Serdica J. Computing **14** (2020), No 1–2, 13–26

Serdica
Journal of Computing

Bulgarian Academy of Sciences
Institute of Mathematics and Informatics

TEACHING PROGRAMMING THROUGH THE "MODULAR ARITHMETIC" MATH PROJECT

Mariyana Raykova, Stoyan Boev

Abstract. Project-based approach was integrated into the Computer science programme at New Bulgarian University to enhance the learning process for first year students. A multidisciplinary project based on main concepts of Programming and Discrete mathematics was designed and developed. The concepts and relations between them were presented by concepts maps and the tasks were ordered according to revised Bloom's taxonomy. So it became possible to evaluate the cognitive learning levels of the students from simple remembering of information to analyzing and creating, to improve the self-learning skills, the ability to search for new information, working in teams and communication skills. The students had the opportunity to apply algorithms related to different data structures, memory and time optimisation that are used in the business and to integrate different programming technologies in a project.

1. Introduction. Mathematics is a fundamental intellectual tool in computing, but computing is increasingly used as a key component in mathematical problem-solving. When we teach these two subjects—Programming and

ACM Computing Classification System (1998): K.3.2, G.2.
Key words: teaching programming, discrete mathematics, modular arithmetic, concept map.

Discrete Mathematics, separately in Computer science program at New Bulgarian University it is very hard for the first year students not only to understand them but to see the connection between the subjects and apply their knowledge in practice. Our point of view is that teaching by the conventional way does not work mainly due to low interest and lack of study skills in students. But combining conventional teaching, computer aided learning, project-based approach and collaboration work, we were able to make the process more clear, structured and attractive for students [4, 5].

So we decided to integrate Discrete mathematics in Programming course by adding a project-based educational element to develop deeper knowledge, skills and competences in both areas. That gave an opportunity for the students to work in multidisciplinary environment and to apply knowledge from one subject area into another.

2. Purposes and Challenges of Project-based Learning. During our experience based on 9 years in teaching Discrete mathematics and Programming to first year students we face some very common issues that concern the secondary education:

- *lack of self learning skills;*
- *lack of search and research competencies;*
- *lack of communication skills and team working capabilities;*
- *lack of interest to find the most effective approach for solving a problem;*
- *sticking to the cognitive learning level of remembering of information.*

In order to overcome these problems and to make programming and mathematics more attractive, understandable, practical and useful for the students, we introduced an interdisciplinary project-based learning in the curriculum of the programming course. Two math projects were proposed to the first year students as part of their programming course—"Modular arithmetic" and "Permutations". Here we will focus on the first one that aims to apply in practice the concepts from programming and modular arithmetic in a single project. Some of the expected results are listed below:

- to develop knowledge in concepts like *integer division with remainder, greatest common divisor of integers, co-prime and prime numbers, ring of integers mod n* and *finite field*;
- to write *reliable, readable* and *reusable* source code supported by comments and technical documentation;

- to find the most appropriate *data structure* and the most effective solution to a problem according to *memory* and *time of execution* (for example discrete logarithm operation vs. exponentiation);
- to improve not only the *algorithmic* but *abstract thinking* (for example set vs. structure, array vs. dynamic array or binary tree);
- to learn how to use *version control* for software development and *github*;
- to develop *quality assurance* skills applying different cases for accurateness measurement of the system;
- to develop *team work capabilities* and *communication skills* via collaboration and presentation;
- to integrate *new* and *environmentally friendly external technologies* in order to re-boost the usability of the project—HTML, CSS, SVG, XML, etc.
- to create *architecture of software library* and develop an *user friendly interface* for *modular arithmetic calculator* making research on existing online solutions.

One of the biggest challenge faced in the project is to develop cognitive learning in students at all the six levels of the revised Bloom's taxonomy (remembering, understanding, applying, analyzing, evaluating, creating). If we start from the first task to the last one we should clearly see that we start from the very first level—remembering of information till the last one—creating.

