Computer Adaptive Testing of Student Knowledge

Sanja Maravić Čisar[1], Dragica Radosav[2], Branko Markoski[2], Robert Pinter[1], Petar Čisar[3]

[1]Subotica Tech, Department of Informatics
Marka Oreškovića 16, 24000 Subotica, Serbia
sanjam@vts.su.ac.rs, probi@vts.su.ac.rs

[2]Technical Faculty "Mihajlo Pupin", Department of Informatics
Đure Đakovića bb, 23000 Zrenjanin, Serbia
radosav@zpupin.tf.zr.ac.rs, markoni@uns.ac.rs

[3]Telekom Srbija, Subotica
Prvomajska 2-4, 24000 Subotica, Serbia
petarc@telekom.rs

Abstract: Technological progress, responsible for the declining costs of computers, coupled with the advancement of computer adaptive software have promoted computer adaptive testing (CAT) in higher education, offering alternatives to the conventional paper and pencil examinations. The CAT testing process, statistically conducted through Item Response Theory, is able to react to the individual examinee, keeping examinees on target with test items of an appropriate level of difficulty. The basic goal of adaptive computer tests is to ensure the examinee is supplied questions that are challenging enough for them but not too difficult, which would lead to frustration and confusion. The paper presents a CAT system realized in MATLAB along with its development steps. The application can run from a Matlab command window, or it is possible to make a stand-alone application that does not require the installation of Matlab. The questions are written in a .txt file. This allows the examiner to easily modify and extend the question database, without specific knowledge of the syntax of any programming language. The only requirement is for the examiner (but it is only required) to follow a pre-determined format of question writing. The program enables the testing of student knowledge in C++.

Keywords: computer adaptive testing; Item Response Theory; e-assessment

1 Introduction

Testing is one of the most common ways of knowledge testing. The main goal of testing is to determine the level of a student’s knowledge of one or more subject areas in which knowledge is checked. Different methods of knowledge evaluations are in use, such as in-class presentations, writing essays, projects, etc. However, the most common “tool” that is used to test knowledge is the test and oral exam. Since the computer as a teaching tool has been in use more and more in recent decades, and since its use has spread to all levels of education, the computer-based test has become very popular.

Out of all testing methods available today, computer adaptive testing provides the maximal balance of accuracy and efficiency. Over the past few decades, CAT has been used extensively in the areas of education, certification, and licensure [3]. This paper presents a computer adaptive test that was realized with the software package Matlab. The application was done in Matlab based on the program code that can be found at the web address [6]. The original code presents a computer adaptive test for GRE (Graduate Record Exam) and enables questions of the following types: analogy, antonym, and fill in the blanks. It was modified to allow for testing of the basic concepts of C++ in the form of multiple choice questions.

The remainder of this paper is organized as follows: Section 2 briefly reviews the theoretical basis of computerized adaptive tests, along with its benefits and drawbacks. Some basic concepts of Item Response Theory are presented in Section 3, as this is the theoretical foundation behind CAT. Section 4 gives a description of the application. Finally, some future research topics are suggested in Section 5.

2 Theoretical Basis of Computerized Adaptive Tests

CAT (Computerized Adaptive Testing) is a type of test developed to increase the efficiency of estimating the examinee’s knowledge. This is achieved by adjusting the questions to the examinee based on his previous answers (therefore often referred to as tailored testing) during the test duration. The degree of difficulty of the subsequent question is chosen in a way so that the new question is neither too hard, nor too easy for the examinee. More precisely, a question is chosen for which it is estimated, with a probability of 50% that the examinee would answer correctly. Of course, the first question cannot be selected in this way because at this point nothing is known about the examinee's capabilities (a question of medium difficulty is chosen), but the selection of the second question can be better adapted to each examinee. With every following answered question, the computer is increasingly better able to evaluate examinee’s knowledge.

