1980

Design and Use of a Computerized Test Generating Program

Edward Schaefer

Laurence A. Marschall
Gettysburg College

Schaefer, Edward, and Laurence A. Marschall. "Design and Use of a Computerized Test Generating Program." American Journal of Physics (1980) 48(7):518-523.

Design and Use of a Computerized Test Generating Program

Abstract
An easy-to-use set of programs for the computerized generation of multiple-choice and easy examinations in an introductory astronomy course is described. The programs allow the user to establish files of test questions and to rapidly assemble printed copies of examinations suitable for photocopying. Written in ALGOL for a Burroughs B6700 computer, the programs can, in principle, be implemented on large mainframe computers or on microcomputers of a size increasingly available to physics departments. The advantages and costs of computerized test generation are discussed.

Keywords
education, educational tools, testing, computers, Burroughs computers, computer-aided design

Disciplines
Computer Sciences | Physics

Design and use of a computerized test generating program

Edward Schaefer[a] and Laurence A. Marschall
Department of Physics, Gettysburg College, Gettysburg, Pennsylvania 17325
(Received 27 September 1979; accepted 15 January 1980)

An easy-to-use set of programs for the computerized generation of multiple-choice and essay examinations in an introductory astronomy course is described. The programs allow the user to establish files of test questions and to rapidly assemble printed copies of examinations suitable for photocopying. Written in ALGOL for a Burroughs B6700 computer, the programs can, in principle, be implemented on large mainframe computers or on microcomputers of a size increasingly available to physics departments. The advantages and costs of computerized test generation are discussed.

I. INTRODUCTION

Not so long ago versatile interactive computer systems were available to only a few teachers at large institutions. The revolution of microelectronics has changed all that. Computers of considerable power are becoming common at schools of all sizes. It is rare to find a physics department without a microcomputer rivaling the mainframe computer of a decade ago (in memory size at least). Terminals abound, and access to the larger time-sharing systems is becoming more common as the cost of computing decreases and as operating systems become more user oriented.

As a result, many computer applications that were dismissed as being too costly or too time and space consuming are now open to a large number of physics teachers. This paper reports on one such application, the use of a computer to assemble examinations for an introductory astronomy course. The success of this program and the ease with which it was implemented prompt us to encourage the development of similar programs by those who might have been reluctant to adopt computerized techniques in the past.

The programs we will describe were written to run on a larger time-sharing system, the Burroughs B6700 at Gettysburg College, but because we feel that the key element in successful computer use is convenience and operating efficiency, and because computer power is so rapidly decreasing in cost, we believe our experience will be of value to a wide range of physics and astronomy instructors at a wide range of institutions.

We shall first describe the logic and operation of TORQUEMADA, a system designed to store and retrieve multiple choice and essay questions. We will then discuss some of our experiences with the system and our recommendations for the design of similar systems.

II. DESIGN OF THE PROGRAMS

TORQUEMADA[1] is a series of programs designed to assist in the production of printed examinations. It has been used successfully for one year in the introductory astronomy course at Gettysburg College. The programs enable the user, operating from a terminal keyboard, to store large numbers of questions on magnetic files; questions may be retrieved at random from these files and a finished test, suitable for reproduction, can be printed on a typewriter terminal attached to the computer. It is, in short, an automated test assembler and editor.

A large number of options are available to the user: Questions may be multiple choice (with four, five, or possible answers) or essay type; any number of diff item files may be established and questions from diff files may be mixed together on the finished examinat multiple quizzes may be made from the same set of q tions with the order of the questions and answer ch permuted from one exam to the next; questions ma printed with blank spaces following to provide room drawings or lengthy answers. The program also perfo "bookkeeping" functions for the user, recording the d a question was last used on an exam, and providing keys multiple choice examinations.

The overall strategy of the quiz-making process is sho in Fig. 1. There are three programs: The first, TORQUE ADA/FILES produces files of questions and stores them the computer. A second program TORQUEMADA/QUIZ selects questions at random from specified categories, inserts special formating and editing instructions automatically (such as a command to insure that questions are not broken across page boundaries), and stores the assembled examination as a "quiz file" in the computer. The final program is a "word-processing" program called RUNOFF, originally developed at Dartmouth College, which was supplied with our college's computer system. The word processor reads the quiz files and produces a finished copy of the exam on the typewriter or the high-speed printer.