3. Project Design by Concept Maps. Concept maps (CMs) represent the relationships between concepts, usually displayed as a diagram. They are commonly used as teaching and assessment tools in the processes of teaching and learning in order to enhance learners' knowledge and competency. There are several important advantages of CMs included in the education cycle: graphical representation of the basic concepts in the topic or in the course content; emphasizing the most important parts of the learning material; give teachers detailed information about the learners' needs and can suggest the use of more appropriate learning materials, as well as the most suitable order in which these can be presented to the learners; can also stimulate learners' creative thinking. Because of the initial structure of CMs, they can be effectively used to hierarchically organize the course's topics, wherein the most important facts are presented at the top of the map, while the less important ones—at the lower levels [12]. This teaching strategy helps students focus on the concept learning. Concepts are the corner stones of the basic knowledge construction and deeper comprehension [12, 14, 15].

Concept maps create an opportunity to measure higher levels of knowledge and competency by using the revised Bloom's taxonomy [11, 13].

Our aim was to create a project that had to be completed by the students in the frame of the programming course in first semester of their study and cover all the concepts described in the map for both subject areas—programming and modular arithmetic (Figure 1).

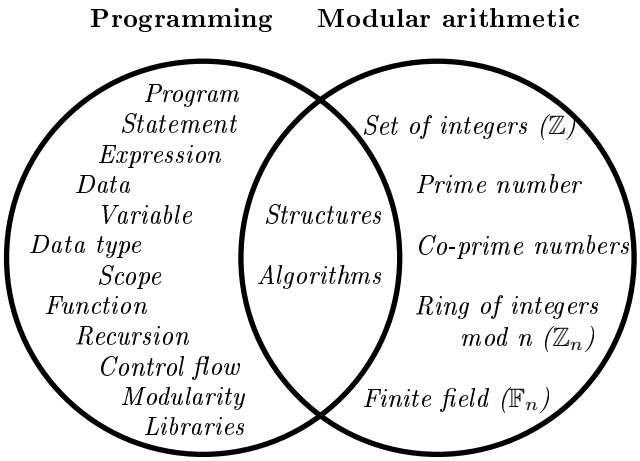

Fig. 1. Basic concepts in the project

The project tasks were developed on the CM given in Figure 2–3 (split for better visualization) [1, 2, 3]. The left-hand side of the CM shows the programming main concepts that are involved in the course "Programming", while the right-hand side shows the modular arithmetic concepts included in the project. Both subject areas meat each other in the concepts *Structures (Data/Discrete)* and *Algorithms* [7].

Using this map the project designer could easily prepare and arrange the assignments in the project and after that the teacher could make an assessment and provide feedback on the course and project completion. In the project we have implemented the revised Bloom's levels of knowledge [11] in increased order, so the first assignment measures just remembering of the information, while the last one measures the evaluation and creation levels. The project follows the so-called "System of tasks methodology" [6] and the order and the difficulty of the tasks in the project are prepared and selected following a strictly designed approach.

4. Project Content, Development and Student Results. The project contains 15 assignments and it was given to the students in the third week of the semester. They had 9 weeks to finish it. The idea behind the assignments of the project is for the students to be able to learn and practice the main concepts from the two subject areas—Programming and Modular arithmetic (Figure 1). And not only that, but also to measure on which cognitive learning level the students are able to learn the concepts and which of the concepts are harder for them. Something more—the project gives them an opportunity to practice their knowledge not only simulating some small problems during classes. Each of the fifteenth assignments measures one or more concepts from the CM given in Figure 1 and combines them for the two subject areas.

The tasks from the project are divided into five set of assignments—Ring operations in $\mathbb{Z}_n$, Division in $\mathbb{Z}_n$, Exponentiation in $\mathbb{Z}_n$, Discrete logarithm in $\mathbb{Z}_n$, Finite field $\mathbb{F}_n$. Each group is classified according to the set of the concepts that are required in both subject areas, the cognitive learning level they measure, the difficulty of the task and the grade the students get if the task is completed.