Some benefits of the CAT are [9] as follows: (a) Tests are given "on demand" and scores are available immediately, (b) Neither answer sheets nor trained test administrators are needed. Test administrator differences are eliminated as a factor in measurement error. (c) Tests are individually paced so that an examinee does not have to wait for others to finish before going on to the next section. Self-paced administration also offers extra time for examinees that need it, potentially reducing one source of test anxiety. (d) Test security may be increased because hard copy test booklets are never compromised. (e) Computerized testing offers a number of options for timing and formatting. Therefore it has the potential to accommodate a wider range of item types. (f) Significantly less time is needed to administer CATs than fixed-item tests since fewer items are needed to achieve acceptable accuracy. CATs can reduce testing time by more than 50% while maintaining the same level of reliability. Shorter testing times also reduce fatigue, a factor that can significantly affect an examinee's test results. (g) CATs can provide accurate scores over a wide range of abilities while traditional tests are usually most accurate for average examinees.

Despite the above advantages, computer adaptive tests have numerous limitations, and they raise several technical and procedural issues [9]: (a) CATs are not applicable for all subjects and skills. Most CATs are based on an item-response theory model, yet item response theory is not applicable to all skills and item types. (b) Hardware limitations may restrict the types of items that can be administered by computer. Items involving detailed art work and graphs or extensive reading passages, for example, may be hard to present. (c) CATs require careful item calibration. The item parameters used in a paper and pencil testing may not hold with a computer adaptive test. (d) CATs are only manageable if a facility has enough computers for a large number of examinees and the examinees are at least partially computer-literate. This can be a great limitation. (e) The test administration procedures are different. This may cause problems for some examinees. (f) With each examinee receiving a different set of questions, there can be perceived inequities. (g) Examinees are not usually permitted to go back and change answers. A clever examinee could intentionally miss initial questions. The CAT program would then assume low ability and select a series of easy questions. The examinee could then go back and change the answers, getting them all right. The result could be 100% correct answers which would result in the examinee's estimated ability being the highest ability level.

The CAT algorithm is usually an iterative process with the following steps:

1 All the items that have not yet been administered are evaluated to determine which will be the best one to administer next given the currently estimated ability level

2 The "best" next item is administered and the examinee responds

3 A new ability estimate is computed based on the responses to all of the administered items.

4 Steps 1 through 3 are repeated until a stopping criterion is met.

The flowchart below serves as an illustration of the CAT algorithm.

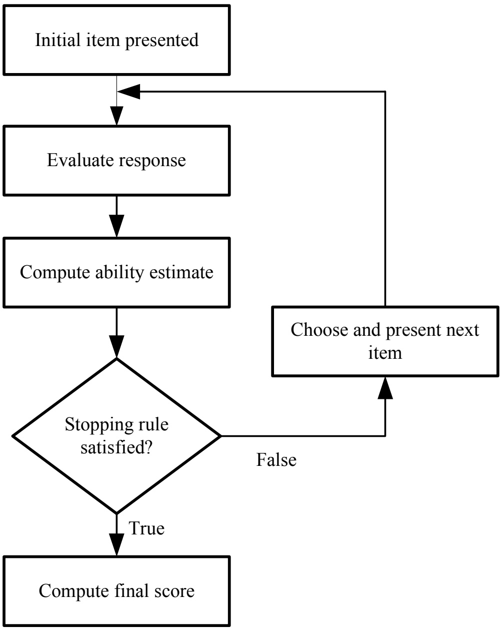


Figure 1
Illustration of the CAT algorithm

Several different methods can be used to compute the statistics needed in each of these three steps, one of them is Item Response Theory (IRT). IRT is a family of mathematical models that describe how people interact with test items [2].

According to the theory of item response, the most important aim of administering a test to an examinee is to place the given candidate on the ability scale [5]. If it is possible to measure the ability for every student who takes the test, two targets have already been met. On the one hand, evaluation of the candidate happens based on how much underlying ability they have. On the other hand, it is possible to compare examinees for purposes of assigning grades, awarding scholarships, etc.

The test that is implemented to determine the unknown hidden feature will contain N items, and they all measure some aspect of the trait. After taking the test, the person taking the test responds to all N items, with the scoring happening dichotomously. This will bring a score of either a 1 or a 0 for each item in the test. Generally, this item score of 1 or 0 is called the examinee's item response. Consequently, the list of 1's and 0's for the N items comprises the examinee's item response vector. The item response vector and the known item parameters are used to calculate an estimate of the examinee's unknown ability parameter.

