DOCUMENT RESUME

ED 207 808	SE 035 501

AUTHOR	Abelson, Harold; diSessa, Andy
TITLE	Student Science Training Program in Mathematics, Physics and Computer Science. Final Report to the National Science Foundation. Artificial Intelligence Memo No. 393.
INSTITUTION	Massachusetts Inst. of Tech., Cambridge. Artificial Intelligence Lab.
SPONS AGENCY	National Science Foundation, Washington, D.C.
REPORT NO	LOGO-29
PUB DATE	Sep 76
GRANT	NSF-EC-40708-X; SMI-76-05406
NOTE	54p.; Contains occasional light and broken type.
AVAILABLE FROM	Artificial Intelligence Lab., 545 Technology Square, Rm. 338, Cambridge, MA 02139 ($2.50).
EDRS PRICE	MF01/PC03 Plus Postage.
DESCRIPTORS	*Academically Gifted; Computers; *Computer Science Education; Curriculum Design; Gifted; Mathematical Applications; Physics; Pilot Projects; *Program Descriptions; Secondary Education; *Secondary School Mathematics; *Secondary School Science; Student Projects
IDENTIFIERS	*Student Science Training Program

ABSTRACT

During the summer of 1976, the MIT Artificial Intelligence Laboratory sponsored a Student Science Training Program in Mathematics, Physics, and Computer Science for high ability secondary school students. This report describes, in some detail, the style of the program, the curriculum and the projects the students under-took. It is hoped that this document can serve not only as a report to the National Science Foundation, but also as an elaboration of the program ideas about what would constitute a model educational environment for high ability secondary school students. (Author)

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. MEMO 393
LOGO MEMO 29

September 1976

STUDENT SCIENCE TRAINING PROGRAM IN MATHEMATICS, PHYSICS AND COMPUTER SCIENCE

FINAL REPORT TO THE NATIONAL SCIENCE FOUNDATION

Harold Abelson, Project Director
Andy diSessa, Associate Director

ABSTRACT:

During the summer of 1976, the Massachusetts Institute of Technology Artificial Intelligence Laboratory sponsored a Student Science Training Program in Mathematics, Physics and Computer Science for high ability secondary school students.

This report describes, in some detail, the style of the program, the curriculum and the projects the students undertook. We feel that all three aspects of our program are unique, even in the context of innovative SST Programs. We hope that this document can serve not only as a report to the National Science Foundation but also as an elaboration of our ideas about what would constitute a model educational environment for high ability secondary school students.

The work reported in this paper was supported in part by the National Science Foundation under grant number EC40708X and SMI76-05406 and conducted at the Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Mass. The views and conclusions contained in this paper are those of the author and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the National Science Foundation or the United States Government.

Contents

I. Introduction

During the summer of 1976, the Massachusetts Institute of Technology Artificial Intelligence Laboratory sponsored a Student Science Training Program in Mathematics, Physics and Computer Science for high ability secondary school students. The program ran for six weeks, from July 12 through August 20 Fourteen students participated, four of whom were housed in MIT dormitories The rest commuted from their homes in the greater Boston area.

This SST Project was run by the Logo Group, a joint research endeavor of the Artificial Intelligence Laboratory and the MIT Division for Study and Research in Education. The research of the Logo Group centers around the implications of computer technology for education, and it is based on the premise that the increasing availability and decreasing costs of this technology will form the foundation for major structural changes in the nature of public education, both in the way that material is taught and in the content that is considered part of a normal education program. In connection with this research the group has developed computer systems, computer languages and computer-controlled devices tailored for educational use, as well as some substantial reformulations of content material, especially in mathematics and physics. Most of this work has been focused at the elementary school level and is supported by NSF Grant EC-40708, entitled "The Uses of Technology to Enhance Education." The SST Program was an opportunity to try out these ideas with a different age group and in an informal and almost ideal setting: an intense exposure with highly motivated superior students

This was the first time that the Artificial Intelligence Laboratory or, to our knowledge, MIT has sponsored an SST program, and we believe that it was extremely successful. This report describes, in some detail, the style of the program, the curriculum and the projects the students undertook. We feel that all three aspects of our program are unique, even in the context of innovative SST Programs. We hope that this document can serve not only as a report to the National Science Foundation but also as an elaboration of our ideas about what would constitute a model educational environment for high ability secondary school students.

II. Style and Environment

In working with the students this summer, covering a curriculum was only part of our goal. We also wanted to communicate to them a sense of how work goes on in a laboratory such as ours, and we viewed this as a process of cultural acclimation rather than as something to be explicitly taught. So before discussing specifics of curriculum and student projects, we mention here some details of the environment and the attitudes that the staff brought to the program.

1. Computation was not treated as a scarce resource.

We were fortunate in having exclusive use of an entire time-sharing system for the duration of the program. This included about eight terminals, each fully equipped for computer graphics, a large projection color television which can be used for graphics, a music synthesizer and a few peripheral minicomputers. Students' access to these facilities was unlimited, whether for completing a homework assignment, working on a project or simply fooling around.

2. We tried to minimize artificial distinctions between student and teacher.

We wanted our program to function, not as a classroom situation, but rather as an intellectual community in which each participant would act as both teacher and learner. So we took care that the program staff would be as accessible and relations between staff and students as informal as possible. Most staff members were present in the laboratory for a large portion of each day, offering help with homework and programming projects or joining in on bull sessions. Even "lectures" were interactive and frequently contained discussions among students and "lecturer" alike All participants addressed each other on a first name basis. In addition, students were given staff members' home phone numbers for after-hours assistance with homework.

3. We stressed cooperation, not competition.

We do not suggest that examinations and insistence that students work individually have no place in an educational program. Indeed, as we shall discuss below, performance on an extremely rigorous examination was one of the criteria we used to select program participants. But we saw no reason to emphasize competition once students had been accepted to the program. On the contrary, we felt a responsibility to acquaint students with the distinction between academic game playing and the way that scientific work is done.

For example, not only did we allow cooperation on homework assignments -- we actively encouraged it, and when students had questions about the material, we encouraged them to work together to find the answer, rather than to automatically ask one of us Incidently, a computer environment is very useful in this respect, for although students may at first feel that they are losing face by going to a colleague for help with theoretical matters, they are quite willing to cooperate in the details of getting a computer program to run. After a while this attitude carries over to the theoretical problems as well.

III. The Curriculum: Computation as an Intellectual Framework for Mathematics and Physics

Students typically worked in our laboratory between 10 AM and 5 PM Of these hours, 1:00 to 2:00 was a break for lunch and 11:30 to 1:00 was a group meeting The rest of the day was reserved for working on projects and access to the computer The group meetings thus comprised only a fraction of each day, but they were focal to the entire program, for they illustrated how computation can provide a framework for understanding advanced topics in mathematics and physics.

1. The group meetings covered differential geometry, physics and topology.

We began with a look at elementary geometry, but from a novel computational point of view. This regards geometric figures as the paths traced by a simulated robot following simple computer programs and forms the basis for a comparison between procedural formulations and axiomatic formulations It also highlights the distinction between the *intrinsic* and *extrinsic* properties of geometric figures, and between formulations that are based upon *local information* and those based upon *global information* (For example, a circle can be described both intrinsically and locally as a curve of constant curvature, while using the familiar equation of analytic geometry relies on an extrinsic, globally defined, coordinate system.) Besides serving as an introduction to geometry and programming, a study of the figures drawn by simple programs leads to interesting questions in number theory. This section of the curriculum ended with theorems about the total curvature of closed paths and a classification theorem describing the figures drawn by an important and general class of programs.

Next we turned to physics and a discussion of force The basic principle here is that applying a force to an object does not directly change the object's position Rather it changes the object's velocity. In order to capitalize on this idea we introduced the concepts of velocity space, and of describing physical interactions in *velocity space*. Many puzzling behaviors of physical systems, such as the way a spinning gyroscope reacts when we try to move it, become transparent when described in terms of velocity space. As an application, we gave a thorough treatment of orbital mechanics, founded upon the observation that, for a particle moving in an inverse-square force field, the velocity space path is a curve of constant curvature, hence a circle. Starting from here we proved Kepler's Laws and gave a rather complete discussion of the first-order perturbation theory of planetary orbits. For example, we derived the precession of Mercury's orbit due to the oblateness of the sun, and discussed the consequent challenge to Einstein's theory of gravitation. Our section on physics concluded with an alternative image of force -- force as momentum transfer -- and a comparison of this with the image of force as a velocity changer.

The third and final portion of the curriculum returned to the study of local and intrinsic geometric properties. We took another look at curvature, defined the notion of a topological deformation, and showed that total curvature is a topological invariant for closed curves. Then we capitalized on the procedural formulation of our geometry: since our geometry is local and intrinsic, it should work just as well on anything which is locally like a plane, in short, on any two-dimensional surface. In this context, the total curvature of a closed path on a two-dimensional surface is a measurement of whether the surface itself is flat or curved. Following these lines we defined *geodesic*, *Gaussian curvature* and total Gaussian curvature and proved that the latter is a *topological invariant* for closed surfaces. This led to a discussion of curvature and geodesics in three and four dimensions and their role in the General Theory of Relativity. Finally, we took another look at curvature in terms of vector fields, proved the Hopf Index Theorem and derived some applications including the Gauss-Bonnet Theorem and the Brouwer Fixed-Point Theorem.

2. Assignments emphasized exploration and active understanding.

It was an important part of the style of the program that almost all of the topics covered in group meetings were supplemented by (and sometimes even motivated by and derived from) exercises, explorations and mini-research projects which the students worked on outside of the group meetings. For example, the elementary geometry section involved students in inventing, classifying and analysing their own figure-programs. The orbital dynamics material involved such activities as guiding a spaceship to dock with a designated target in a computer animated orbital simulation. The differential geometry section incorporated computer sessions in which students explored the special geometry of the surface of a cube and also determined global properties of surfaces which of which they could observe only local properties.