We shall only discuss the operation of the first two programs, which were written by the authors. Text-editing and word-processing programs are becoming increasingly common on both large and small machines. Several are available in BASIC, written especially for microcomputers; anyone writing a quiz-generating program such as ours should have little difficulty finding a text-editor to suit his needs.

A. File-making program

Since the establishment of computer files is largely a routine task, the file-making program was designed with speed and simplicity in mind. The user "converses" with the program through a terminal keyboard, giving the computer commands (see Appendix B) to perform all the secretarial functions needed to maintain the question file. To establish a file of questions, the user first identifies the file by typing in its name (e.g., "Astronomy/1/4 choice" might be a file of four-option multiple-choice questions on basic astronomy). He or she then specifies the type of questions that will be stored in the file: four-, five-, or six-option multiple choice

 0002-9505/80/070518-05$00.50

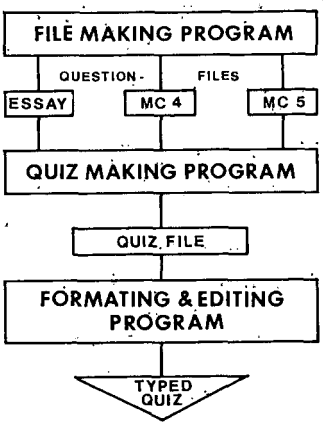

Fig. 1. Test generating scheme of TORQUEMADA.

questions or essay questions. Once a file is "configured" in this fashion, the program will only accept or retrieve questions of the specified type from that file. The user next types in a list of categories to be used in the file. Each category is assigned a title and a number. The program accepts category numbers from 1.00 to 99.98 and will store up to 100 questions under each category, giving the user the option of almost 10 000 categories and one million questions. In practice, the files are much smaller. We have adopted a category system similar to that proposed by Turner,[3] and our astronomy files currently contain over a thousand questions in about 100 categories coded by subject. One of the digits of the category number can be used, if desired, to indicate the behavioral objectives[4] or difficulty level of the question.

Once the categories for the file have been established, questions may be stored at any time by simply typing in an ADD command and a category number. The computer then asks the user to enter the question and, in the case of multiple choice questions, to supply choices for the answer. The question is automatically assigned a number under the specified category and stored in the computer. Additional commands enable the user to delete a question or an entire category, to store the same question under several different category numbers, to list questions (or the entire file) for review, and to edit or replace lines of stored text. Once the basic command system of Appendix B is learned, a question can be entered into the computer in less than a minute. Supplied with a suitable list of questions, we have been able to establish a file of several hundred items in less than a week of intermittent typing.

B. Quiz-making program

This program may be used to generate an examination from any file or set of files that has already been established with the previous program. Unlike the file-making program, the quiz-making program does not have a command structure. Instead, the computer carries on a dialogue with the user, enabling him to select the questions and the format desired for the final printed text. If the questions are multiple choice, the answers to the questions are shuffled automatically. The order of the individual questions is also randomized by the computer, but the user is given the option of shuffling or not shuffling together questions from more than one file. (The latter option is useful if it is desired to pitch a selection of essay questions from one file at the end of a set of multiple choice questions from another file.)

Headings and other items of text may also be inserted where desired. The program automatically inserts editing commands for the word-processing program. These provide for indenting of the questions, underlining, subscripting and superscripting, justification of margins, and blocking of the text so that questions are not split across page boundaries.

A sample dialog with the quiz-making program is shown below with the computer's questions in capital letters.

?NAME OF OUTPUT FILE

This specifies the name that will be assigned to the completed quiz file. Any name acceptable to the computer can be entered, e.g., "TEST1," "EXAMINATION/1/ECONOMICS," "SURPRISEQUIZ," etc.

