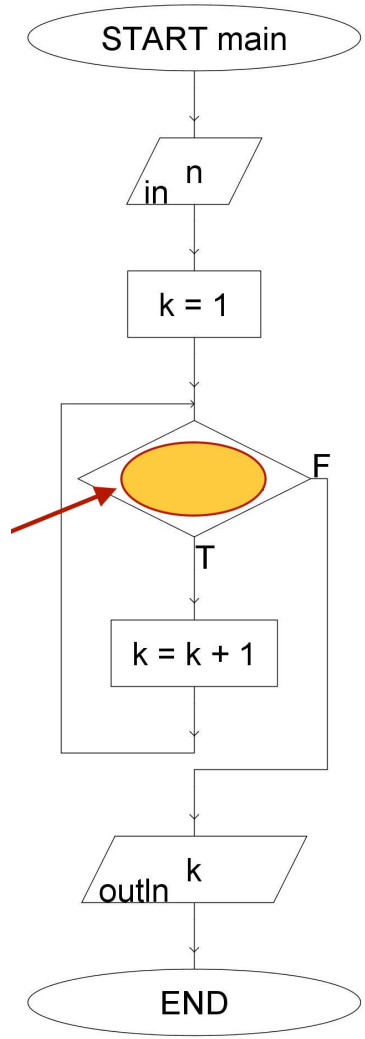


Student survey: Tiny problem 1

- To investigate the ability to link loop components and problem statement

Problem 1

Identify the correct condition...



- Options:

$$2^k = n \quad 2^k \leq n \quad 2^k < n \quad 2^k > n$$

- Less than **40%** of the students provided the correct answer ($\leq$)

Student survey: Tiny problem 1

- To investigate the ability to link loop components and problem statement

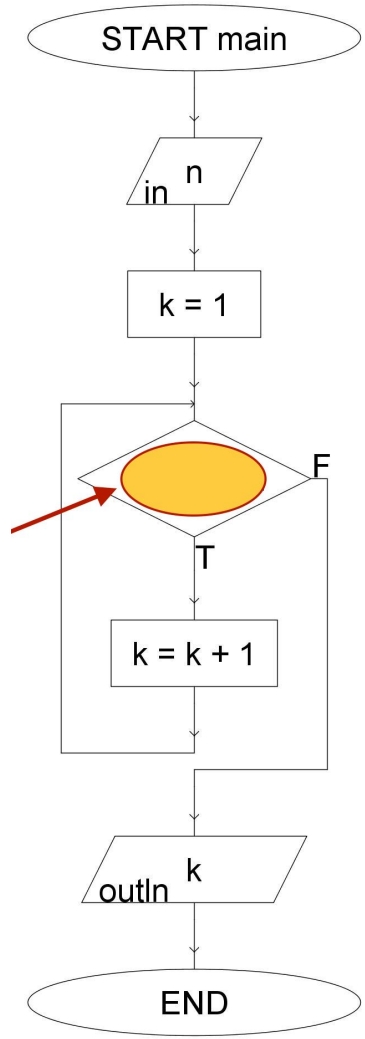
Problem 1

Identify the correct condition...

- Options:

$$2^k = n \quad 2^k \leq n \quad 2^k < n \quad 2^k > n$$

- Less than **40%** of the students provided the correct answer ($\leq$)



Student survey: Tiny problem 2

- To investigate mastery of the “mechanics” of iteration

Problem 2

How many iterations for $m = 15$ and $n = 44$?

```
int x = m, y = n;

while ( x > 1 && y > 1 && x != y ) {

  if ( x < y )
    y = y - x;
  else
    x = x - y;
}
if ( x == 1 || y == 1 )
  printf( "m=%d e n=%d sono primi fra loro", m, n );
else
  printf( "m=%d e n=%d non sono primi fra loro", m, n );
```

Student survey: Tiny problem 2

- To investigate mastery of the “mechanics” of iteration

Problem 2

How many iterations for $m = 15$ and $n = 44$?