3. We also presented special lectures on subjects suggested by the students.

During the first week of the program, we asked students to suggest topics for special lectures. These lectures were held about once a week in place of the usual group meeting. Lectures included:

"Algebraic Number Fields and the Angle Trisection Problem," by Hal Abelson

"Science and Science Fiction," by Bonnie Dalzell

"Artificial Intelligence," by Marvin Minsky

"A Computational View of the Skill of Juggling," by Howard Austin

"Recursive Programming and Pattern Matching," by Danny Hillis

"Large Program Design and Organization," by Neil Rowe

"Infinity," by Andy diSessa

"Structured Programming and Debugging," by Henry Lieberman

IV. The Projects

In addition to the curriculum discussed above we felt strongly that our program should have an active component, in which students are given a chance to *do* science rather than to merely *learn about* science. Every student in the program, therefore, participated in one of three groups which worked on extensive computer-based projects. Each group met every day and was directed by a staff member of the Logo Project

1. The physics group designed a system for modeling stress in complex structures.

The physics group was led by Margaret Minsky a senior in computer science and a staff member at Logo. Their project was developing a "stress machine," a computer program for simulating mechanical stress in structures such as bridges and buildings. They began with a study of stress and strain, what these are and how they are computed. Then they had to decide upon a way to model physical objects in the computer. The group chose to imagine objects as being built out of elastic rods joined together at vertices. For each rod one can specify its resting length and its elasticity. Rods join together at vertices, at which one can specify external forces on the structure (such as the force of gravity) or various constraints determining how the vertex is allowed to move when the structure deforms.

Next came the choice of a computational scheme. This proceeded as follows: start with a vertex, compute the resultant forces on it, and allow it to move. This puts stress on the neighboring vertices via the connecting rods, and the displacement of these vertices is likewise computed. This, in turn, stresses further vertices, and so on, until the structure settles to equilibrium. Watching the process on a computer display gives a feeling for how stress "propagates" through a structure. Indeed, the process was chosen to be a reasonable model of the actual physical process of "computing" stress and strain. Of course, there are many questions to be resolved in this rough outline. In what order should the vertices be updated? Are the computations to be done in serial or in parallel? How much difference do these choices make? In exploring these questions, students tried variations on their basic program and compared results

The method the students developed is rather different from the way in which such simulations are done in most "professional" programs for stress analysis, which represent such structures as systems of linear equations Although the students' method is less efficient, it is conceptually clearer and more general since it can deal with cases where the characteristics of the structure change under stress (for example, a rod breaking when stressed too much). In addition, the dynamic approach of "stress propagation" can often give insight into the nature of structures and ideas for designing better structures.

This kind of insight is illustrated in a sequence of experiments performed by Mike Sannella, one of the students in the group He first built a simple truss bridge, shown in Figure 1. Then he applied the stress machine, and watched the bridge deform. (See Figure 2.) This led him to the idea that the bridge was somehow trying to get into the shape of a suspension bridge, then to the idea that perhaps the suspension bridge shape is a more natural one for objects under stress. So he next built a suspension bridge (Figure 3). Watching it deform (Figure 4) suggested that he next needed to reinforced the ends. In this way, he developed a whole series of bridges, each suggested by observing how the previous one deforms or fails under stress.

2. The biofeedback group explored ways to control a computer using muscle tension.

The second group, under the direction of staff member Paul Goldenberg, contributed to a long-term research interest of our laboratory -- enabling physically handicapped people to use the computer as a communication device. Communication here

