IEEE
Annals
of the History of Computing

Volume 42 Number 2 APRIL-JUNE 2020

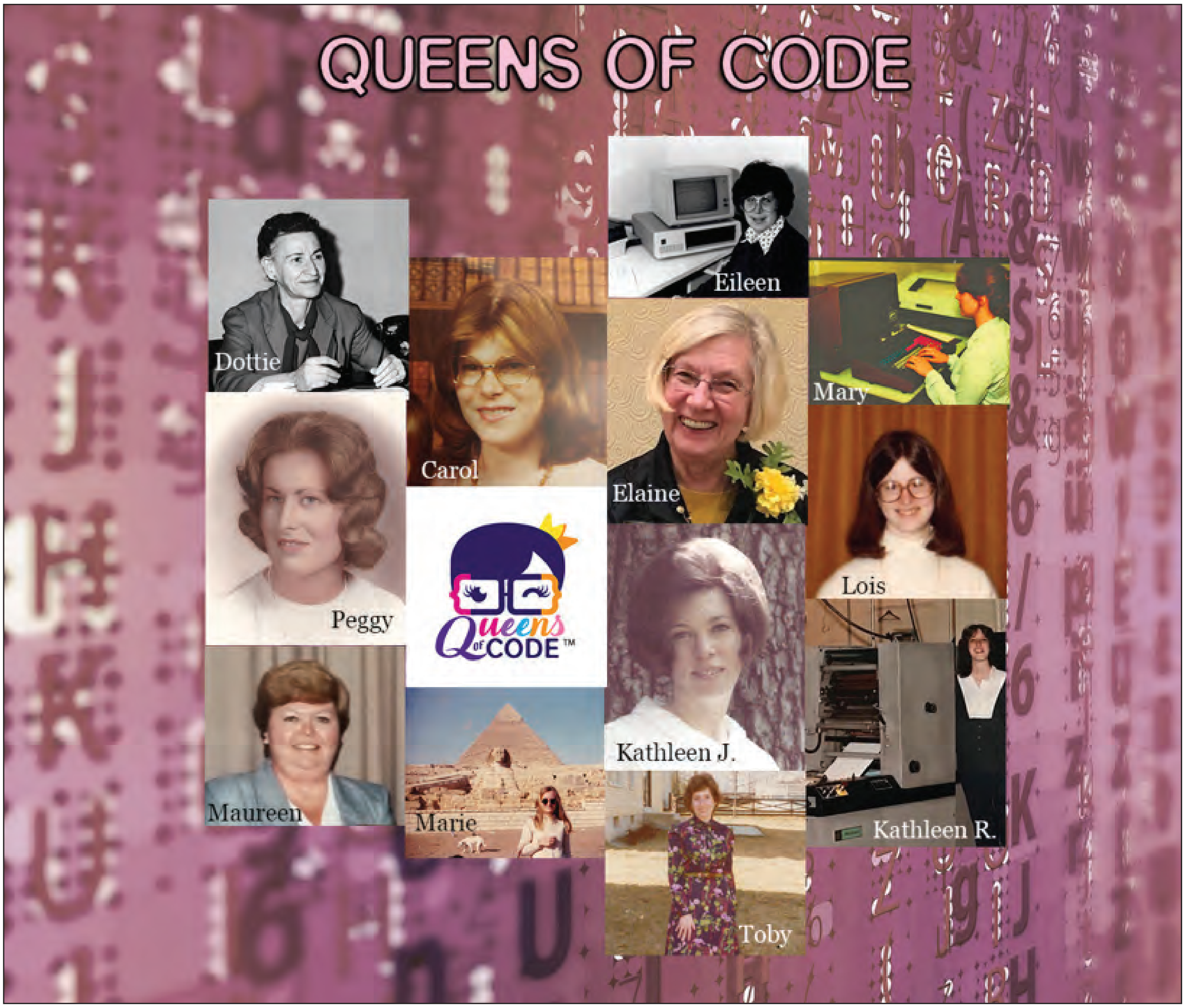

Image Courtesy of Eileen Buckholtz

Queens of Code

IEEE

IEEE COMPUTER SOCIETY

www.computer.org/annals

Contents

April-June 2020
Volume 42 Number 2

Published by the IEEE Computer Society

Departments

ISSN: 1058-6180

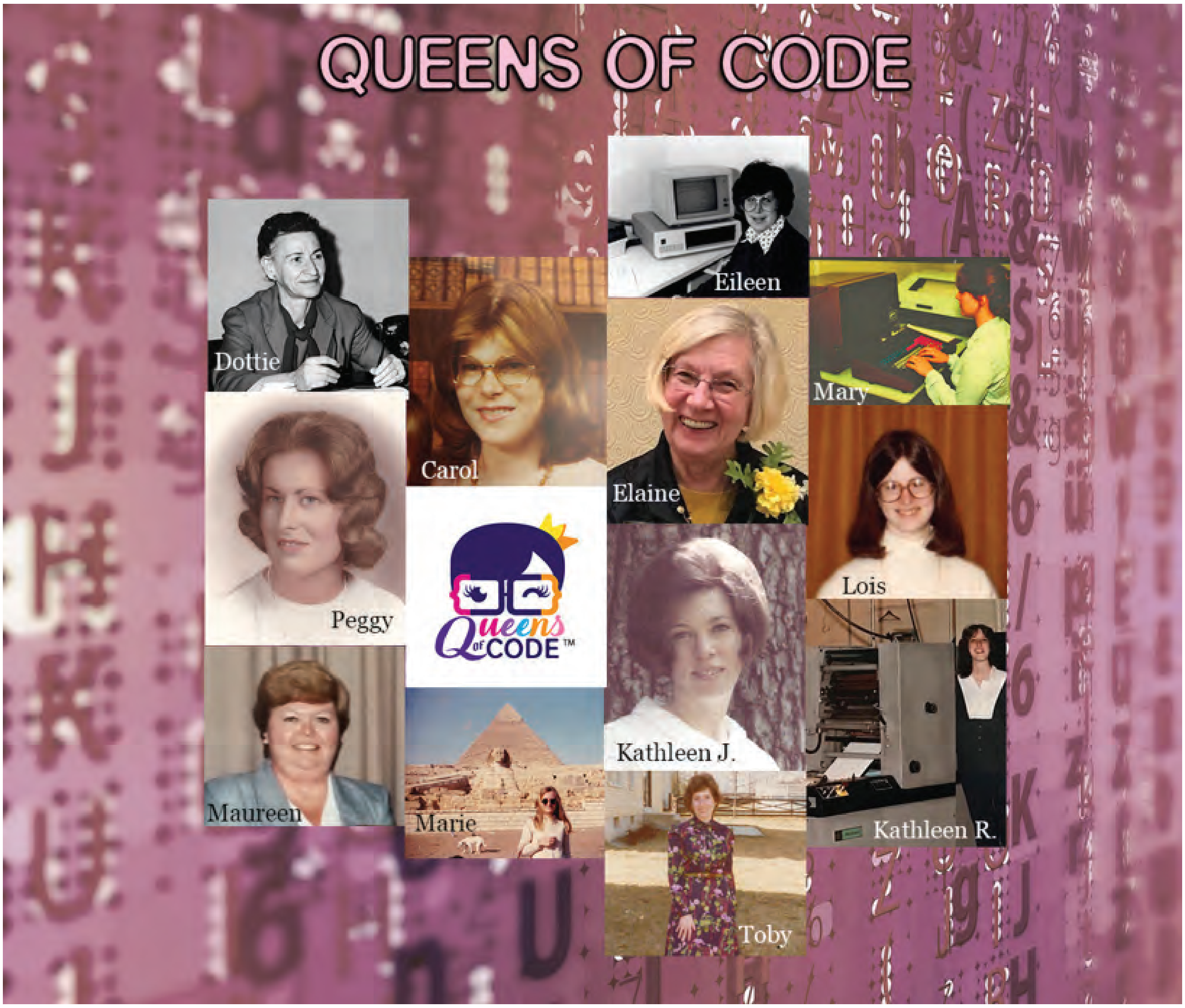

Image Courtesy of Eileen Buckholtz

IEEE Annals of the History of Computing

Column

From the Editor's Desk

Gerardo Con Diaz
University of California, Davis

I am thrilled to introduce our second issue of the year. We open in Italy during the 1950s, when the University of Pisa and Olivetti led the first Italian project to create an electronic calculator. Giovanni Cignoni and Fabio Gadducci dive deep into the University's archives to reassess the emergence of computer science in Italy as a business and a scientific discipline. From Pisa we move to Binago, the small town north of Milan where an internationally renowned model maker named Roberto Guatelli was born. Silvio Hénin and Simona Casonato examine Guatelli's life and work to reflect on what it means to create, preserve, and trade replicas of ancient machines. Travelling east, we go to New Zealand to reassess the standard narratives about its first computer. Brian Carpenter shows that IBM was not, in fact, the first brand of computer to arrive at that country, as an ICT 1201 had already been in place for months by the time the New Zealand Department of Education acquired its first IBM 650. We end this world tour of computing among a global community of astronomers. Michael Scroggins and Bernadette Boscoe examine the history of FITS, the file format that astronomers have used since the 1980s to overcome incompatibilities between their operating systems.