All the assignments from number 1 till 4 are by the set of assignments Ring operations in $\mathbb{Z}_n$ and are requiring and measuring the same cognitive learning levels (*remembering*, *understanding*, *applying*) and concepts from both subject areas. They measure programming concepts *data types*, *primitive and compound arrays*, *function definition*, *arithmetic operators*, *control flow operators*, together with mathematical concepts—*ring of integers mod n*, *elements from the ring*, *operations in the ring*, *division with reminder*.

Ring operations in $\mathbb{Z}_n$		
1. Define a C++ function that fills in an array used to store the elements of $\mathbb{Z}_n$.		
2. Define a C++ function that performs the operation addition of elements in $\mathbb{Z}_n$.		
3. Define a C++ function that performs the operation subtraction of elements in $\mathbb{Z}_n$.		
4. Define a C++ function that performs the operation multiplication of elements in $\mathbb{Z}_n$.		
Programming concepts	**Mathematical concepts**	**Cognitive learning levels**
Data types —primitive (int, bool) *Function (definition)* *Control flow operators* *Arithmetic operators*	*Ring of integers mod n (elements)* *Ring of integers mod n (operations)* —addition —negative —subtraction —multiplication *Integer division with remainder*	*Remembering* *Understanding* *Applying*

The project starts with a trivial case (*Task 1*) where the students have to

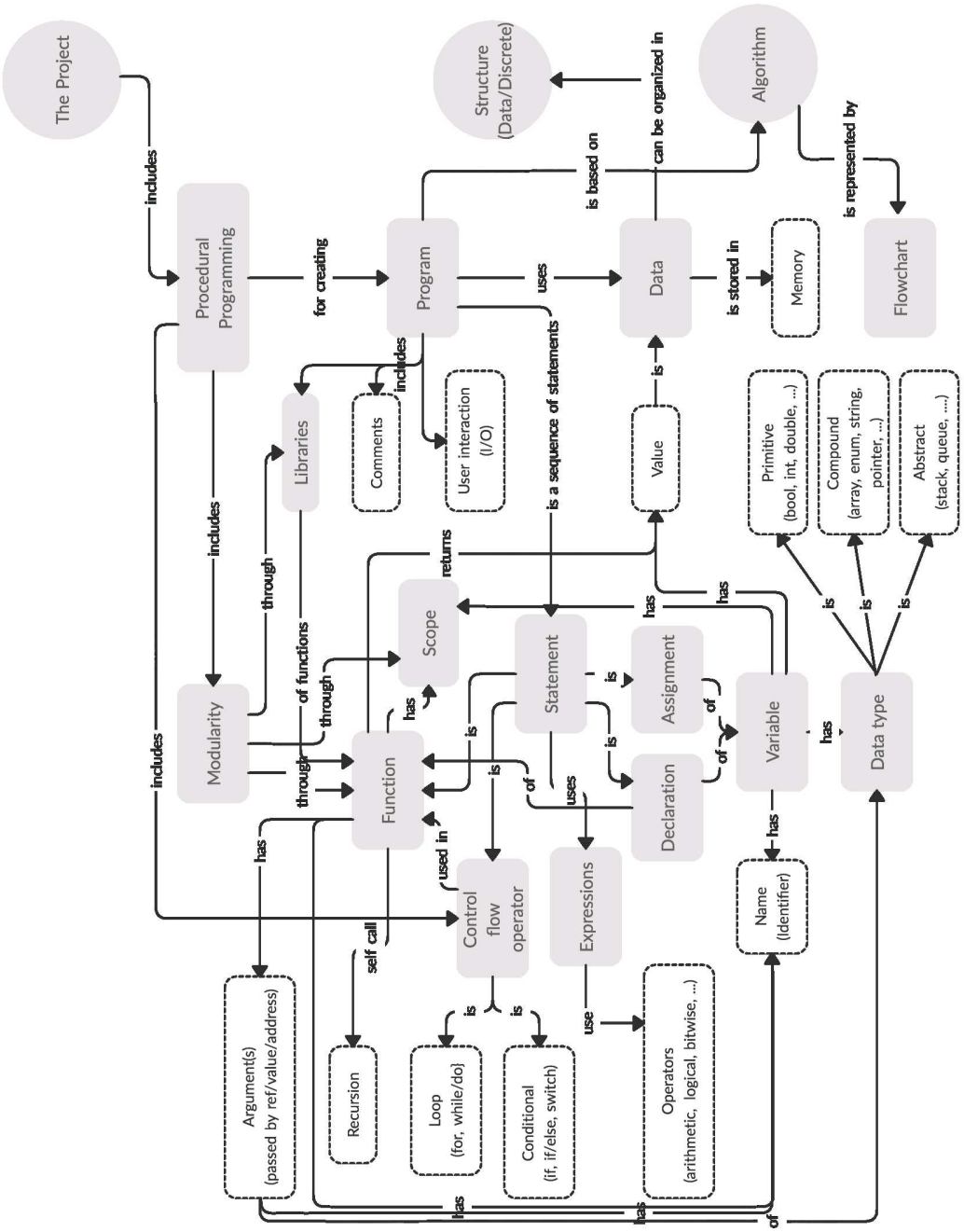

Fig. 2. Project concept map (split, part 1)