- Options:

  1 2 3 4 or more The loop will not terminate

- About **60%** of the students identified the right answer (3)

Student survey: Tiny problem 3

- To investigate the understanding of combinations of (nested) control flow constructs

Problem 3

Which of the following programs are functionally equivalent?

Tiny problem 3

```
int x = m, y = n;

while ( x != y ) {

  while ( x < y )
    x = x + m;
  while ( x > y )
    y = y + n;
}

printf("risultato: %d", x);
```

```
int x = m, y = n;

while ( x != y ) {

  if ( x > y )
    y = y + m;
  else
    x = x + n;
}

printf("risultato: %d", x);
```

```
int x = m, y = n;

while ( x != y ) {

  while ( x < y || x > y ) {
    x = x + m;
    y = y + n;
  }
}

printf("risultato: %d", x);
```

```
int x = m, y = n;

while ( x != y ) {

  if ( x < y )
    x = x + m;
  else
    y = y + n;
}

printf("risultato: %d", x);
```

```
int x = m, y = n;

while ( x != y ) {

  if ( x < y )
    x = x + x;
  else
    y = y + y;
}

printf("risultato: %d", x);
```

Tiny problem 3

- To investigate the understanding of combinations of (nested) control flow constructs

Problem 3

Which of the following programs are functionally equivalent?

- Only less than **20%** of the students were successfull

Most popular algorithms to address iterations

- sum/average of a number sequence
- power function
- counting odd/even numbers in a sequence
- factorial function
- min/max values of a sequence
- Euclid's GCD algorithm
- input data control (`do-while`)
- math number sequence
- first *n* multiples of a number
- number base conversion
- iteration over an array
- pictures drawing with chars
- *n*-th element of a sequence
- drawing a polygons

What emerged from the investigation

- control flow and iteration are deemed to be key concepts for introductory programming
- extra-computing prerequisites
  - mathematical/logical background
  - text comprehension
- students' vs teachers' perceptions
  - major difficulties in the teachers' opinion: complex loop conditions
  - major difficulties in the students' opinion: also nested iterations
- students' difficulties confirmed by their performance in the three small tasks included in the survey

What emerged from the investigation

- control flow and iteration are deemed to be key concepts for introductory programming
- extra-computing prerequisites
  - mathematical/logical background
  - text comprehension
- students' vs teachers' perceptions
  - major difficulties in the teachers' opinion: complex loop conditions
  - major difficulties in the students' opinion: also nested iterations
- students' difficulties confirmed by their performance in the three small tasks included in the survey

What emerged from the investigation

- control flow and iteration are deemed to be key concepts for introductory programming
- extra-computing prerequisites
  - mathematical/logical background
  - text comprehension
- students' vs teachers' perceptions
  - major difficulties in the teachers' opinion: complex loop conditions
  - major difficulties in the students' opinion: also nested iterations
- students' difficulties confirmed by their performance in the three small tasks included in the survey

What emerged from the investigation

- control flow and iteration are deemed to be key concepts for introductory programming
- extra-computing prerequisites
  - mathematical/logical background
  - text comprehension
- students' vs teachers' perceptions
  - major difficulties in the teachers' opinion: complex loop conditions
  - major difficulties in the students' opinion: also nested iterations
- students' difficulties confirmed by their performance in the three small tasks included in the survey

Other thoughts

- Lack of alignment between math and informatics syllabi
- Robotics environments to motivate students
- Object-first approach?

Conclusions

- Iteration is a central concepts for novice programmers
- Most examples of programs may induce a stereotyped use of iteration. . .
- . . . Catalogue of more varied and interesting examples?
- . . . to develop students' abstraction and generalization skills
- Curricular issues (Math & Logic vs. Informatics)

Thanks for your attention. . .

Any questions?

Thanks for your attention. . .

Any questions?