According to the item response theory, maximum likelihood procedures are applied to make the calculation of the examinee's estimated ability. Similarly to item parameter estimation, the afore-mentioned procedure is iterative in nature. It sets out with some a priori value for the ability of the examinee and the known values of the item parameters. The next step is implementing these values to compute the likelihood of accurate answers to each item for the given person. This is followed by an adjustment to the ability estimate that was obtained, which will in turn improve the correspondence between the computed probabilities and the examinee's item response vector. The process is repeated until it results in an adjustment that is small enough to make the change in the estimated ability negligible. The result is an estimate of the examinee's ability parameter. This process is repeated separately for each person taking the test. Nonetheless, it must be pointed out that the basis of this process is that the approach considers each examinee separately. Thus, the basic problem is how the ability of a single examinee can be estimated.

The estimation equation used is shown below:

$$\hat{\theta}_{s+1} = \hat{\theta}_s + \frac{\sum_{i=1}^{N} - a_i[u_i - P_i(\hat{\theta}_s)]}{\sum_{i=1}^{N} a_i^2 P_i(\hat{\theta}_s) Q_i(\hat{\theta}_s)} \tag{1}$$

where: $\hat{\theta}_s$ is the estimated ability of the examinee within iteration *s*, a_i is the discrimination parameter of item *i*, i = 1, 2,... .N.

- u_i is the response made by the examinee to item *i*: $u_i = 1$ for a correct response, $u_i = 0$ for an incorrect response.
- $P_i(\hat{\theta}_s)$ is the probability of correct response to item *i*, under the given item characteristic curve model, at ability level $\hat{\theta}$ within iteration *s*.
- $Q_i(\hat{\theta}_s)=1 -P_i(\hat{\theta}_s)$ is the probability of incorrect response to item *i*, under the given item characteristic curve model, at ability level $\hat{\theta}$ within iteration *s*.

The CAT problems have been addressed before in the literature [1], [4], [5].

3 Computer Adaptive Tests Based on IRT

For computer adaptive tests which implement IRT (Item Response Theory) a relatively large base of questions for a given task is developed and their informational functions are defined. A well-formed question bank for CATs contains questions that together provide information through a whole range of properties (θ). The examinee starts the test with an initial estimate of theta (θ), which may be identical for each examinee, or it may be used as predefined information available on the candidate (e.g. results attained in other tests, marks or information from the professor). The question is administered on the basis of the initial theta estimate and immediately evaluated by the computer that generated the test.

3.1 Question Selection

With computer adaptive tests (CAT) based on IRT the subsequent question is selected on the basis of the examinee's scored answers to all previously set questions. In the initial phase of CATs, though only the first or first two questions have been evaluated, the subsequent question is selected based on the rule of "step" – if the first question was answered correctly, the examinee's previous theta estimate will be increased by some value (e.g. 0.50); while, if the first given answer was incorrect, the original theta estimate will be decreased by the same value. As the test continues, an answer sheet is generated which consists of at least one correct and one incorrect answer to the question, thus the MLE (*Maximum Likelihood Estimation*) is used to calculate the new theta estimate, which is based on all the answers that the examinee has given up to that point in the test [11].

After each processed question, the new theta estimate is used for selecting the next question. That question is an un-administered question from the question bank that provides the most information for the currently estimated theta value. Figures 2, 3, and 4 illustrate the "maximum information" questions selected in the computer adaptive test. Figure 2 presents information functions for 10 questions, for the initial theta estimate for a fictitious examinee (indicated by a vertical line). This value is presented at 0.0, which is the mean value of the theta scale. The values of information are calculated for all questions for that theta level. Figure 2 shows that Question 6 provides the most information of the 10 questions for theta = 0.0. Thus, Questions 6 is processed and evaluated [11].

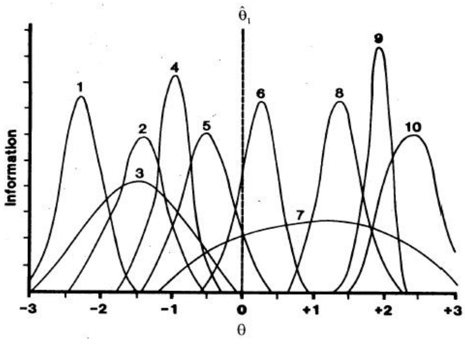


Figure 2
Information functions for the 10 questions [11]

Based on this score (incorrect answer, in this case), the new theta value is defined with a step 1.0, and thus now it is -1.0. Based on the rule of question selection with maximum information, Question 4 was selected (Figure 3) because at the given theta level it contains the most information, and it is evaluated. Given the assumption that the answer to Question 4 is correct, the MLE can be used for the new theta estimate.