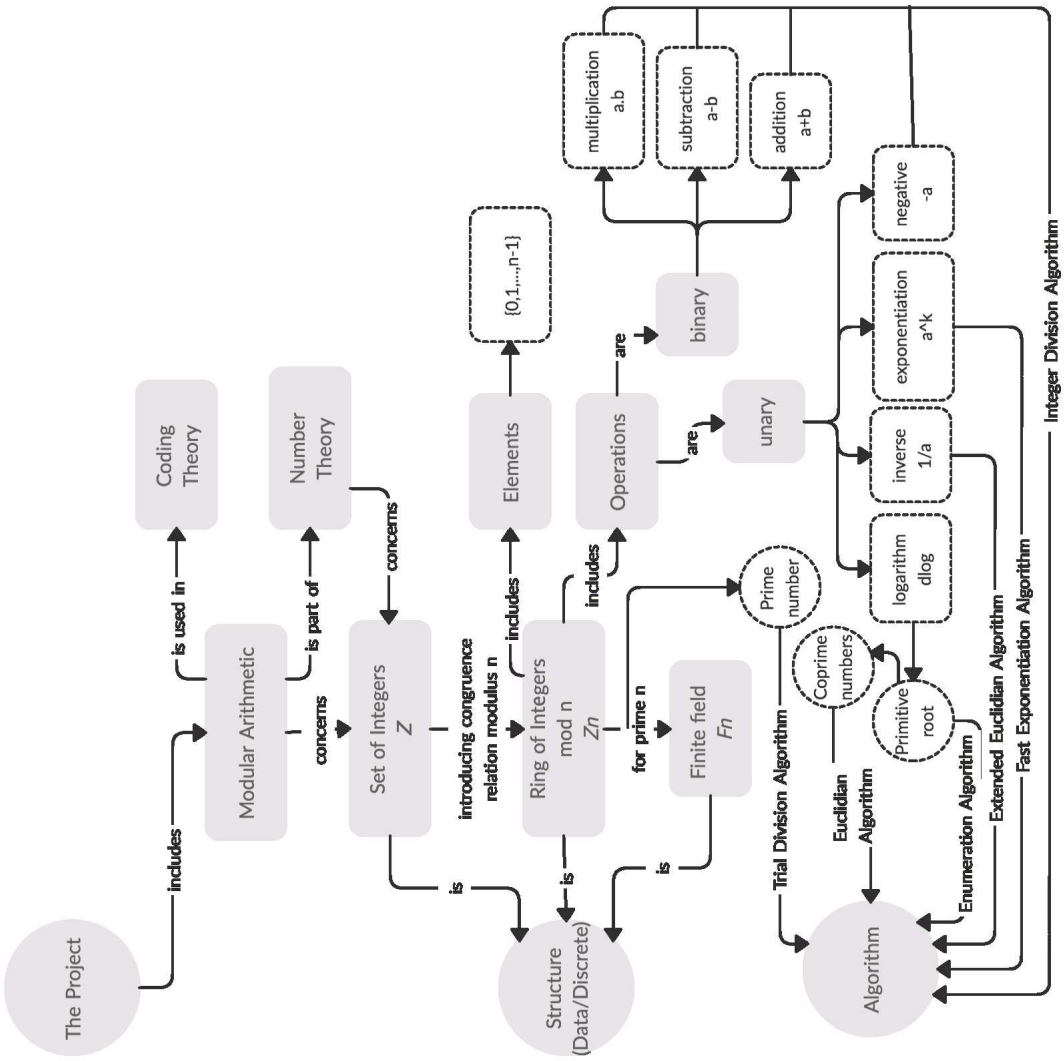

Fig. 3. Project concept map (split, part 2)

fill in an array with values that follow a given rule. In our case we have the $\mathbb{Z}_n$ quotient ring and the number n of data items that should be known in order to complete the task is n. Students have to construct a dynamic array and fill it with all possible reminders, which can be obtained when dividing by n or in other words with values from 0 till $n-1$. This task should measure the level of *remembering of information*, as the functionality for filling in an array with values is a task that is trivial. The concepts that are trained here are *data types and variables*, *arrays*, *dynamic arrays*, *function declaration*, *function definition*, *function call*, *parameters passed by value*, *parameters passed by address*, *loops* [8, 9, 10]. The task measures if the concept of *quotient ring* is learned.

The assignments from number 5 to 8 are from the set of assignments Division in $\mathbb{Z}_n$ and they are designed to measure first five cognitive learning levels—remembering, understanding, applying, analyzing, evaluating. They measure and apply the programming concept *comparison operators* and mathematical concepts *inverse and division operations in quotient ring*, as well as *greatest common divisor of integers* and *co-prime numbers*.

Division in $\mathbb{Z}_n$		