?WILL THIS BE A SAMPLE TEST

This prevents the machine from updating the "last date used" record for the questions chosen. If the answer is YES, questions will not be updated; if it is NO, the current date will be written in the records for each question finally selected.

?NAME OF INPUT FILE

This must be the exact name of a question file already established in the computer.

HOW MANY CATEGORIES DO YOU WISH TO USE?

This provides for a random selection of questions by category. The user decides beforehand how many categories and questions he wishes from the file. This choice is not final; the program allows additional questions to be chosen later if not enough suitable questions are found in a first random search.

PLEASE INDICATE WHICH CATEGORIES YOU WISH TO USE, AND HOW MANY QUESTIONS YOU WANT FROM EACH.

The category code number and the number of questions from each category are entered separated by commas (and carriage returns if more than one line is needed). For example, "1.65,2,7.65,3,8.80,1" would request two questions from category 1.65, three from category 7.65, and 1 from category 8.80. The computer then proceeds to select questions at random from the question file until its "shopping list" is filled. Questions are presented on the screen, along with the last date that the question was used on an exam. The user may accept or reject each selection. If the question is rejected, a new question from the category will be chosen until an acceptable one is found or until all questions in the category have been presented.

ACCEPT OR REJECT?

If a question is accepted for inclusion in the exam, the user may also specify that a blank space follow the question for the inclusion of drawings or to leave space for an answer. The codes for acceptance and rejection are

A—Accept the question as is.

A:S—Accept the question followed by a short gap.

A:L—Accept the question followed by a long gap.

R—Reject the question and select another from the same category.

N—Reject this question and go on to the next category.

DO YOU WANT TO USE MORE CATEGORIES?

If the user wishes to select more questions at random, he commands the computer to return to the HOW MANY CATEGORIES?...PLEASE INDICATE WHICH

. . .AND HOW MANY QUESTIONS. stage of the program by answering YES here. Otherwise the computer will proceed onward.

DO YOU WISH TO USE ANY SPECIFIC QUESTIONS?

If the user responds with a YES here, the computer will ask HOW MANY? and ?WHICH ONES. The user specifies the exact numbers of the questions desired from the file. Since it is assumed that the user is familiar with the questions he specifies, only the first line of each question is presented in response to this command, in contrast to the random selection option where the entire question is displayed.

DO YOU WANT TO USE ANOTHER FILE?

A YES response enables the user to specify another file of questions that may be drawn upon for the exam. The computer will respond ?NAME OF INPUT FILE and questions can then be selected by category or by question number as before.

DO YOU WANT TO MIX ITS CONTENTS WITH THOSE OF THE LAST FILE?

A YES response to this question provides the option of mixing together questions from two or more files. A NO answer separates the chosen questions from the other questions in the examination. Essay questions, for instance, may be grouped together at the end of a set of multiple choice questions.

PLEASE INSERT THE HEADINGS FOR THE TEST.

TYPE "END" ON A NEW LINE WHEN FINISHED.

The user enters headings such as "Astronomy Test Number 1" and other text, such as instructions, which normally come at the beginning of an examination.

PLEASE INSERT THE TRAILERS FOR THE TEST.

TYPE "END" ON A NEW LINE WHEN FINISHED.

This is the last command before the completion of the quiz. Any final lines, such as PLEASE SIGN THE HONOR PLEDGE may be inserted here.

DO YOU WANT TO MAKE ANOTHER TEST WITH THE SAME QUESTIONS?

This offers the option of duplicating the test on a new file but with the questions in a different order, enabling multiple versions of the same exact exam to be give to a class to discourage cheating. The permuted exams are generated and stored in a matter of a few seconds.

After the multiple copies are made, or if the answer to this question is NO, the program terminates by writing the quiz file onto the disk of the computer. The test is now ready to be printed out on the typewriter or the printer using the RUNOFF program.

C. Printing out the finished exam

The operation of a text-editing program may vary from machine to machine. The RUNOFF editor which we used simply requires that the user type in the name of the quiz file which was generated by the quiz-making program. At Gettysburg College, a Diablo 1620 terminal provides high-quality copy which is photocopied and distributed to the class. The last page of the printout contains a list of the file numbers of the questions chosen and, in the case of multiple choice exams, a listing of the correct answers.