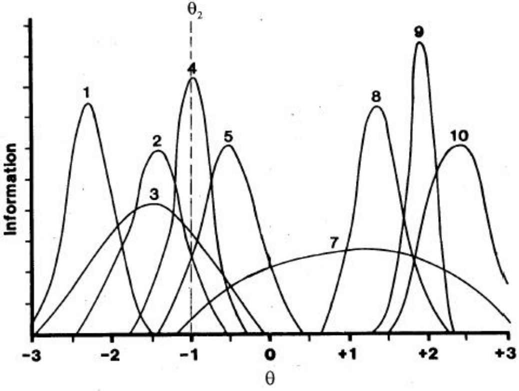


Figure 3
Information functions for 9 questions [11]

The result is theta = -0.50. Again, by selecting the question based on the (Figure 4). The evaluation, theta estimation and question selection continues until the criterion for termination is not met [11].

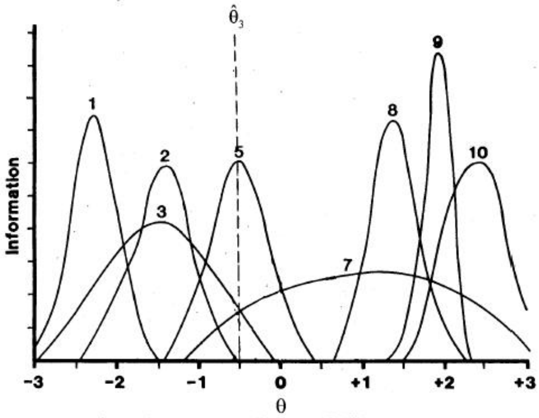


Figure 4
Information functions for 8 questions [11]

3.2 Termination of the Computer Adaptive Test

One of the most important properties of these adaptive tests is the criterion of discontinuing the test may vary depending on different goals of the test. Some tests are used for selection or classification, e.g. whether the subject has managed the acquisition of a certain unit of the learning material, which student will be admitted for secondary school or university, or who will be chosen for a job. Other tests are used for counseling or clinical purposes. The goal of such tests is determine the abilities of the subjects as well as possible. In the context of adaptive tests, these two aims are realized by the two different rules of test termination.

The aim of the classification is that the candidate's results are compared with some cutoff value. The aim is to create the most precise classification. In order for this to be implemented in the context of computer adaptive tests, the theta estimate and its standard error measurement is used. The candidate is classified as above the cutoff value (expressed on the theta scale) if the theta estimate as well as its 95% confidence interval (calculated as ± two standard error measurement) is above or beneath the cut score. As CAT can evaluate this decision after every processed question, the test can be terminated when this condition is met. The result of this test will be the sum of the classification made for the group of examinees where all will have a 5% error rate. The error rate can be controlled by the size of the SEM confidence interval around the theta estimate.

When CATs are not used for classification, a different rule is applied for the termination of the test. In that case it is advisable to evaluate every examinee to the desired level of precision, which is determined in advance by the level of standard error measurement.

This will results in the sum of "equally precise" evaluations, so that all examinees will have results which are equally precise, thus defining a new concept, "fair test". In order to implement equally precise evaluation, CAT enables the user to specify the level of the SEM desired for every examinee. Assuming that the question bank contains enough questions correctly spread along the theta scale and it is possible to continue the test long enough for the examinee, this goal will be realized if the test is terminated when the given SEM level is achieved [11].

3.3 Development of a CAT

According to [3] the final pool of items should consist of approximately five to ten times the number of items that an examinee will eventually see when taking the test. Thus, for a 30-item test, an item bank of 150-300 quality items is highly recommended. Item writing, in and of itself, is a tedious and rigorous process. Developing the initial set of items that will eventually be reduced through the analysis process is a major undertaking, as upwards of 400 items may be needed in order to get to a final pool of 150-300 items.