5. Define a C++ function that finds the pairs of elements in $\mathbb{Z}_n$ with product 1. Save the found elements in a two-dimensional array. Save the first element in the first row and the second element—in the same column of the second raw.		
6. Define a C++ function that finds the inverse of an element in $\mathbb{Z}_n$, if it exists. Use the array from Task 5.		
7. Define a C++ function, that finds the inverse element of an element in $\mathbb{Z}_n$, if it exists. Use Bézout's Identity. Compare the complexity with the one of Task 6.		
8. Define a C++ function that performs division of elements in $\mathbb{Z}_n$ when the operation is defined, and -1 otherwise.		
Programming concepts	**Mathematical concepts**	**Cognitive learning levels**
Data types	*Ring of integers mod n (elements)*	*Remembering*
—primitive (int, bool)	*Ring of integers mod n (operations)*	*Understanding*
Function (definition)	—inverse	*Applying*
Control flow operators	—division	*Analyzing* (Task 7)
Arithmetic operators	*Greatest common divisor*	*Evaluating* (Task 7)
Comparison operators	*Co-prime numbers*	

The set of assignments Exponentiation in $\mathbb{Z}_n$, includes only two assignments with numbers 9 and 10. They measure the same cognitive learning levels, but the assignments are with greater difficulty and require understanding of more complex concepts—*bitwise operators* from programming, *exponentiation operation in quotient ring* and *binary expansion of integers* from mathematics.