III. HARDWARE REQUIREMENTS AND COSTS

The programs we are currently using are written in ALGOL and designed to run on a Burroughs B6700 with two 40 megabyte disk files. There is no reason, however, why they could not be written in FORTRAN or BASIC and run on a smaller machine. There is a lower limit to the complexity of the machine required, however. Much of the usefulness of the programs derives from the computer's ability to access files rapidly and to provide output in typewritten form. A micro- or minicomputer system with floppy disc storage and a high-quality printer is the minimum required to run a similar test generator, in our estimation. Such systems are well within the budget of many physics departments; $2500 will buy a computer and disk of the required size. Since this sort of computer is already being used by many physics departments for computation or for running experimental apparatus, the high-quality printer (about $3000) is the only purchase required to produce computer-generated exams. Cheaper printers are available for less than $1000, but the economy-minded user may have to sacrifice useful features, such as lowercase font and the ability to subscript and superscript.

Test generation programs of the sort we describe here are not uncommon,[5] but never before has there been such an opportunity for the efficient and convenient implementation of such systems. The authors know, for instance, of a similar test-generation system running on an 8080-based microcomputer with 48K of storage and a floppy disk drive.[6] Computer-managed instruction on this scale is no longer a luxury item.

Turner[7] suggests that computer-based test files can be costly, but our experience is the opposite. Secretarial costs for typing are actually less than in precomputer days, because our program takes care of formating and the editing capabilities of the computer system make it easier to produce error-free copy.

The actual cost of computer time is not taken into account here. At our school, and at many other small colleges, computer time is not billed to the individual user. It is our experience that a lengthy quiz file may require several hours of input–output (I/O) time to establish (much less in real processing time). The assembling of a fifty-question exam takes about one hour in front of a terminal, but only about a minute of central processor (CPU) time.

IV. ADVANTAGES OF COMPUTER-GENERATED EXAMINATIONS

The obvious advantages of a computerized test-generating system are speed and convenience. Other benefits of the system, however, deserve particular mention.

Turner[8] points out that questions may be better written if they are not composed under the deadline of an approaching examination. In this regard, the computerized file is as useful as the card file he describes. We often enter questions into the computer after a class, when the material we have covered is fresh in our mind. Using a portable terminal, an instructor can even enter questions into the file whenever and wherever a good idea occurs to him.

Turner also stresses the value of careful item analysis of test questions. If the test-generating program is coupled to an analysis program, and if a mark-sense reader is used on multiple-choice questions, a great deal of time can be saved

in administering tests to large classes, resulting in an improved sensitivity to the effectiveness of the testing procedure. We should point out, however, that our system also accommodates essay questions, which are not amenable to such easy analysis.

There are several additional advantages of a test-generating program of the type we have described. First of all, the requirement that categories for the quiz be selected in advance enables the construction of more balanced tests than might otherwise be possible. The list of question categories used to select items for the exam provides a thematic overview of the test that can be comprehended at a glance. The program's requirement that the user approve each question promotes a further balance of emphasis. Since this system was completed we have received several unsolicited comments from students complimenting us on the comprehensiveness and fairness of our final examinations.

The ease with which tests can be assembled promotes the more frequent use of examinations, a process which, if wisely done, can stimulate the student to keep abreast of assignments while providing both student and teacher with useful diagnostic information. Practice exams can also be generated as study aids for student use. Make-up exams, which many are reluctant to offer simply because of the drudgery of making up new questions, can easily be assembled and administered to students who justifiably miss an exam.

The bookkeeping provided automatically by the program is also of value in providing a different sort of balance. In reviewing old examinations produced without the computer, we were surprised how often we had repeated questions word for word in consecutive years, usually without any intention to do so. Since the "last use" of each question is recorded and displayed during the quiz-making process, such repetition is less likely to occur. The user is also stimulated to provide a greater variety of questions for the file.