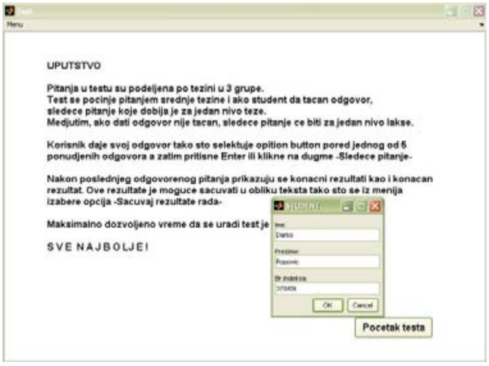
Once the initial item pool is established, data is collected on each item. IRT analyses typically require at least 300 data points for each item, with 500 being preferred. Since it is not advisable to attempt to get 300 people to complete all items in the initial item pool, often the items have to be split into sub-pools small enough to collect accurate data.

With a sufficient sample size of examinees, the item parameters (discrimination, difficulty, and guessing) can be estimated. These parameters are used to determine which items will be retained in the final item pool, and which items will be revised or discarded. The final pool is then entered into the CAT system, which then creates optimal item selection paths for test takers.

4 Description of the Application

The program that can be found at the web address [6] presents a computer adaptive test and was modified to enable the testing of student knowledge in C++. The application can run from a Matlab command window, or it is possible to make a stand alone application that does not require the installation of Matlab. The MATLAB and Simulink product families are fundamental computational tools at the world's educational institutions. Adopted by more than 5000 universities and colleges, MathWorks products accelerate the pace of learning, teaching, and research in engineering and science. MathWorks products also help prepare students for careers in industry, where the tools are widely used for research and development [10]. Some examples of implementing Matlab as an educational tool can be found in [7], [8].

After starting the program the main window is displayed as is the dialog box for entering basic data on the student (name, surname and index number). Pressing the Enter command button starts the test, as shown in Figure 5.



UPUTSTVO

Pitanja u testu su podeljena po tezini u 3 grupe.
Test se pocinje pitanjem srednje tezine i ako student da tacan odgovor,
sledece pitanje koje dobija je za jedan nivo teze.
Medjutim, ako dati odgovor nije tacan, sledece pitanje ce biti za jedan nivo lakse.

Korisnik daje svoj odgovor tako sto selektuje opition button pored jednog od 5
ponudjenih odgovora a zatim pritisne Enter ili klikne na dugme -Sledece pitanje-

Nakon poslednjeg odgovorenog pitanja prikazuju se konacni rezultati kao i konacan
rezultat. Ove rezultate je moguce sacuvati u obliku teksta tako sto se iz menija
izabere opcija -Sacuvaj rezultate rada-

Maksimalno dozvoljeno vreme da se uradi test je

S V E N A J B O L J E !

Ime: Dario

Prezime: Popovic

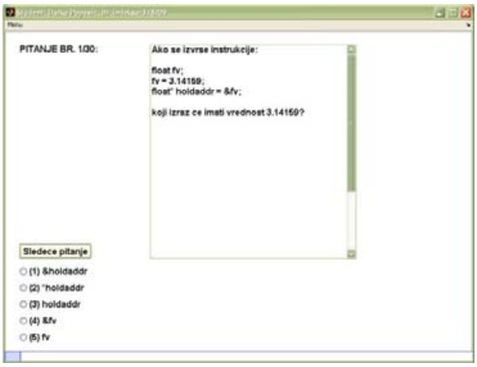
Br indeksa: [illegible]

Figure 5

Startup screen

After pressing the button *Pocetak testa* (Start), the function *pocetak_testa* (test_start) is called and the visibility of objects that are no longer needed has to be set to "off" and the visibility of the edit control (for question displaying), option buttons (for showing multiple choices as answers) and patch object is set to "on". Then the function *final_test* is called, which has two output parameters: an array with correct/incorrect answers (in this case 30) and the second parameter is an array which contains the time (given in seconds) that has elapsed since the student has given the answer for each questions.

After registering for the test a new window opens with the first question. At all times the student can see on the screen which question the student is on, the total number of questions, the text of the question with multiple choice answers, as can be seen in Figure 6. At the bottom of the screen there is a progress bar which illustrates the progress of the student during the test.



PITANJE BR. 1/30:

Ako se izvrse instrukcije:

```
float fv;
fv = 3.14159;
float* holdaddr = &fv;
```

koji izraz ce imati vrednost 3.14159?