Exponentiation in $\mathbb{Z}_n$		
9. Define a C++ function that performs the exponentiation operation for an element in $\mathbb{Z}_n$, based on a memory efficient algorithm.		
10. Define a C++ function that performs the exponentiation operation for an element in $\mathbb{Z}_n$, based on the right to left binary method. Compare the complexity with the one of Task 9.		
Programming concepts	**Mathematical concepts**	**Cognitive learning levels**
Data type —primitive (int, bool) *Function (definition)* *Control flow operators* *Arithmetic operators* *Comparison operators* *Bitwise operators*	Ring of integers mod n (elements) *Ring of integers mod n (operations)* —exponentiation *Binary expansion of an integer*	*Remembering* *Understanding* *Applying* *Analyzing* (Task 10) *Evaluating* (Task 10)

Discrete logarithm in $\mathbb{Z}_n$ group consists of three assignments and measures the same five cognitive learning levels. The programming concepts are the same as in the previous section, but the mathematical concepts are new—*discrete logarithm* and *primitive root in $\mathbb{Z}_n$*.

Discrete logarithm in $\mathbb{Z}_n$		
11. Define a C++ function that checks if a given number is a primitive root in $\mathbb{Z}_n$.		
12. Define a C++ function that finds all primitive roots in $\mathbb{Z}_n$.		
13. Define a C++ function that calculates a discrete logarithm in $\mathbb{Z}_n$ if the operation is valid and returns -1 otherwise. Compare the complexity with the one of Task 10.		
Programming concepts	**Mathematical concepts**	**Cognitive learning levels**
Data type —primitive (int, bool) —compounded (array) *Function (definition)* *Control flow operators* *Arithmetic operators* *Comparison operators*	Ring of integers mod n (elements) *Ring of integers mod n (operations)* —discrete logarithm *Primitive root in $\mathbb{Z}_n$*	*Remembering* *Understanding* *Applying* *Analyzing* (Task 13) *Evaluating* (Task 13)

The last section measures all cognitive learning levels. The programming concepts here are more complex—*user interaction*, *modularity*, *libraries*, *flowcharts*, *memory optimization*, *time optimization*, and the new mathematical concept is *finite field with prime number of elements*.

Finite field $\mathbb{F}_n$		
14. Define a C++ function, that checks if $\mathbb{Z}_n$ is a finite field.		
15. Create a console application for modular calculator. Compare the result with some of the existing open source solutions.		
Programming concepts	**Mathematical concepts**	**Cognitive learning levels**
User interaction	*Finite field (elements)*	*Remembering*
Modularity	*Finite field (operations)*	*Understanding*
Libraries	*Prime number*	*Applying*
Comments		*Analyzing* (Task 15)
Flowchart		*Evaluating* (Task 15)
Optimization		*Creating* (Task 15)
—memory		
—time		

The project definition consists not only of list of tasks, but in the parts where a more difficult and nontrivial algorithm is required, we supply some theoretical notes. This notes are used in order to help or lead the students to the right direction and to the point where they should be able to search or research in order to complete the task. When measuring higher cognitive learning levels, the project assignments require from students to search for more effective solution, that is readable and requires less resources and is enough clear to read. For example some of the most efficient algorithms in cryptography require high powering of elements in the ring $\mathbb{Z}_n$. In other words if $a \in \mathbb{Z}_n$ and m is a sufficiently large positive integer then our target is to compute a^m for minimum time. Task 9 and 10 from the project concern that problem and two approaches are given:

1) Find the smallest positive integer k, such that $a^k \equiv_n 1$ and then

$$a^m \equiv_n a^{m \bmod k}.$$

Example: Let $n = 7$, $a = 3$, $m = 100$. In $\mathbb{Z}_7$ we have

$$\begin{aligned} 3^2 &= 3.3^1 \equiv_7 2 \\ 3^3 &= 3.3^2 \equiv_7 3.2 \equiv_7 6 \\ 3^4 &= 3.3^3 \equiv_7 3.6 \equiv_7 4 \\ 3^5 &= 3.3^4 \equiv_7 3.4 \equiv_7 5 \\ 3^6 &= 3.3^5 \equiv_7 3.5 \equiv_7 1 \end{aligned}$$

and it follows that $k = 6$ and $3^{100} \equiv_7 3^{100 \bmod 6} \equiv_7 3^4 \equiv_7 4$.