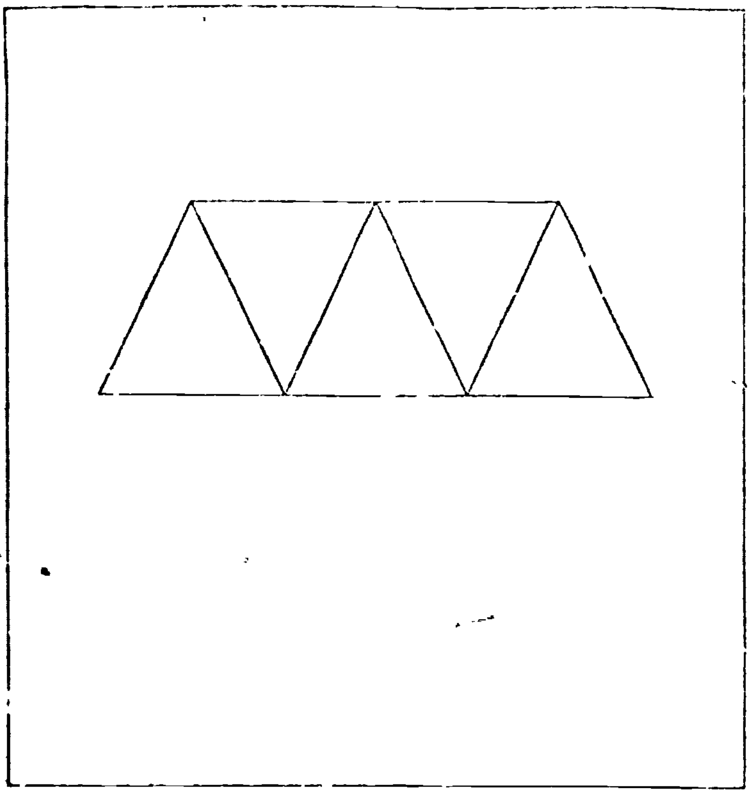


Figure 1: A simple truss bridge

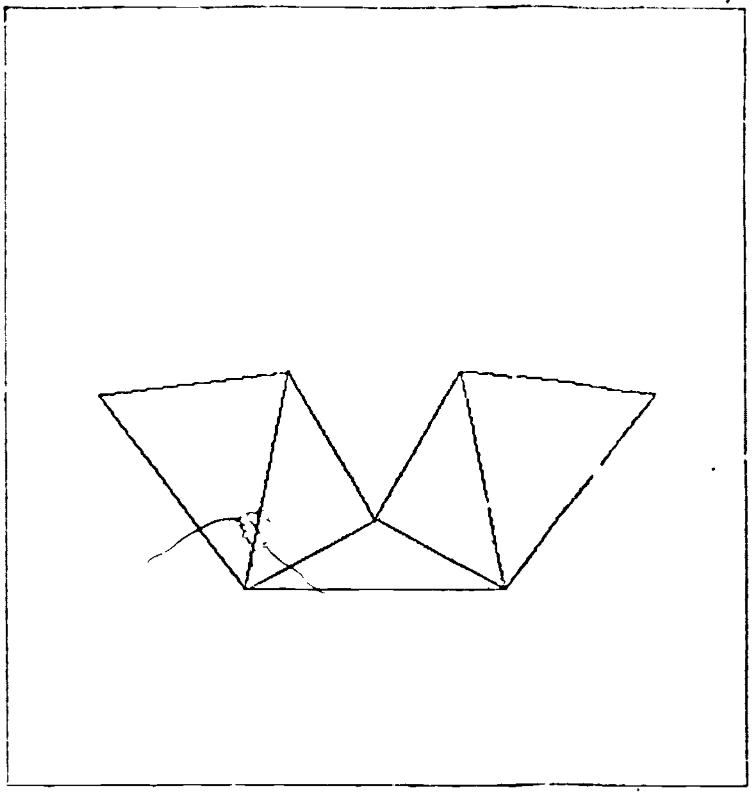


Figure 2: The bridge deforms under stress

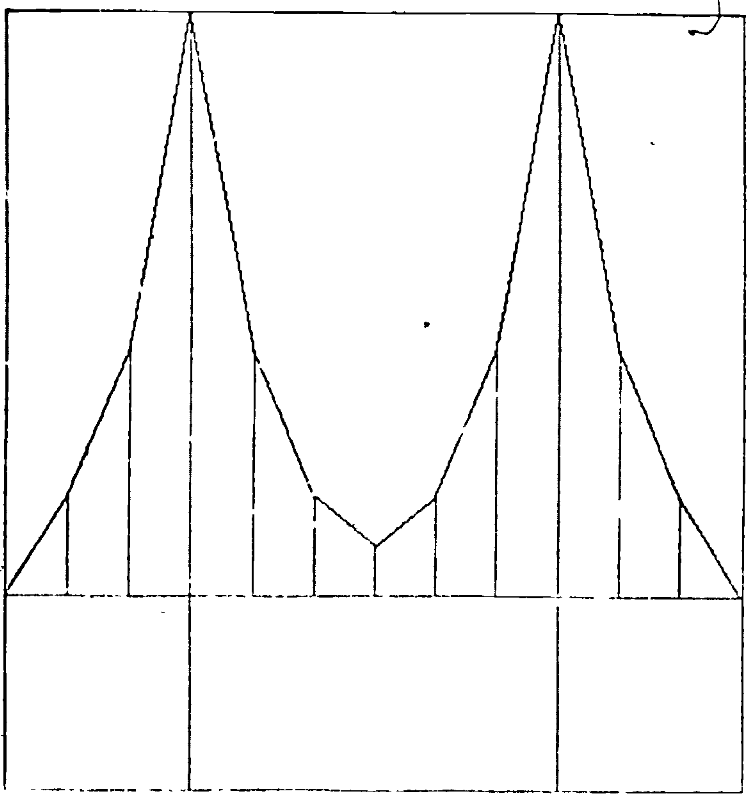


Figure 3: A suspension bridge

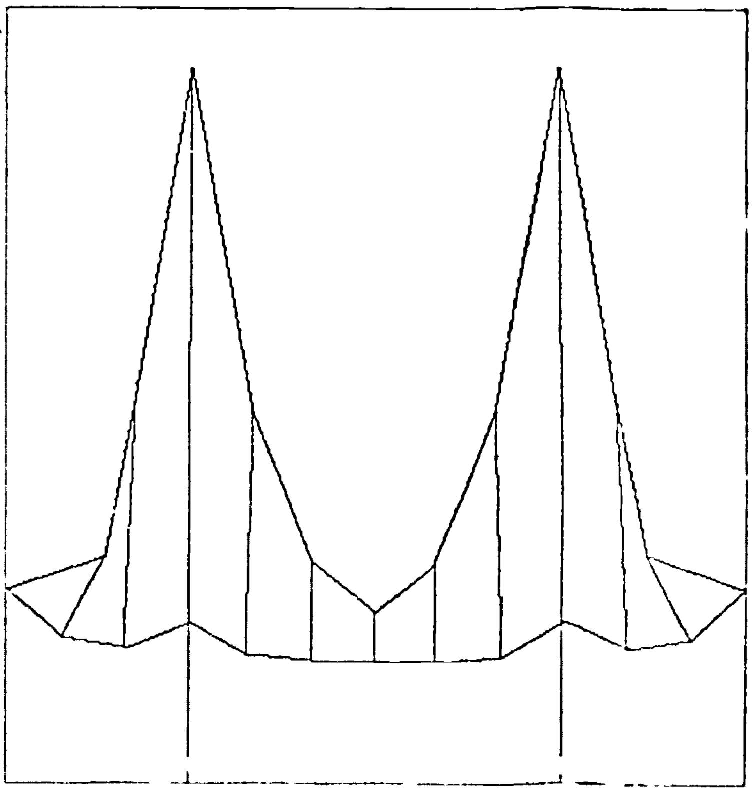


Figure 4: The bridge deforms

refers not only to writing, but also drawing, playing music, playing games and so on. The group's work this summer centered on the following question -- if we interface an electromyograph (EMG) as a computer input device, can we use this to control the computer through selectively varying muscle tension? This research would be of benefit, for example, to people who are victims of severe cerebral palsy and are consequently unable to write or type Although a number of workers in this area have developed "myoelectric switches" for the handicapped, no one, to our knowledge, has made use of muscle tension as other than an "off-on" switch. Theoretically, the computer should be able to recognize different levels, or perhaps even distinguish patterns, of muscle tension.

The group began the summer by getting a background in physiology, and learning about cerebral palsy and other neuromuscular disorders. They studied how normal voluntary muscle control is achieved, and used the computer to develop simulations of muscles being controlled via feedback mechanisms. In order to obtain a better understanding the problems of the handicapped, the group spent two days at the Crotched Mountain Rehabilitation Center in New Hampshire.

Ellen Hildreth and Bruce Edwards, MIT seniors and members of the Logo Project, assisted Goldenberg in running the group. They interfaced the EMG to our computer system via a microcomputer, applying signal processing methods to transform the raw EMG signals into a form which the high school students could use in developing programs. They also gave talks on signal processing techniques to give the group a basic understanding of how the interface worked

The group used this interface to develop a number of demonstration programs which illustrated computer control through muscle tension One of these was a drawing program which allowed a person to draw using muscle tension by controlling a moving cursor, selecting commands such as move forward, move back, move left, move right One of the students, David Hoch, set up a computer tic-tac-toe game in which a person plays against the computer, using muscle tension to select the desired move Another student, David Glazer, experimented with schemes for dynamically controlling the range, scale and number of levels of muscle tension which can be recognized by the computer

3. The color group experimented with computer graphics and visual perception.

The third group, under the direction of research staff member Henry Lieberman, investigated color graphics using a large screen projection color television adapted as a computer output device. Much of the group's work centered on creating computer art which illustrated aspects of visual perception For example, by drawing in red and green one can produce sterograms, which give a three-dimensional effect when viewed with red-green glasses Other explorations dealt with various models for "color spaces," or geometric representations of the collection of colors according to the relative amount of red, blue, and green, or according to hue, saturation and intensity One student, Charles Lowenstein, used such a representation to interpolate a sequence of colors between any two given colors, and he incorporated this scheme into various computer designs

Besides working on these projects the group learned about theories of visual perception, current research in computer vision and details of computer graphics systems They also visited other facilities at MIT involved in computer graphics research

V. Participant Selection

In addition to our listing in the NSF national brochure, we publicized the program in letters which were sent in February to most of the high schools in the greater Boston area, about 50 letters in all. Applicants were required to supply individual and school forms, two letters of recommendation and also complete an examination included with the application materials. (These materials are included as Appendix A of this report.)

The exam served a number of purposes. Besides its obvious use as a selection criterion, we felt that it could serve as an effective self-screening device whereby applicants who were half-hearted about wanting to participate in the program would not devote the considerable effort required to complete the exam And we suspect this actually was the case, for of the 133 application packages we mailed out, only 47 were returned with completed exams.

A third reason for including the exam is that we wanted to provide some bright high school students with something which is missing from most educational programs -- the opportunity to work on hard problems Students are rarely given problems that they are expected to work on for weeks or months, even though this is the kind of problem usually encountered in scientific work. Both the director and associate director of our program participated in SST projects when in high school and were required to complete such qualifying exams. Both agree that this, their own first encounter with hard scientific problems, was for them a significant educational experience In keeping with this attitude we supplied detailed exam solutions to all applicants (See Appendix B.)

Of the 47 complete applications we accepted 15 and designated 7 alternates All 15 students we selected wrote indicating that they would participate, and so none of the alternates were selected

In selecting participants highest weight was given to performance on the exam, not based so much on whether the answers were correct, as on the degree of insight and originality of the proposed solutions Using the exam was probably the only meaningful way we could have made the selection since, almost without exception, applicants had highly superior grades and recommendations, and we do not consider scores on standardized tests to be a reliable indicator of potential intellectual achievement.

VI. Other Activities

Program participants were provided with temporary identification cards indicating their association with the MIT Artificial Intelligence Laboratory. They were also given access to the MIT library system and, or a nominal fee, to the campus athletic facilities

On August 19th an open house was held at the laboratory for students' friends and relatives Each project group gave a brief presentation illustrating their work, and guests were given an opportunity to use the laboratory equipment

VII. Problems and Comments

The major problem we encountered this summer was that the project groups were late in getting underway. Each of the projects demanded a substantial amount of hardware and software development by our staff, and this was not completed by the beginning of the summer. As a result, the initial meetings of the project groups could deal with their topics only from a theoretical point of view, and most of the actual project work was done towards the end of the summer. We would not, however, expect this to be a problem should we conduct such a program in the future, for the initial start-up efforts for these projects is now complete.

Another problem which arose was that one of the students decided to drop out of the program. Although he did not attend the program after the first day, it was more than a week before he notified us of his decision. In future programs we would anticipate the possibility of sudden drop-outs and be prepared to give last-minute acceptance to one of our alternates.

VIII. Evaluation

The main goal of our program was to establish an environment which encourages active participation in learning. We wanted to impart a commitment to understanding and a sense of what scientific research is all about, goals which involve, but go beyond, "learning the material." Beyond our own commitment the most vital link in our attempts to establish such an environment was the constant availability of computer facilities. It seems reasonable therefore to measure the success of our program in part by the interaction between students and the computer.

1. The computer as research tool -- We mentioned above the role of the computer in excercises associated with the curriculum These exercises were many times designed to involve students in studies approaching the depth and complexity of the three group projects. We were extrememly pleased to see how often and how effectively students applied the computer in these studies. Several students regularly turned in extensive and original mini-research reports as part of their assignment write-ups

2. The computer as understanding exerciser and evaluator -- Although we regularly commented on homework papers, the first-hand and immediate experience of seeing a solution work or discovering and verifying a new hypothesis seemed to contribute more to students' confidence in their understanding In general those parts of our curriculum most closely tied to this kind of experience seemed the most successful Even the special lectures benefitted. For example, a couple of students made practical use of the lecture on recursive programming and pattern matching to write programs to perform symbolic differentiation, a significant achievement for students at this level.

3. The computer as a design medium for self-expression -- Even those students who did not venture far intellectually from the presented curriculum used the computer in personal and individual ways. From computer-generated pictures and designs to music we saw many examples of imaginative and skillful use of the machines.

4. The computer as toy -- To indicate the degree to which the computer functioned as a natural part of the environment, we could cite numerous examples of games and pranks which were invented, from computer ping-pong to dynamic simulations of airplanes and sailboats, from tic-tac-toe to complex board games, from a program to generate junk mail to interactive language "understanding" systems. Students devoted a great deal of effort to such projects, many of which made use of such knowledge acquired during the program as particle dynamics and linguistic parsing As an expression of personal, human involvement we were pleased to see this kind of work go on, especially when a playful bull session on games can turn into a discussion of the representations of knowledge, heuristics and other strategies which could make the computer into a formidable opponent.

We saw other evidence of the success of our program in involving students' full energies. Some students engaged in serious discussions of career opportunities related to following up their experience in our program. Several had gained such a positive impression of MIT that we felt obliged to talk with them about the possible negative aspects of attending MIT or similar institutions

To be sure, we did not accomplish all we had planned to, nor did everything go smoothly. Computers crashed, some lectures had rough edges, some material was ignored or not understood. But, in all, we are highly encouraged with our first attempt at an SST Program. It's appropriate to close this section with a couple of spontaneous remarks made by students at the end of the program.

"After two weeks of this program I couldn't imagine going back to regular school."

"In six weeks here I learned more than in a year in high school"

IX. Academic Staff

Hal Abelson, who directed the project, is Lecturer in the MIT Mathematics Department and in the Division for Study and Research in Education. He has been a member of the MIT faculty since 1973 and a staff member at the Artificial Intelligence Laboratory since 1970. He holds and A.B. degree from Princeton University (1969) and a Ph D in Mathematics from MIT (1973) He is interested in how science and mathematics can be made more intuitively accessible through the application of computer technology.