Security is a perennial hobgoblin in any computer system. Since the quiz files themselves are removed immediately after the assembled test is printed out, the only possible security problem might arise from unauthorized access to the question files. Such access is difficult; moreover the large number of questions on each file makes it more of a job to memorize the file than to learn the course materials. (Consequently we chose not to "scramble" the text of our files, although our computer system allows this to be done with relative ease.)

There are some who view the complete question files as more of a learning resource than as a temptation to academic fraud. We know of one professor who publishes a listing of the questions for his courses and distributes them through the campus bookstore. Not only does it encourage study of the material, he claims, but it also eliminates the unfair advantage of fraternity or sorority members who have access to extensive collections of back exams.[6]

Other features of the program further discourage cheating. Since permutations of the same exam can be produced in virtually the time it takes for the computer to type them, we find it easy to give multiple versions of an exam to large classes that are forced to sit close together in the lecture hall. We have also tested the randomness of question selection by administering two tests of *different* questions chosen from the *same* categories to two halves of an astronomy class of 60. The results were essentially the same. We take this to mean that, if the files are constructed wisely, make-up examinations can be produced with less unfairness to the student and less danger of cheating than might otherwise be possible.

There are limitations to the program we have described, of course. Inevitably there will be times when tests cannot be generated automatically because the computer or printer is down. Turner's card-file system is, in this regard, less vulnerable to the vagaries of modern electronic circuitry. But what happens when the photocopy machine refuses to work properly? No system is completely immune to Murphy's law.

Graphics must be supplied by the user after the test is printed by computer. We can think of no inexpensive way around this, although Lippey remarks on one system which supplies "connect the dots" diagrams to the student.[9] We have found this a minor inconvenience, however, when compared to the overall ease of creating examinations that are otherwise complete.

Our program is limited in its applications and in its complexity. It is probably more trouble than it is worth to establish large question files for small courses that are not conducted each term or each year. Nor is our program as ambitious as the "interactive" testing programs described by Alfred Bork.[10] These exams are administered directly by the computer and provide the student with feedback and retesting on questions that are missed. Such dialogue programs require more extensive design considerations and much larger hardware costs than the system we describe. (We require only one terminal, for instance.) Nevertheless, our program can be of great use to self-paced learning situations, greatly decreasing the amount of time required to generate the many different mastery quizzes involved in these courses.

V. SUMMARY AND RECOMMENDATIONS

So far, our experience with computer-assembled tests is overwhelmingly positive. A prime factor in this evaluation is the ease with which our programs can be used by the instructor. The program requires only rudimentary typing skills to produce error-free, neatly formated examinations. Were it not for these features, for instance, if extensive keypunching were required, computerized test-generation might be as laborious as any other method.

For those considering similar applications, we would urge the following guidelines to producing a program which is more than a cumbersome curiosity:

(1) The program should be user oriented. Ours operates in conversational mode from a terminal. It automatically prompts the user for information, requires no attention to formating (excess text is automatically carried over to the next line), and permits the editing of the item files if any changes are needed after a question is entered.

(2) The program should allow as much flexibility of format as possible. For physics-oriented programs in particular, provision for superscripts and subscripts, underlining, and spacing for graphics should be provided.

(3) The system should be relatively inexpensive. A truly interactive testing system requires a large number of terminals. Ours can be operated on modest equipment.

(4) The system should foster improved testing techniques. Ours inherently promotes a balanced choice of questions, as described above. In the future we intend to modify the system so that analysis of student response to individual

items can also be recorded.

Appendix A contains a sample of an exam produced by TORQUEMADA, including questions from three item files. A detailed manual of the TORQUEMADA system, including more extensive examples of its operation, is available from one of the authors (L.M.).

ACKNOWLEDGMENT

We thank Richard Wood and Greg Winegar of the Gettysburg College Computing Center for their aid and encouragement.