2) Find the binary representation of m and compute $a, a^2, a^{2^2}, a^{2^3}, \ldots, a^{2^{k-1}}$, where k is the number of digits in binary representation of m.

Example: Let $n = 7$, $a = 3$, $m = 100$. *Then* $100 = 64 + 32 + 4 = (1100100)_2$, $k = 7$ *and compute:*

$$\begin{array}{lll} 3^2 & = (3^1)^2 & \equiv_7 2 \\ 3^4 & = (3^2)^2 & \equiv_7 2^2 \equiv_7 4 \\ 3^8 & = (3^4)^2 & \equiv_7 4^2 \equiv_7 2 \\ 3^{16} & = (3^8)^2 & \equiv_7 2^2 \equiv_7 4 \\ 3^{32} & = (3^{16})^2 & \equiv_7 4^2 \equiv_7 2 \\ 3^{64} & = (3^{32})^2 & \equiv_7 2^2 \equiv_7 4 \end{array}.$$

Hence

$$3^{100} = 3^{64+32+4} \equiv_7 4.2.4 \equiv_7 4.$$

These algorithms are demonstrated in the listing below which presents an illustrative realization with C++ written by students.

```
bool IsNumInRing(int num, int SizeOfRing) {
        if (num < SizeOfRing && num >= 0)
                return true;
        return false;
}
int Powering(int base, int power) {
        int result = 1;
        while (power > 0)
        {
                result *= base;
                power--;
        }
        return result;
}
int FirstMethodFastPowering(int SizeOfRing, int base, int power) {
        if (!IsNumInRing(base, SizeOfRing))
                return -1;
        else
        {
                int k = 2;
                int result = Powering(base, k)% SizeOfRing;
                while (result != 1) {
                        k++;
                        result = (result*base) % SizeOfRing;
                }
                return power % k;
        }
long long int SecondMethodFastPowering (int num, int pwr2){
     int x;
     long long int ret = num;
     int pwr = 1 << pwr2;
     for (x = 0; x < pwr; x++)
       ret *= num;
     return ret;
}
```

We evaluate the project results with grades from 3 (fair) till 6 (excellent). To be evaluated as *fair*, the students need to complete the tasks up to 8 inclusive.

For the higher grade *good* they need to complete the tasks 9 and 10, for a *very good* grade they have to propose solutions of tasks 11, 12, 13, and for *excellent* grade, of tasks 14 and 15. Since more than 60% of the students had made an attempt to complete the project, this immediately helped them to get better results on the final exam in the Programming course. We had around 10% fewer students failed on the final exam compared with students from previous years, when we had not given them such projects. There were a number of students who not only solved the entire project, but built additional functionality in their projects and solved tasks, they had created themselves. With these students we reached the highest cognitive learning level—creation. They added visual effects to the project or saved the results in different formats and different sources. Out of all students who failed the final exam 8% tried unsuccessfully to solve the tasks, and around 30% did not try to work on the project.

5. Conclusions. Using the concept map approach helped us to create a model of the learning content connected with the subject area, that is more structured, visible and helpful for the students. The complexity of tasks is growing from more trivial and easier assignments to more specific and complex ones. So it is much easier for us to check what parts of the contents are completed by the students and what level of cognitive learning is achieved. Taking a look at the CM (Figure 2), we can make the following conclusions:

1) Most of the students were able to complete the project inclusive task 8 that measures the applying and the analyzing levels.

2) Implementing a more complex algorithm turned out to be difficult for most of the students—they were reluctant to try tasks 9 and 10 which measure the highest cognitive learning levels (analyzing and evaluating).

3) Only 3% of the students were able to complete the project.