Andy diSessa, who acted as associate director of the project, is Special Lecturer with the MIT Division for Study and Research in Education He holds an A.B. degree from Princeton University (1969) and a Ph.D. in Theoretical Physics from MIT (1975) He has been a member of the Logo group since 1972. His interests are primarily in developing more active and intuitively attractive structures for understanding physics and mathematics. He is also interested generally in the application of computer and other modern technology to science and mathematics education

Bruce Edwards is a senior at MIT majoring in Electrical Engineering and Computer Science He has been a member of the Logo group since 1974 During the summer he developed the microcomputer software for interfacing the EMG to our timesharing system.

Paul Goldenberg, who directed the biofeedback group, has been a member of the Artificial Intelligence laboratory research staff since 1974 He has previously taught in elementary and junior high school, at Simmons College and the University of Chicago Laboratory Schools. He has also published mathematics textbooks and curriculum material

Ellen Hildreth is a senior at MIT majoring in Mathematics She has been a member of the Logo group since 1974. Her interests are in applying computer technology to elementary education, and she is exploring ways to integrate computer programming into a wider range of activities for young children This summer she aided in developing signal processing techniques for use in the biofeedback project

Henry Lieberman, who directed the color group, has been working at the Logo project since 1973, initially as an undergraduate and since 1975 as member of the Artificial Intelligence Laboratory's research staff. He holds a B.A. in Mathematics from MIT (1975). He has previously worked on an implementation of the Logo computer language written in Lisp and, more recently, has developed the software for the color graphics system used at our laboratory.

Margaret Minsky, who directed the physics group, is a senior at MIT majoring in Mathematics. She has been a member of the Logo group since 1974 and has taught computer programming to both elementary school students and high school students

Neil Rowe is a graduate student in Computer Science at MIT. He holds an S B in Electrical Engineering and Computer Science from MIT (1975). He has worked with the Logo group since 1972 and has taught computer science in a variety of informal courses for high school students. This summer he worked with the directors in organizing and running the project.

X. Acknowledgements

In addition to the academic staff, thanks are also due to **Eva Kampits**, who served as administrative officer for the project, and to **Greg Gargarian** and **Donna Barry**, who provided secretarial support. It is mainly through Gargarian's efforts that we were able to arrange MIT housing for our commuting students and provide participants with library cards.

XI. Student Participants

Maureen Brown
48 Edgewood Ave
Chicopee, MA 01003
Cathedral High School

Denise Dzwonczyk
339 Cook Lane
Marlboro, MA 01752
Marlboro High School

David Glazer
20 Monadnock Dr.
Shrewsbury, MA 01545
Shrewsbury High School

Lorraine Gozzo
45 Court Rd
Braintree, MA 02184
Braintree High School

Frank Hale
5 Tippy Cart Rd.
Canton, MA 02021
Boston College High School

David Hoch
72 Blake Rd.
Brookline, MA 02146
Maimonides High School

Thomas Kimble
1959 Teehan Lane
Baldwin, N.Y. 11510
Baldwin High School

Chun Chee Lau
1010 Rutland Rd
Brooklyn, N.Y. 11212
George W Wingate High School

Charles Lowenstein
170 Cliff Rd.
Wellesley, MA 02181
Milton Academy

Stephen Phillips
51 Geraldine Dr.
Norwood, MA 02062
Norwood High School

Michael Sannella
493 Worcester St.
Wellesley, MA 02181
Rivers Country Day School

Nancy Saraf
63 Highland St.
Hyde Park, MA 02136
Boston Latin Academy

Mark Sobolewski
143 Hillcrest Rd.
Concord, MA 01742
Concord Carlisle High School

Daniel Yates
3150 Palm Aire Dr.
N. Pompano Beach, Florida
Pine Crest High School

Appendix A

Application package, including:

- program information
- student application form
- school nomination form
- recommendation form
- exam

APPENDIX A

[illegible]
[illegible] Technology [illegible]
Cambridge, M[illegible]
Telephone [illegible]

February 20, 1976

THANK YOU FOR YOUR INTEREST IN THE MIT SUMMER PROGRAM

First we must appologize to those of you outside of the Boston area since unfortunately the NSF announcement failed to mention that ours is a commuting only program. We have no funds to support nor can we accept responsibility in any way for transportation and board.

Enclosed is an application form, a form to be filed by your school and an entrance examination. You must see to it that these forms and your solutions to the exam are returned complete by April 1. We will notify applicants of their status, accepted, rejected or alternate by April 15 and in return we expect the final decision from those accepted before May 1.

Please be aware that the examination is a difficult one. Don't expect to be able to do it in an afternoon. The problems are meant to test your abilities at longer time scale and less clearcut work than is normally expected in highschool. We will be looking for your recognition of partial results and at your methods of attack as much as "the right answer".

WHAT YOU MUST DO:

1) Fill in the application form
2) Give the school form to the appropriate person (principal, headmaster ...)
3) Ask two of your teachers to write letters of recommendation for you. Other people who know you are acceptable but only if your connection to them involves your study or work in "scientific" areas.
4) Complete the exam.

All of the above must be recieved by us before April 1. 1) and 4) should be sent to us in one package. The school form in all cases should be sent directly by your school. Letters of recommendation may be sent by the school with its form or by the teachers under separate cover.

Good Luck..

The M.I.T. Summer Student
Science Program.

APPENDIX A

The MIT Artificial Intelligence Laboratory
announces a
Science Training Institute
for High Ability Secondary School Students
July 12 through August 21, 1976

This is an opportunity for about 15 students to work at the Massachusetts Institute of Technology Artificial Intelligence Laboratory, using the Laboratory's computer facilities as tools in exploring topics in mathematics and physics. Topics will be drawn from two main areas. The first centers around computer simulation and analysis of simple psycho-biological behavior: starting with tropisms, use of sensory data, feedback, maze-solving and extending to pattern recognition and signal processing. The second concerns an intuitive approach to mechanics, in particular to orbital mechanics, based on the geometry of velocity space.

Activities: The training institute will be held on weekdays for six weeks, from July 12 to August 21, 1976. Students will be encouraged to work independently and given access to a computational laboratory whose facilities include a time-sharing system and graphic display terminals, a computer-controlled color TV, small robot devices, a speech synthesizer, and some mini-computers for exploring "real time" applications. The six week period will be divided into roughly three phases as follows:

> Phase I (approximately one week): Students will gain initial facility with the computer language and equipment through introductory projects. Seminars will focus on programming issues and also provide introduction to the two areas mentioned above.
>
> Phase II (approximately two weeks): Seminars will go more deeply into theoretical material in mathematics and physics. Students will undertake simple projects in both areas.
>
> Phase III (approximately three weeks): Each student will work on an advanced project, and staff will provide individual supervision. Seminars will cover short informal introductions to a number of related topics such as differential geometry, symmetry and invariants, computational geometry and artificial intelligence.

Staff: This program will be supervised by Drs. Harold Abelson and Andrea diSessa, both of the MIT Artificial Intelligence Laboratory.

Fees, Transportation, Meals: This institute is designed for students commuting to MIT from the greater Boston area. Participants are expected to furnish their own transportation to and from MIT. (The AI Lab is located at 545 Technology Square on Main Street in

Cambridge, within easy walking distance of the Kendall Square station on the MBTA Red Line.) Lunches will not be provided by the program, but may be purchased at various cafeterias and shops in the MIT area. There are no additional expenses, tuition or fees.

Selection of Participants: Applicants should have a strong background in mathematics, but previous computer experience is not required. Preference will be given to students completing the 11th grade and who have taken elementary physics. Applicants must be academically in the upper 25% of their high school class, but selection will be based more on ability to work independently, as evidenced in letters of recommendation, than on grades per se.

Applications: Application materials including forms to be completed by the applicant and her or his school, an application examination, and forms for two letters of recommendation may be obtained by writing

Dr. Harold Abelson
Artificial Intelligence Laboratory
545 Technology Square
MIT
Cambridge, Mass. 02139

Completed application materi. must be returned by April 1, 1976. Notification of selection decisions will be made on or about April 15, and applicants selected to the program will have until May 1 to accept.

This program is sponsored by the MIT Artificial Intelligence Laboratory in cooperation with the National Science Foundation.

In the operation of the project and in selecting individuals for participation in, and for administration of, the project, MIT will not discriminate against any person on the grounds of race, creed, color, sex or national origin.

NSF Form 418, Nov. 1968

Form approved
OMB 99-R0225

APPENDIX A

Application for Participation in the Training Program
for Secondary School Students Supported by the
National Science Foundation

STUDENT APPLICATION FORM

Host Institution: Massachusetts Institute of Technology

Project Director: Dr. Harold Abelson
Artificial Intelligence Laboratory MIT
545 Technology Square
Cambridge, Mass. 02139

Project Dates: July 12 - August 21, 1976

A complete application includes: (1) this form with every item filled in, (2) the School Nomination Form, (3) two letters of recommendation, and (4) the entrance examination. All forms are to be sent to the project director (not to the National Science Foundation) by April 1, 1976.

Supply a complete answer to each item, writing NONE where appropriate.