APPENDIX A: SAMPLE TEST ASSEMBLED BY COMPUTER

(1) Atmospheric pressure on Mars is what fraction of that on Earth?
- (a) 0.1%
- (b) 10%
- (c) 1%
- (d) 50%

(2) The overall reaction that produces energy in main sequence stars is
- (a) $2C + O_2 = 2CO_2 +$ energy
- (b) $3He = C +$ energy
- (c) $CH_3 + HOH = CH_2 + 2H +$ energy
- (d) $H + H + H + H = He +$ energy
- (e) $C + N + O + H_4 = CNO + 2He +$ energy

(3) The current age of the universe is thought to be
- (a) about 1.8×10^{10} years
- (b) 500 million years
- (c) 10^8 years
- (d) 40 004 years
- (e) 5 billion years

(4) Which of the following did Galileo *not* discover with his new telescope?
- (a) Sunspots
- (b) The moons of Uranus
- (c) Mountains on the moon
- (d) Multitudes of new stars in the milky way
- (e) Phases of Venus

(5) What is the sun's highest altitude for observers at latitude $-20°$?
- (a) $+90°$
- (b) $+63\ 1/2°$
- (c) $+73\ 1/2°$
- (d) $+50°$

****Essay Questions Follow****

(6) It has been said that the Copernican revolution was not just the work of a single man. Name four men who contributed to the downfall of the Aristotelian system during the 16th and 17th centuries. For each, briefly describe his work and point out clearly how this work weakened a specific assertion of the Aristotelians.

PLEASE SIGN YOUR NAME_____.

APPENDIX B: COMMANDS[11] RECOGNIZED BY TORQUEMADA/FILES

Definitions:

A ⟨category number⟩ is a number between 1.0 and 99.98 having no more than two digits to the right of the decimal point.

A ⟨question number⟩ is a number between 1.0001 and 99.9899 having four digits to the right of the decimal point

DIRECTORY: This command provides a listing of all the categories present in the file, the category code numbers, and the number of questions stored in each category.

INSERT ⟨category number⟩: This command establishes categories in the file. The user is asked for a title.

DELETE ⟨category number⟩: This command will delete all questions stored under the specified number and remove the category classification from the file. (A "PURGE" command removes the questions, but leaves the category in the file.)

LIST ⟨category number⟩: This command provides a listing of the first lines of all questions within a given category, along with their respective question numbers.

ADD ⟨category number⟩: This command allows the user to add a question to the category specified. The computer asks for the question, asks for choices of answers if the file is a multiple-choice type file, and automatically assigns question numbers to the questions.

COPY ⟨question number⟩ to ⟨category number⟩: This command allows a single question to be assigned to several categories.

DELETE ⟨question number⟩: This command removes a question from the file.

LIST ⟨question number⟩: This command lists the entire text of the question and any choices of answers if the question is a multiple-choice type.

EDIT ⟨question number⟩: This command enables the user to use the question editing commands to alter a question already in the files. An "EXCHANGE" command switches the correct choice of a multiple choice question with one of its alternatives. A "REPLACE" command will replace a question or a choice of answer with alternate text entered through the keyboard.

[a] Present address: 617 Rockford Road, Silver Spring, MD 20902.

[1] Named for the director of the Spanish Inquisition at the end of the 15th century. Our testing methods, we trust, are more benign than his.

[2] A few recent articles on word-processing systems for microcomputers: D. L. Fitchorn, Kilobaud **20**, 22 (1978); P. Hughes, *ibid.* **26**, 110 (1979); A. Domuret, *ibid.* **30**, 32 (1979); and F. R. Rudeschel, Byte **4** (6), 22 (1979).

[3] L. E. Turner, Am. J. Phys. **46**, 914 (1978).

[4] *Taxonomy of Education Objectives: The Classification of Educational Goals. Handbook 1. Cognitive Domain,* edited by B. S. Bloom (McKay, New York, 1956).

[5] G. Lippey in *Computer Assisted Test Construction* (Educational Technology Publications, Englewood Cliffs, NJ, 1974), pp. 14–17.

[6] W. Dodrill (private communication).

[7] Reference 3, p. 914.

[8] Reference 3, pp. 916–917.

[9] G. Lippey, Ref. 4, p. 13.

[10] Alfred Bork, Am. J. Phys. **47**, 9 (1979).

[11] These are the most widely used commands in the TORQUEMADA system. A complete listing is tabulated in the user's manual available from the authors.