4) Concepts from C++ CM such as *bitwise operators*, *recursion*, are the most critical parts in the project.

5) Concepts from mathematics CM such as *fast power algorithms*, *prime root*, *discrete logarithm* and *quotient field* are the breaking points and only half of the students were able to complete the tasks connected with them.

The project gives an opportunity for the students to work in multidisciplinary environment and to apply knowledge from one subject area into another improving their study skills, to develop a real project during their first semester of education, which can be used in their portfolio and last but not least—to work

in a team and create basic libraries, that can be reused in other courses in their studies, e. g., Cryptography.

We are planning to further pursue this project-based learning idea and develop concept maps for other math projects such as "Analytical geometry", "Permutations" and "Finite automaton".

REFERENCES

[1] Anohina A., D. Pozdnakovs, J. Grundspenkis. Changing the Degree of Task Difficulty in Concept Map Based Assessment System. In: Proceedings of the IADIS International Conference "e-Learning 2007", Lisbon, Portugal, 2017, 443–450.

[2] Raykova M., H. Kostadinova, K. Kalpakchieva-Dushanova. Concept Maps Used as Accumulative Test Items Generator. In: 11th Annual International Conference on Computer Science and Education in Computer Science, Boston, USA, 2015, 28–35.

[3] Raykova M., H. Kostadinova, G. Totkov. Concept Maps Generator Based on Accumulative Test Items. In: 12th Annual International Conference on Computer Science and Education in Computer Science, Fulda, Nürnberg, Germany, 2016, 159–175.

[4] Raykova M., S. Boev. Enhanced Teaching and Learning Informatics and Mathematics Through Integration. In: 13th Annual International Conference on Computer Science and Education in Computer Science, Albena, Bulgaria, 2017, 105–122.

[5] Raykova M., S. Boev. How to improve teaching in discrete mathematics via programming and vice versa. In: 14th Annual International Conference on Computer Science and Education in Computer Science, Boston, MA, USA, 2018, 211–228.

[6] Asenova P., M. Marinov. System of Tasks in Mathematics Education. *Mathematics and Informatics*, **62** (2019), No 1, 53–71.

[7] Cormen T. H., C. E. Leiserson, R. L. Rivest, C. Stein. Introduction to Algorithms. 2nd ed., McGraw-Hill and MIT Press, 2001.

[8] Lippman S. B., J. Lajoie, B. E. Moo. C++ Primer. 5th ed., Addison-Wesley Professional, 2012.

[9] Stroustrup B. The C++ Programming Language. AT&T Labs Murray Hill, New Jersey, 1997.

[10] Eckel B. Thinking in C++. 2nd ed., Prentice Hall, 2000.

[11] Anderson L. W., Krathwohl D. R. A Taxonomy for Learning, Teaching, and Assessing. Abridged ed., Boston (MA), Allyn and Bacon, 2001.

[12] Moreira M. A. Why Concepts, Why Meaningful Learning, Why Collaborative Activities and Why Concept Maps? *Aprendizagem Significativa em Revista/Meaningful Learning Review*, **1** (2011), No 3, 1–11.

[13] Anderson L. W., D. R. Krathwohl (eds). A taxonomy for learning, teaching and assessing: A Revision of Bloom's Taxonomy of Educational Objectives. Complete edition, New York, Longman, 2001.

[14] Novak D. J. Learning, creating, and using knowledge: Concept maps as facilitative tools in schools and corporations. Mahwah, Lawrence Erlbaum Associates, 1998.

[15] Novak D. J., A. J. Canas. The Theory Underlying Concept Maps and How to Construct Them. Technical Report IHMC CmapTools 2006-01. Florida Institute for Human and Machine Cognition, 2006.

Mariyana Raykova, Stoyan Boev
Informatics Department
New Bulgarian University
21, Montevideo Blvd
1618 Sofia, Bulgaria
e-mail: {mraykova,stoyan}@nbu.bg

Received September 21, 2020
Final Accepted December 18, 2020