1. ________________________________ [ ] Male [ ] Female
(Student's last name) First Middle

2. Social Security Number: ______________

3. Date of Birth ______________

4. Address: ____
Street & Number City State ZIP Telephone

5. Father's Name: ______________ Address: ______________
Mother's Name: ______________ Address: ______________

6. High School: ____
Name Street City State ZIP

7. What is the approximate population of your town? ______________
Is it a metropolitan suburb? ______________

8. Do you plan to attend college? ______________
For what degree(s)? ______________

9. Major academic area of interest: ______________

10. Other hobbies and strong interests: ______________

11. Describe briefly your participation in SCIENCE activities, both in and out of school (include clubs, fairs, rallies, office held, etc.) ______________

(over)

12. Describe briefly your participation in NONSCIENCE activities, both in and out of school (include athletics, operetta, band, clubs, debates, theatre, etc.):

____

____

____

... on (both in and out of school): ____________

____

14. Have you ever participated in a student science program supported by NSF? __________
If your answer is yes, please supply the following information:

Name of Host institution: ____________________

Location: ________________ Dates: ______________________

Director: ________________

Subject field(s) you studied: ____________________

Description of Program: ________________________

15. PARENT'S CONSENT: As the parent/guardian, I certify that my son/duaghter/ward has my permission to participate in this project for secondary school students. It is my understanding that he/she will be subject to the regulations of MIT and the project. I understand that should a health emergency arise, I will be notified, but that if I cannot be reached by telephone, such medical treatment as deemed necessary by competent medical personnel is authorized. I understand that transportation and escort of program participants to and from scheduled activities is not the responsibility of MIT, and I release MIT and its employees from any liabilities arising from such transportation.

____
Signature of Parent or Guardian

16. ____________________ ______________________________
Date of Application Signature of Applicant

NSF Form 419, Nov. 1968

Form approved
OMB 99-R0225

APPENDIX A

Application for Participation in the Training Program for Secondary School Students Supported by the National Science Foundation

SCHOOL NOMINATION FORM

Massachusetts Institute of Technology

Dr. Harold Abelson
Artificial Intelligence Laboratory MIT
545 Technology Square
Cambridge, Mass. 02139

July 12 - August 21, 1976

A complete application includes: (1) this form with every item filled in by the school principal or his designee, (2) the completed Student Application Form, and (3) any local forms (including letters of recommendation) as required by the host institution. All forms are to be sent to the project director (not to the National Science Foundation) by April 1 to guarantee consideration of this application.

Supply a complete answer to each item, writing NONE where appropriate.

1. ______________________________________ [ ] Male [ ] Female
Last name of Nominee First Middle

2. ______________________________________
Address of Nominee City State ZIP Telephone

3. ______________________________________
Name of High School Address City State

Which of these best describes this school?
[ ] Public [ ] Private [ ] Parochial

Which grades are included? [ ] 7-9 [ ] 7-12 [ ] 9-12
[ ] 10-12 [ ] Other_______________.

How many pupils are in your school?______
How many in the 12th grade?_______

4. The nominee named above is presently enrolled in the ________ grade, is expected to be graduated in _________, and stands (class rank) _________in a class of ________

5. List scores obtained by this student on nationally standardized tests (I.Q., PSAT, SAT, ITED, etc.). Please identify these tests accurately. If you enclose a transcript on which these data appear, you need not repeat it here.

Name of Test	Date	Score	Percentile

(over)

(no.5 continued)

Name of Test	Date	Score	Percentile

6. Please complete the following table. In Column 1 list the grades obtained by this student (you may enclose a transcript instead of completing Column I). In Column II indicate whether the course is offered in your school by checking those applicable.

	I	II		I	II
General Mathematics			General Science		
Elementary Algebra			Biology		
Intermediate Algebra			Chemistry		
Plane Geometry			Physics		
Solid Geometry			Earth Science		
Trigonometry			Geology		
College Algebra			Advanced General Science		
Analytic Geometry			Advanced Biology		
Mathematical Analysis			Advanced Chemistry		
Calculus			Advanced Physics		
Other Mathematics			Other Science		

7.

Name of Principal

Name and title of person completing this form

Date of Application

Signature

Letters of recommendation or additional comments may be attached to this form.

February 23, 1976

APPENDIX A

Application for participation in the Training Program for Secondary School Students supported by the National Science Foundation.

Recommendation Form

For the student

Name: ______________________

Address: ______________________

School: ______________________

For the respondent

Name: ______________________

Address: ______________________

Position (If teacher, then subjects taught): ______________________

____

Please comment on the applicant's abilities and talents particularly in the areas of math and science. We are especially interested in her/his ability to work independently. It would help us to know how long and how well you have known the applicant. Completed forms should be returned by April 1, 1976 to:

Dr. Harold Abelson
Artificial Intelligence Laboratory
545 Technology Square, M.I.T.
Cambridge, Mass. 02139

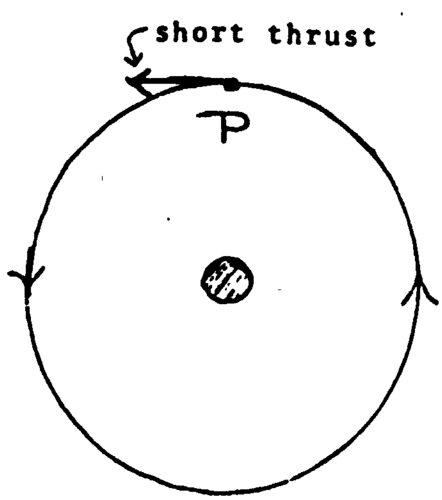
MIT Summer Program--Application Exam

Problem 1 Which is larger, *sin(cos x)* or *cos(sin x)* ?

Problem 2 By a "power of 2" is meant a number of the form 2^n for some integer n.
(a) Find a power of 2 that begins 52....
(b) Given *any* initial sequence of digits can more digits be added on the right to produce a power of 2?

Problem 3 If you look carefully at a television set you can see horizontal lines. The picture on the screen is generated by sweeping out one line at a time. But in what sequence are the lines swept out: starting at the top and working down, or starting at the bottom and working up? Devise a simple experiment to find out (without using any equipment except perhaps pencil and paper). Can you make a rough estimate of how long it takes the beam to make a vertical scan?

Problem 4 A spaceship is in a circular orbit about the earth. When the ship reaches the point marked *P*, the pilot changes orbit by applying a short tangential thrust (to increase his velocity). Sketch the resulting orbit, indicating how it differs from the original one. Explain.



APPENDIX A

Problem 5 This problem is about writing algorithms for a simple kind of computing device called a "turtle." Think of the turtle as a robot that lives on a large rectangular grid. Some the squares on the grid are white and some are black. The turtle's job is to walk around on the grid from one square to the next and decide things about the figure formed by the black squares. The specific things the turtle can do are:

move.right--move one square to the right

move.left--move one square to the left

move.up--etc.

move.down

print a message

The turtle can also test the following conditions:

color--outputs *black* or *white* according to the color of the square currently occupied by the turtle

at.right.edge--is true if the square currently occupied by the turtle is at the right edge of the grid

at.left.edge--etc

at.top.edge

at.bottom.edge

In addition the turtle has four counters, called *C1*, *C2*, *C3* and *C4*. The things the turtle can do with a counter are clear it (set it to zero) increment or decrement it by one, and examine its contents.

Assume that the turtle always starts out at the upper left hand corner of the grid.

Here, for example, is a program which tests whether the number of black squares on the grid is odd or even:

```
to find.parity
      clear C1
loop: if color=black then update
      if not at.right.edge then
                 move.right
                 go loop
      if not at.bottom.edge then
                 move.to.next.row
                 go loop
      if C1=0 print "even"
      if C1=1 print "odd"
      stop
```

```
to update
     if C1=0 then increment C1
     if C1=1 then decrement C1
```

```
to move.to.next.row
loop: if not at.left.edge then
             move.left
             go loop
         move.down
```

Notice how the program uses subroutines (update and move.to.next.row) to make the program easier to understand.

Now here are the problems:

(a) Write a program which has the turtle test whether or not the figure formed by the black squares is a solid rectangular area. (Since the rectangle will be made up of little black squares, its sides must necessarily be parallel to the sides of the grid.) You may assume if you like that none of the squares along the edges of the grid are black.

(b) Suppose the turtle's counters are broken so that they can only count single digit numbers. So whenever a counter is at 9, incrementing it will set it to 0. Whenever a counter is at 0, decrementing it will set it to 9. Can the turtle still do the problem in part (a)? If yes, write a program to accomplish this. If no, show why it cannot be done.

Appendix B

The following exam solutions were sent to all program applicants. We would like to thank Neil Rowe for preparing the solutions.

APPENDIX B

The Artificial Intelligence Laboratory
545 Technology Square
Cambridge, Massachusetts 02139
Telephone 617 253-6215

May 3, 1976

Dear Applicant to the M.I.T. NSF SSTP,

Many of you asked us for a set of answers to our entrance exam, and as the exam should have been a learning experience as much as a "test" we are happy to oblige. We enjoyed reading and thinking about your solutions and hope you can say the same for ours.

Sincerely,

Andy diSessa

Andy diSessa

AS:db
enclosure

APPENDIX B

MIT NSF Summer Program

Application Exam Solutions

We should point out that our purpose in having you do this exam was to get some idea of how you tackle difficult problems in physics and mathematics. Thus we were disappointed that some papers merely gave answers without explanation to some of the problems. The further along you get in these fields the less important merely getting answers becomes, and the more important understanding <u>why</u> a right answer works.

Problem #1

The statement of the problem implied that one could just choose some random value of x, substitute it in, figure out which expression is larger, then give that as the answer. For instance, let x be 0. Then cos 0 is 1 and sin 0 is 0. Hence sin (cos 0) is sin 1, and cos (sin 0) = cos 0 = 1. Hence we're comparing sin 1 and 1, and since 1 is not 90 degrees if you took the problem to be in degrees, and 1 is not $\pi/2$ radians if you took the problem in radians, clearly sin 1 < 1. Hence sin (cos x) must be smaller than cos (sin x). (By the way, you really should have taken the problem to be in radians rather than degrees since the values of cos and sin have no dimensions: they are ratios. In particular they do not have units of degrees for a value. Therefore since they were themsleves used as arguments of trig functions, you should conclude that the arguments were supposed to be in radians, i.e. pure ratio.)