○ (1) &holdaddr
○ (2) *holdaddr
○ (3) holdaddr
○ (4) &fv
○ (5) fv

Figure 6
Screenshot of a question

```
function [ans_array,time_arr]=final_test
[a b c t1]=ask_qn(1,1,grupa_pitanja,ones(4,4),1);
             ans_array = b;
              time_arr = t1;
end
```

As shown, the next function that is called is the function *ask_qn* which has five input parameters and four output parameters. In the function *ask_qn* everything is handled in one *for* loop which is repeated as many times as there are questions *(comm_arr*). The first calculation is for the determination of the question's difficulty that needs to be answered.

The questions are divided based on their difficulty into three groups, easy, medium and difficult question (parameter *question_set* could be 1, 2 or 3).

```
deciding_factor=ask_1;
```

```
question_set=normalize_qno(question_set,deciding_factor,1,3);
```

At the beginning, the parameter *question_set* is 1 and also the parameter *ask_1*. The parameter *ask_1* determines by how much to increase or decrease the parameter *question_set*. In this case, the test starts with a question of medium difficulty, which is assigned in test the results with number 2. If the student gives a correct answer to this question, the algorithm of the test passes to the first question in the group of difficult questions (assigned number 3), and if the answer given to the first question is incorrect then the group with easy questions is selected (assigned number 1). The questions are written in a txt file and they are invoked by calling the appropriate function in the program. This allows the examiner to easily modify and extend the question database, without knowledge of the syntax of any programming language; it is only required to follow a determined format of questions writing.

Also, the type of question that will be selected as the next question is determined. In this test there is only one type of questions (MCQ), but it is also possible to set some other types of questions (analogy, antonym, fill in the blanks etc.). So, in this case the array *com_arr(i)* consists of only the ones.

```
ask_1_char_type=question_type(com_arr(i));

ask_1_char_type = 'pitanja'

function question_str=question_type(number)

if(number==1)

    question_str='pitanja';

end
```

The next parameter that is necessary to obtain is *question_status*, which contains data in form of a matrix (question difficulty and the type of question). In the case of questions with a difficulty of level three and only one type of the question, it would be a 1-dimensional array initialized with the *ones (1,3)*. After that the function *ask* is called:

```
[ask_1 q_time]=

=ask(ask_1_char_txt,question_status(com_arr(i),question_set))
;
```

The first parameter of the function gives the information from which .txt file to read the questions, and the second parameter *question_status (com_arr(i),question_set))* obtains the information from which line in the .txt file to start reading. The output parameters are placed in variables *qn_time* and *ans_array*:

```
qn_time=[qn_time q_time];

ans_array=[ans_array ask_1];
```

The next step is to start measuring the time that passes before the student selects any of the five given options as answer. The elapsed time is recorded in the variable *q_time=[q_time toc]*; there is verification whether the given answer is correct (`if(taster==num_tline)`) and if it is, the related variable is set to 1.

```
        q_time=[q_time toc];
         if(taster==num_tline)
                  output_check=1;
                end
end
```

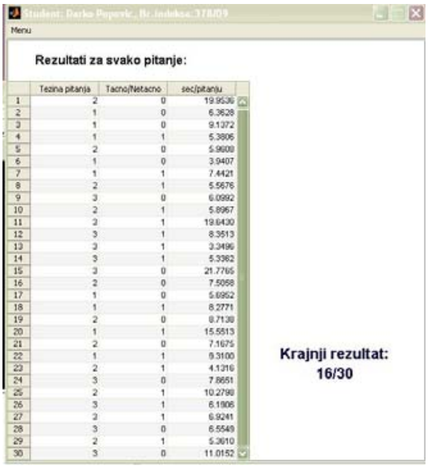
Once the student has given answers to all questions, the program continues to run in the function *pocetak_testa* from the part where the function was called:

```
[a b]=final_test;
```

where *a* is the array with answers and *b* is the array with the time elapsed per each question. The final result is calculated with the call of the function *totaling* with the parameter *a*.

```
total_marks=int2str(totaling(a));
```

After answering the last question, the examinee can see their results immediately on the screen. If the examinee selects the option to save the test results, the appropriate function parameters are called. From the text file can be seen the level of the question's difficulty, whether or not the answer was correct or incorrect, and the time needed for answering each question (i.e. until pressing the command button Next question/Show results).