But how can we be <u>sure</u> that this inequality is true for every value of x? Well, there are lots of ways. One way is to graph cos (sin x) and sin (cos x) and compare them. But this isn't a proof.

Another way might be to say, "Well, playing around with > and < signs is hard. Most of the stuff I know about sin and cos involves = relationships. How can I convert this problem into an = relationship?" One way might be to note that we already know one value of x, 0, for which cos(sin x) > sin(cos x). So if this isn't true for all x, there must be some value, call it y, for which the graph of cos(sin x) "crosses over" sin(cos x), a place where cos(sin y) = sin(cos y).

So if we can show that there is no value of y for which

sin (cos y) = cos (sin y)

then we can be sure that cos(sin x)>sin(cos x) for all values of x. To show this note first that a cosine is just a sine curve shifted by $\pi/2$ radians. Therefore

APPENDIX B

$\sin(\cos y) = \sin((\pi/2) + \sin y)$

So either:

(a) $\cos y = (\pi/2) + \sin y$

(b) $\cos y = \pi - ((\pi/2) + \sin y) = (\pi/2) - \sin y$

or:

(a) $\cos y - \sin y = \pi/2 = 1.57\ldots$

(b) $\cos y + \sin y = \pi/2 = 1.57\ldots$

Let's concentrate on solving (b). (It turns out that (a) can be solved by an almost exactly identical method.) Is there any angle for which the sum of both the sine and cosine is equal to 1.57? If so, it would surely have to lie in the first quadrant -- otherwise, either sine or cosine would be negative, and there's no way to add a negative number and another number less than or equal to +1 in order to get +1.57.

So try graphing ($\sin y + \cos y$) in the first quadrant. Note that both $\sin y$ and $\cos y$ are the mirror images of one another about $\pi/4$: $\sin y$ is just $\cos(\pi/2 - y)$. To see this symmetry in an equation notice

$\sin(\pi/2 - y) + \cos(\pi/2 - y) = \cos y + \sin y = \sin y + \cos y$

Now let's use this knowledge to analyze what happens to the graph of ($\sin y + \cos y$) as y varies from 0 to $\pi/2$, that is, through this first quadrant. It will start at 1, then climb (sin starts increasing faster than cos starts decreasing) to some peak, level off, and then symmetrically curve back down to 1. Since the curve is symmetrical about $\pi/4$, this peak must occur for $y = \pi/4$! Hence the maximum will be

$\sin(\pi/4) + \cos(\pi/4) = \sqrt{2}/2 + \sqrt{2}/2 = \sqrt{2} = 1.414\ldots$

which is less than 1.57 ! Hence there's no value of y for which

$\cos y + \sin y = 1.57\ldots$

An even easier way to show that ($\cos y + \sin y$) = 1.57 has no solution is to note that the graph of ($\cos y + \sin y$) is, in fact, just a scaled up and shifted over sine curve:

$\sqrt{2}\sin(y + \pi/4) = \sqrt{2}((\sin y)(\cos \pi/4) + (\cos y)(\sin \pi/4))$

$= \sqrt{2}(\sqrt{2}/2)(\sin y + \cos y) = (\sin y + \cos y)$

so ($\sin y + \cos y$) is just a sine curve shifted over by $\pi/4$ and scaled up by $\sqrt{2}$. But if the scale factor is $\sqrt{2}$, the curve never gets higher than $\sqrt{2}$. Hence it can never be as high as 1.57.

APPENDIX B

You can also use calculus to show this. The derivative of (sin y + cos y) is (cos y - sin y). When this = 0, (sin y + cos y) has a either a maximum or a minimum. Well, (cos y - sin y) = 0 when either y = ($\pi/4$) or ($5\pi/4$). But (sin y + cos y) is equal to $\sqrt{2}$ when $y=\pi/4$ and $-\sqrt{2}$ when $y=5\pi/4$. Hence the absolute maximum of the function must be $\sqrt{2}$.

Thus by three difference approaches we see that equation (b) on the previous page can have no solution. We can prove the same thing for equation (a) by considering the fourth quadrant maximum of (cos y - sin y), and showing it too must be equal to $\sqrt{2}$.

Hence cos (sin x) must be greater than sin (cos x) for all values of x.

Problem #2

First look at the powers of 2:

2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536, 131072, 262144, 524288, ...

so the 19th power of 2 starts with the digits 52.

Can we find such a power for any sequence of digits? Let's try to put this more formally:

Given an integer x, we want to show it is possible to add some sequence of digits (let's say, n extra digits in all) to the right of it so that the result is some power of 2, call it the mth power.

Or in other words.

Given an integer x, we want to show there must be some integers m, n, and r such that

$$10^n x + r = 2^m$$

where $r < 10^n$.

or, transforming this into an inequality:

Given an integer x, we want to show there must be some integers m and n such that

$$10^n x < 2^m < 10^n(x+1)$$

Those exponents in the above expression are messy. It's hard to see how to prove what we want unless we can put m and n into a simpler context. Well, so what tricks do we know for getting rid of exponentials? How about logarithms? So we can rewrite the expression we're trying to prove as

$$\log_5(10^n x) < \log_5(2^m) < \log_5(10^n(x+1))$$

where $\log_5$ stands for the logarithm to the base 5, but you can also use any other base if you like. (Base 5

APPENDIX B

was chosen here because it allows us to change a power of 5 directly into an integer, which simplifies matters somewhat.) This trick preserves the inequality because if $x < y$, then $\log_5 x < \log_5 y$.

So now let's use some of the laws of logarithms:

$$\log_5 x + n \log_5 10 < m \log_5 2 < \log_5(x + 1) + n \log_5 10$$

But $\log_5 10 = \log_5 5 + \log_5 2 = 1 + \log_5 2$, so

$$\log_5 x < m \log_5 2 - n \log_5 2 - n < \log_5(x + 1)$$

$$\log_5 x < (m - n) \log_5 2 - n < \log_5(x + 1)$$

Call $(m - n)$ by a new name, k, and the problem becomes:

Given an integer x, show there exist integers k and n such that

$$\log_5 x < k \log_5 2 - n < \log_5(x + 1)$$

Can we prove this now? Well, logarithms can be thought of as the sum of an integer (the characteristic) and an fractional part less than one (the mantissa). The integer part of this inequality needn't concern us much because if we ever find an integer, k, which gives us the right fractional part for $k \log_5 2$, we can just set n to bring us into the right range. So the real problem is how to be sure that we can find a k such that $k\log_5 2$ has a fractional part between the mantissas of $\log_5 x$ and $\log_5(x+1)$.

We can use an analogy. Think of a clock whose circumference is 1. And suppose the second hand of this clock ticks clockwise in units of $\log_5 2$, as measured along this circumference. Then the question can be restated in terms of this funny clock, "Starting with the hand at the top of the clock, can it fall within some small region between the mantissas of $\log_5 x$ and $\log_5(x+1)$, as measured clockwise from the top of the clock, if it ticks forward some number of times?"

The first step is to show that the number of places on the clock face reached by the hand in this "irrational tick" process is infinite. We can show this by assuming the opposite, that the hand position starts to "repeat" after a certain number of cycles. If a poisition repeats then there must be some number of ticks, k, which makes an integral number of rotations, n, from that position until the repetition of the position. Thus

$$k \log_5 2 = n$$

so

$$5^{(k\log_5 2)} = 5^n$$

APPENDIX B

or since $5^{(k\log_5 2)} = (5^{\log_5 2})^k = 2^k$

$$2^k = 5^n$$

which says that an integer power of 2 is also a power of 5 ! But this is impossible because of prime factorization. Hence the clock ticks off an infinite number of points on the clock circumference.

Now if all those points one could reach were evenly distributed over the clock circumference, the problem would be solved, because that would mean that you could always find a point between two arbitrarily selected other points on the circumference. But we can't be sure this is so, because the points might just "bunch up" around certain points on the clock face and leave other places "blank."

What we'll show is first, the points must bunch up somewhere, and second, that in fact they bunch up everywhere (i.e. are "evenly distributed")!

1) Divide the clock circumference into 10 equal parts. Since our second hand makes an infinite number of ticks, at least one of these 10 parts contains an infinite number of ticks. Now divide that part into 10 subdivisions. One of those must contain an infinite number of ticks. Repeat the process until you have found an interval as small as you like which must still contain an infinite number of points. Thus for any ϵ, no matter how small, you con find an interval of length ϵ or less which contains infinitely many points. This is bunching.

2) Given an ϵ, pick two of the points in an ϵ interval found as above. The points must be less than ϵ apart. If the first point is n_1 ticks and the second is n_2 ticks then we can write an equation which says the distance traveled by the second hannd is within ϵ of an integral number, r, of revolutions.

$$r < n_1\log_5 2 - n_2\log_5 2 < r + \epsilon$$

But this says

$$r < (n_1 - n_2)\log_5 2 < r + \epsilon$$

which means some number of ticks, $(n_1 - n_2)$, effectively moves the second hand a very small distance on the clockface. So if the second hand makes a mark every $(n_1 - n_2)$ ticks, those marks will be closer than ϵ apart and will uniformly mark up the whole face (moving less than ϵ between marks).

To solve the original problem pick ϵ much less than the distance between the mantissas of $\log_5 x$ and $\log_5(x+1)$. Certainly there will be many marks in the "large" mantissa interval. Pick one of them, say the m^{th} mark on the clockface. That mark is $m(n_1 - n_2)$ ticks, hence $m(n_1 - n_2)$ will have a fractional part

APPENDIX B

in the right place.

(Note in this argument $n_1 > n_2$ and mantissa log(x+1) > mantissa log(x). If that is not the case the argument is only trivially changed.)

Problem #3

The television screen contains phosphors which glow with varying intensity in response to the strength of an electron beam passing over the screen. The principle of the phosphor is such that it continues to glow after the electron beam is removed. This glow fades with time, and reasonably quickly, so that most of the time the screen is really pretty dark. But if the electron beam redraws the screen fast enough, it gives the impression of a constant glow -- because the human eye cannot distinguish events happening too close it time. Actually the intensity at some point on the screen varies time something like this:

Notice that sharp jump in intensity. That's probably the most "recognizable" feature of the intensity change. So to solve this problem we ought to make use of it. But it won't do just to stare at particular points on the screen; when things happen too fast the eye can only "average out" the intensity to some constant value.

There are two basic types of solutions to this problem, related but distinct. Both use the idea of converting time into distance. Thus an action which is occurring too fast in time for the human eye to perceive it becomes a much easier problem of perceiving distances.

Horizontal movement solutions

Probably the simplest approach is to take a pencil (actually, you can use your finger alone just as well), hold it vertically, and pass it quickly between your eye and the screen from left to right. Now, as we have said, most of the time any part of the screen is really pretty dark -- it's only when the electron beam passes and for a short time afterwards that the part is lighted. And when this happens it is necessarily lighted quite brightly. So most of the time while you're passing the pencil in front of the screen, the pencil

APPENDIX B

will be contrasted against a dark background: the pencil won't be blocking any screen light. It's only at a few times (or a few places) in the pencil's transit that its background will be illuminated strongly, causing the pencil to block screen light and hence causing a "shadow."

So as you pass the pencil in front of the screen you will see several pencil images or "shadows" spaced pretty much evenly across the screen. These correspond, in the relatively brief time which it took for the pencil to cross the screen from left to right, to each time that the electron beam redrew the screen. The faster you move the pencil, the fewer shadow images you will see, because the shorter transit time allows the electron beam fewer times to complete its cycle, each time bringing the pencil into bold relief as a shadow on the screen.

Now we're told in this question that the picture is not flashed on the screen all at once, but is being drawn in either a top-to-bottom or bottom-to-top manner. Hence this "blocking" effect of the pencil on its background will occur later at the later-drawn parts of the screen. Hence the pencil images will appear slanted even when you are careful to hold the pencil exactly straight up. For instance, in the top-down case the bottom of the screen will be drawn later. Hence the pencil image will appear tilted so that the top is farther to the left than the bottom. (Remember, we're assuming we're drawing the pencil from left to right.) By observing a television you can indeed confirm this tilt, and hence, the picture is being drawn from top to bottom.

(Incidentally, you will note an exactly opposite tilt if you draw the pencil from right to left. In this case, the time lag on the bottom will cause the bottom to appear farther to the left than the top. So it appears to tilt the opposite way. This fact accounts for the curious apparent "bending" of the pencil as you hold it vertically and wave it back and forth in front of the screen. The pencil appears to be made of rubber.)

The time it takes to draw the picture can be found by the same experiment. If you see three pencil images in one pencil transit, that must mean the screen was redrawn three times in the time it took the pencil to cross the screen. So you just need to measure the time it takes the pencil to cross the screen area and divide by three. (Or, if you can move your hand fast enough, the speed at which the apparent pencil slant is exactly 45 degrees to the vertical is the same as the speed of the sweep.) If you have a feeling for how long a second is, surely you can measure (roughly) a tenth of a second or so. Or

APPENDIX B

else you can make an estimate of the velocity of your hand and divide that into how far it travels across the screen.

Vertical movement solutions

If you run the pencil up and down instead of right and left, you will see more pencil images going up than going down. This is because down is "chasing" the scan while up is running against it. Think about that.

Another approach is sort of the reverse, using a light (as opposed to dark) image. Just make a hole in the paper with the pencil, then pass the paper in front of the screen from bottom to top or vice versa. The analysis of either of thse up and down methods is not too hard and we leave it to you.

A variant on this theme might be considered the "stroboscopic" approach. Make a second hole in the paper near its center, stick the pencil through it, and spin the paper about this axis (best to use thick paper). You should see images grouped around a circle, with more images on one side than the other. The side with fewer images is "chasing" the scan i.e. is going in the same direction as the scan. You can arrange things so that the number of images in the circle divided into the time for one revolution gives you a single scan time.

Non-solutions

A number of other interesting experiments were suggested for this problem. However, we are not convinced that any of them will work.

One suggestion was to play around with the vertical hold knob on the set. Maybe if you could get the picture to "flip" at a rate near to its scan rate you could get something like stroboscopic effects. Unfortunately, TV sets are deliberately designed so that this phenomenon cannot occur. The vertical hold knob is just intended for fine adjustments of the signal frequency and phase, and there's little need for it to compensate for errors in frequency as large as the scanning frequency itself. Besides, allowing this much variance in the set would make it awfully hard to adjust.

Another suggestion was to use the vertical hold to get the bottom half of the picture in the top half of the screen, and vice versa. But this doesn't gain you anything because the television picture is still being drawn the same way; it's just the phase of the set relative to the station signal that is off.

Another bunch of ideas regard turning the set on or off. But it's very hard to make any definite

APPENDIX B

statements about what's happening to the set in this process. There are just a bunch of amplifiers controlling the beam position, and it's nearly impossible to predict (even from the specifications!) what a network of amplifiers will do when suddently turned on or off. Often a considerable amount of chance is involved in which amplifier will come up to voltage first in a turnon, or decay slowest in a turnoff, or how fast either of these things will happen. So sets may differ as to what way the lines appear or disappear, or even the same set may act differently at different times. Thus turnons and turnoffs are not very useful experiments.

Another suggestion was to blank out the screen except for a small column. But this just effectively narrows the size of the screen, and doesn't make it possible to see anything new. Related ideas involved studying the edges of the scan lines.

Then there were the solutions that involved just staring at the set and trying to see some kind of motion. But after all, if you could tell that the picture was being drawn in a particular direction on the set, this would be commericially disadvantageous. Flickering sets are hard on the eyes, and Consumer Reports would find out and tell everyone not to buy that model of set. So manufacturers would want to be extra sure this couldn't happen, and make sure that the scan frequency is significantly faster than what a human eye could see.

Then there were the "60 cps solutions", which didn't even require a television. Since standard alternating current is 60 cycles per second, one might reason that the TV scan time is 1/60th of a second. But there is no way to verify this hypothesis except by experiment. Besides, TVs have to generate a lot of other frequency signals than 60 cps (as for instance, the horizontal beam position at about 16000 cps), so there's no reason they can't generate, say, a 50 cps scan frequency. And besides, if 60 cps is possible, why not 30 cps or 120? How could you tell the difference from looking at your set?

Finally, an award for something or other goes to the suggestion of using the paper and pencil to write a letter to the manufacturer to ask for the set specifications.

Problem #4

There was a good deal of variety in the solutions to this problem. Nobody gave a purely analytic solution, as it should be, since the problem can involve some fairly advanced math that way. But

APPENDIX B

unfortunately many people had a great deal of trouble making qualitative, more or less intuitive arguments. Many people didn't seem to understand the point of Newton's Second Law, $F = ma$, which says basically that force is a changer of velocity, not directly a changer of position.

To see this, let's look at a short thrust carefully, one which makes a given change in velocity, Δv.

$$F = m\Delta v/\Delta t$$

During the thrust $a = \Delta v/\Delta t$ is constant (assuming F is) so

$$v = v_o + at$$

Position is given by

$$x = x_o + v_o t + (1/2)at^2$$

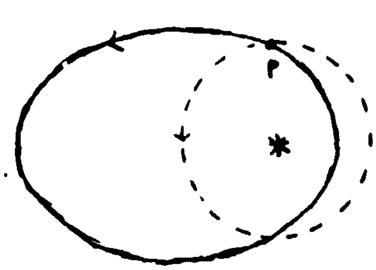
which differs from what would happen without the thrust by the term $(1/2)at^2$. Since, for our thrust of duration Δt we have

$$(1/2)at^2 = (1/2)(\Delta v/\Delta t)(\Delta t)^2 = (1/2)\Delta v\Delta t$$

this difference goes to zero as Δt goes to zero. The moral is that a short impulse does not change position, only velocity.

Thus a short thrust at some point, p, in the orbit (the top) has no effect except to cause the ship to leave that point at a different velocity. A tangential thrust adds a change in velocity which is tangential, the same direction as the old velocity, hence the new velocity points in the same direction as the old one, only it's longer.

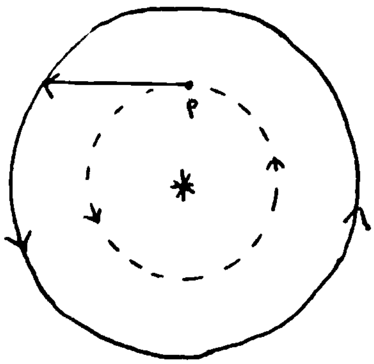
This rules out

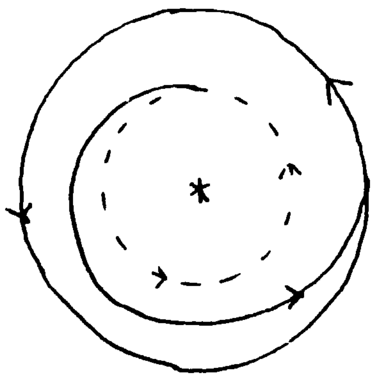


since, in this orbit, the new velocity at the point of thrust obviously points slightly away from the earth (it must in order to cross the old orbit).

Once the thrust is done it can have no more effect. So the following cannot happen:

APPENDIX B

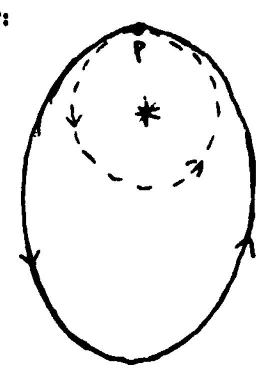




Neither of these situations allows the ship to return to p, as it must, since undeflected orbits always close. (The orbit is undeflected after the thrust.) By the way, the new orbit must go outside the old one for any number of reasons. You can say "more energy" or simply note that with an increased velocity gravity does not have enough time to change the velocity as much as before; the orbit at that point must bend less away from a straight line.