Student: Darko Popović, Br. indeksa: 378/09

Rezultati za svako pitanje:

	Tezina pitanja	Tacno/Netacno	sec/pitanju
1	2	0	19.9536
2	1	0	6.3628
3	1	0	9.1372
4	1	1	5.3806
5	2	0	5.9608
6	1	0	3.9407
7	1	1	7.4421
8	2	1	5.5676
9	3	0	6.0992
10	2	1	5.8967
11	2	1	19.6430
12	3	1	8.3513
13	3	1	3.3496
14	3	1	5.3362
15	3	0	21.7765
16	2	0	7.5058
17	1	0	5.6952
18	1	1	8.2771
19	2	0	8.7138
20	1	1	15.5513
21	2	0	7.1675
22	1	1	9.3100
23	2	1	4.1316
24	3	0	7.8651
25	2	1	10.2798
26	3	1	6.1906
27	3	1	6.9241
28	3	0	6.5549
29	2	1	5.3610
30	3	0	11.0152

Krajnji rezultat:
16/30

Figure 7

Test results view

Conclusions

Computerized adaptive tests offer many advantages over conventional paper and pencil tests: efficiency in the form of reduced testing time; appropriate challenge for the individual examinee’s ability level; flexibility in arranging testing time and location; the potential for the use of sophisticated dynamic graphics via the computer; immediate scoring; the potential for the synchronous collection of data during testing and so on [12].

This paper reports on the use of a computer adaptive test for examining a student’s knowledge in C++. The motivation behind this work was to investigate techniques for the improvement of student assessment. Future work will involve the further analysis of the test statistics and the improvement of the classification of questions based on the student's test results.

References

[1] Baker, F. “The Basics of Item Response Theory”, Chapter 5, Estimating an Examinee’s Ability, http://echo.edres.org:8080/irt/baker/chapter5.pdf, 2001

[2] Embretson, S. E., Reise, S. P., “Item Response Theory for Psychologists”, Mahwah NJ, Lawrence Erlbaum Associates, 2000

[3] Fetzer, M., Dainis, A., Lambert, S., Meade, A., PreVisor’s PreViewTM Computer Adaptive Testing (CAT) in an Employment Context, April 2008, White Paper, www.previsor.com/pdf/WPCAT.pdf

[4] Ju, Gin-Fon N., Bork, A., “The Implementation of an Adaptive Test on the Computer”, Fifth IEEE International Conference on Advanced Learning Technologies (ICALT'05), Kaohsiung, Taiwan, 2005, pp. 822-823, http://doi.ieeecomputersociety.org/10.1109/ ICALT.2005.274

[5] Kardan, S., Kardan, A., ” Towards a More Accurate Knowledge Level Estimation”, 2009 Sixth International Conference on Information Technology: New Generations, Las Vegas, Nevada, pp. 1134-1139, 2009, http://doi.ieeecomputersociety.org/10.1109/ITNG.2009.154

[6] Mathworks, Computer Adaptive Test Demystified, http://www.mathworks.com/matlabcentral/fileexchange/12467-computer-adaptive-test-demystified-gre-pattern

[7] Sadeghi, S., Mirsalim, M., Hassanpour Isfahani, A., “Dynamic Modeling and Simulation of a Switched Reluctance Motor in a Series Hybrid Electric Vehicle”, Acta Polytechnica Hungarica, 2010, Vol. 7, No. 1, pp. 51-71

[8] Takács, M., Szakál, A., Csikós Pajor, G., “Software Supported Mathematics Teaching”, Proceedings of the 3rd International Conference on Information Technology Based Higher Education and Training, ITHET Conference, July 4-6, 2002, Budapest, Hungary

[9] http://echo.edres.org:8080/scripts/cat/catdemo.htm. Rudner, L., An On-line, Interactive, Computer Adaptive Testing Tutorial

[10] http://www.mathworks.com/academia/

[11] http://www.psych.umn.edu/psylabs/catcentral/

[12] www.rpi.edu/~faheyj2/SB/diss1.doc. Bringsjord, E., “Computer-Adaptive Versus Paper-and-Pencil Testing Environments: An Experimental Analysis of Examinee Experience”