The easiest way to see what <u>does</u> happen is to note that both before and after the thrust v is perpendicular to the radius from the earth at p. Therefore, p must be a minimum radius or a maximum one. (Think about that!) From what we said above, it evidently is a minimum.

If you know Kepler's Law that orbits are always ellipses with the center of attraction at one focus, then what we've said leads to a unique solution below:



Notice particulary the counterintuitive effect that the elongation must be <u>perpendicular</u> to the thrust!

Certainly what we've done can be done with formulas, but the important point is that when you use formulas which don't tell the whole story (as nearly everyone did) you must pay very careful attention to the qualitative information which you have available.

Problem #5

In general, two things may be noticed about the solutions we received for this problem. First, they were often unnecessarily numerical. Sure, this exam does ask for some knowledge of math, but the

APPENDIX B

characteristic of a good scientist or engineer is the ability to sense when numbers are relevant to solving a problem and when they are not. So we had quite a few people answer "no" to part (b) about doing this with broken counters. Usually these people gave a correct solution to part a) using counters. But they jumped to the conclusion that, since their algorithm wouldn't work with broken counters, then no algorithm could work with broken counters.

Secondly, the programs we received were often very hard to read because they lacked sufficient explanation. Many people just sent us a program by itself. But if you know something about programming, you'll realize that naked programs are awfully hard to understand, no matter what computer language they're written in. There's a whole art to "documentation", preparing information along with a program to make it understandable to someone else, and it's a central problem for people who do a lot of programming. We didn't see much of this art in evidence. Remember, an answer per se to some hard problem is not much value unless someone else can understand what you've done.

There are lots of different ways to do this problem. Here are a few. Note that Algorithms C and D can also be used with broken counters. (Note: in all these methods, the order of scanning is first from left to right, then top down, from the upper left corner.)

Algorithm A -- Coordinates method

1. Find the x-coordinate (counting squares from the left) of the first black square, and save it in c1.
2. Find the x-coordinate of the next white square in the row, and save it in c2.
3. Look at next row. If it contains no black squares, go to step 8.
4. If x-coordinate of first black square is not the same as c1, say "no".
5. If x-coordinate of next white square is not c2, say "no".
6. If there are any more black squares in this row, say "no".
7. Go to step 3.
8. If there are any more black squares, say "no", otherwise say "yes".

Algorithm B -- Row and column measuring method

1. Find the first black square.
2. Measure the number of black squares in this row until the next white square, put in c1.
3. If there are any more black squares in this row, say "no".
4. Go to the next row.
5. Find the first black square. If none in this row, go to step 9.
6. Measure the number of black squares in this row until the next white square.
7. If this number is not the same as c1, say "no".
8. Go to step 3.
9. Repeat steps 1 through 8 for columns instead of rows.
10. If you still have not said "no", say "yes".

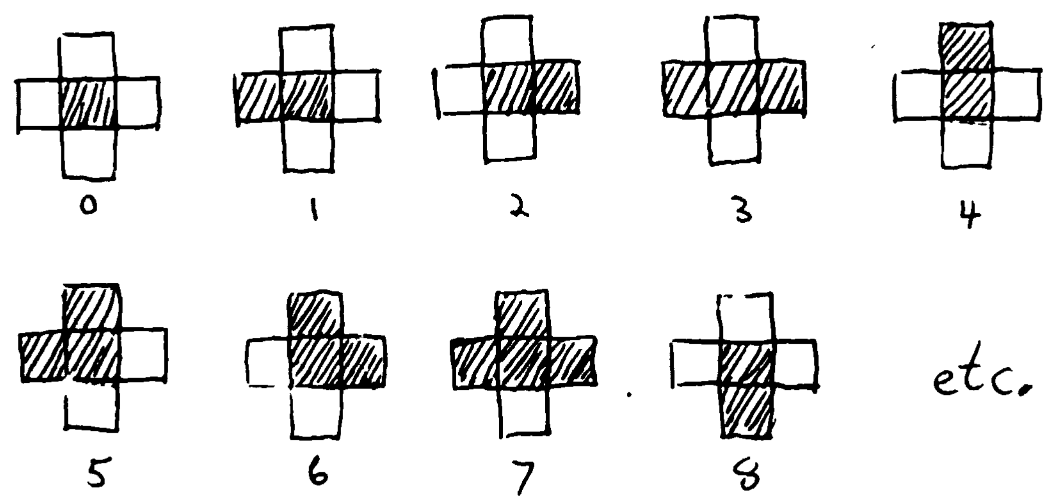
APPENDIX B

Algorithm C -- Row-matching method

1. Find the first row containing black squares.
2. If the black squares are not all together in one clump, print "no".
3. Go to the next row.
4. Is this row identical to the one above it? If so, go to step 2.
5. Are there any more black squares in this row or rows beneath it?
6. If so, print "no", otherwise print "yes".

Algorithm D -- Square neighbor classification method (for rectangle with sides > 1 block)

1. Define the following code for classifying each square, which involves summing four numbers. These numbers are 1 if the left neighbor is black, 2 if the right neighbor is black, 4 if the upper neighbor is black, and 8 if the lower neighbor is black.
2. If there is no or more than one black square each of neighbor type 5, 6, 9, or 10, print "no".
3. If there are black squares of neighbor type 0, 1, 2, 3, 4, 8 or 12, print "no".
4. Otherwise print "yes".



Appendix C

The following article about the summer program appeared in the MIT newspaper *Tech Talk* on August 25. A slightly revised version also appeared in the *Boston Herald Advertiser* on September 12.

NSF Summer Project Wins Praise in All Quarters

By KATHARINE C. JONES
Staff Writer

For the first time this summer MIT ran a Student Science Training Program (SSTP) made possible by a National Science Foundation SSTP grant.

The program lasted from July 12 through August 21 and was offered by MIT's LOGO Group, a research project of the Artificial Intelligence Laboratory and the Division for Study and Research in Education. The LOGO Group is concerned with applying computers to human education and development.

The fourteen students registered in the program were the first high school-age group to study intensively under a computer-based curriculum developed by LOGO for teaching mathematics and science. The curriculum has been developed over the past six years and is a prototype of the kind of educational innovation which will be made possible by the widespread use of computers in schools.

"What's so great is learning a new approach to problem solving," said David Glaser of Shrewsbury, Mass., who will enter his senior year at Shrewsbury High School in September.

Daniel Yates of North Pompano Beach, Fla., and Pine Crest High School, termed the summer "very enjoyable and educational," although he said at first he was "sort of overwhelmed" to discover what computers are capable of doing.

The typical class day lasted from 9am to 5pm with an hour break for lunch. Focus of each day was an hour-and-a-half lecture on subjects such as geometry, topology, differential geometry and physics. Students spent the remainder of the day using the computer to complete problem sets for the lecture or to work on one of three special projects.

The special projects are considered the most interesting aspect of the summer program by the two project directors, Dr. Harold Abelson, and Dr. Andrea diSessa, special lecturer, of the MIT Division for Study and Research in Education.

One student group worked on a physics project developing a computer program that simulates mechanical stress. The students designed structures and simulated stress in a variety of situations. For example, they might "build" a bridge and then try to collapse it.

More advanced applications of this program could lead to work with bones.

"It is known that bones assume their shapes to some extent as a result of stress on them," Dr. diSessa said. "With this computer program we can try to 'grow' bones. Our hope is the program will be a good model of the way nature works."

Director of the physics project was Margaret Minsky, a senior in mathematics from Boston, who worked with the group as part of an Undergraduate Research Opportunities Program (UROP) project. She is the daughter of Dr. Marvin Minsky, Donner Professor of Science in the MIT Department of Electrical Engineering and Computer Science.

A second student group worked with a projection color television set (like the giant Advent television set in the Student Center) modified to operate under computer control. They drew pictures by programming the computer and used color to draw three dimensional figures. By making designs, they learned about visual and color perception.

Director of the color project was Henry Lieberman of Cambridge, a member of the sponsored research staff of the Artificial Intelligence (AI) Laboratory.

The third student group attached an electromyograph (EMG)—a machine that records the action potential of muscles in a living subject—to a computer. Their aim was to develop a system which would allow someone to program a computer through selectively controlling muscle tension. Such research ultimately could benefit severely handicapped people who cannot type or hold a pencil.

Director of the bio-feedback group was E. Paul Goldenberg of Newton Highlands, also a member of the sponsored research staff of the AI Laboratory. He was assisted by Bruce Edwards, a senior in electrical engineering from Boston, and Ellen Hildreth, a senior in mathematics from Ashland, Mass., who are both working in LOGO on UROP projects. Neil Rowe, a graduate student in electrical engineering, also assisted Drs. Abelson and diSessa in running the Project.

As the program began its final week Dr. Abelson said, "I'm very pleased with the results of the program and the students' accomplishments. They adapted well to the cooperative atmosphere the program encouraged."

Dr. diSessa said, "An environment where students have as much computer time as they want is a highly stimulating environment. It lets them be active in their learning."

NSF has awarded SSTP grants annually since 1959. This year there are 125 projects involving 4,400 high school students. Essentially all projects are held on college campuses and run by the colleges.

HIGH SCHOOL STUDENTS David Glaser (standing) and Daniel Yates work on a bio-feedback project as part of Student Science Training Program sponsored this summer by MIT's LOGO Group. An electromyograph (EMG) is attached to Mr. Glaser's arm and records muscle tension. Mr. Yates monitors the computer console. Mr. Glaser can make a design on the console by regulating muscle tensions in his arm.