Dave Walden, our Anecdotes editor, has prepared two anecdotes for us. First is a very unusual one by Eileen Buckholtz that recounts the origins of *Queens of Code*—a project she is leading to preserve the histories and experiences of the women who worked in information technology at the National Security Agency from the 1960s to the 1980s. This Feature Anecdote gives us a glimpse of an exciting ongoing effort that has already recorded the histories of 75 queens of code. Zbigniew Stachniak reflects on the history of the MCM/70, one of the earliest computers for personal use to be mass manufactured. Stachniak has written extensively about this computer before, but new insights about its design and marketing emerged after the York University Museum in Toronto acquired additional prototypes of the machine. This anecdote reveals that we need to think about Toronto's computing history more carefully when we account for the history of personal computing.

The issue closes with an interview with Mike Williams, the 2007 IEEE Computer Society, and a book review of Meryl Alper's book, *Giving Voice*.

I also have exciting news to share with you: Thomas Haigh and Mark Priestley were awarded the 2019 Bernard S. Finn IEEE History Prize for "Colossus and Programmability" (*IEEE Annals* 40:4). Congratulations, Tom and Mark!

To end this letter, I would like to express my gratitude. Even despite the chaos and uncertainty that surround us, the *IEEE Annals* is moving forward with its second issue of the year. This was possible because of the hard work and collaborative spirit of many members of our community. Our authors, department editors, article editors, and reviewers volunteered time from their unpredictable schedules to make this issue possible, and the IEEE staff was always ready to support us. Thank you.

Digital Object Identifier 10.1109/MAHC.2020.2992095
Date of current version 29 May 2020.

Department: Anecdotes

Queens of Code

Eileen Buckholtz
Director, Queens of Code Project

INTRODUCTION TO THE QUEENS OF CODE

Queens of Code is a women's technology history project—a collection of stories, experiences, and insights from women who worked in information technology at the National Security Agency (NSA) in the 1960s, 1970s, and 1980s. NSA's computing women programmed and managed the most sophisticated systems of their day and I was one of them. I started this project in 2018 to collect the stories of the agency's women technology pioneers and recognize their contributions because I believed that if we did not document these stories now while many of us are still living, our history would never be told. The National Cryptologic Museum and NSA's historians offered encouragement. I reached out to women I had worked with, and dozens signed up. Participants were asked to complete a detailed questionnaire and write their stories. All material had to be approved through NSA's prepublication review. We have been networking online for almost two years and have more than 75 women in the group. The goals for the project are recognition of the Queens of Code in the history of computing, expanding the understanding of how women worked in early computing, and inspiring more young women to pursue STEM careers. We are sharing our stories in presentations, articles, and interviews.

Because these NSA women's jobs were often top secret and they worked on the most sensitive national security programs, they could not discuss what they did, even with their families. In many cases, they could not even confirm they worked for NSA. They and their computing activities have been, for practical purposes, a secret for more than 50 years.

Women have always been in the workforce—although their contributions to science have often gone unrecognized. In the 20th century, women worked for the U.S. government and military, not just in clerical, nursing, and other "women's" positions, but in specialized technical fields such as cryptology, mathematics, and computing. The U.S. military during World War II actively recruited educated and talented women, including those from some of the best colleges, to fill critical vacancies and to "free a man to fight." These women often found themselves doing tedious work, but gained a foothold in the technical workplace.

According to Liza Mundy's *Code Girls*, over 10,000 women were a critical part of the cryptologic mission, some working with the early computing machines.[1] In the U.S. many women who had technical skills were sent home after the war to free the jobs for men returning from war. More generally, women's place in computer history has not been publicized because it has largely been HIStory, focusing on hardware and the male inventors,[2] as I saw on my visit to the Computer History Museum in Mountain View, California, in July 2018.

Fortunately, modern cryptology, in particular, was welcoming to women from the start. Elizebeth Friedman and Agnes Driscoll led the way in the 1920s and 1930s.[3] The work of the "Code Girls" during World War II was critical for

Digital Object Identifier 10.1109/MAHC.2020.2982751
Date of current version 29 May 2020.

winning the war. Like their contemporaries at NASA, whose story was told in the bestselling book and hit movie *Hidden Figures*, the women at NSA, walking in the footsteps of their World War II sisters, have broken ground from the 1960s on as they contributed to advances in computing in the world of cryptology.[4]

Many of the Queens of Code were recruited by NSA right after college and worked in computing technology for 30, 40, and even 50-year careers. I was one of those queens, hired in 1970, with one of the first undergraduate degrees in computer science in the country. Starting from data systems interns and rising to senior leaders and computer science experts, we were on the forefront of computer technology development. In the 1960s, 1970s, and 1980s, our agency had the most sophisticated computers in the world as well as the most challenging information processing requirements. By 1968, NSA had more than 100 computers spread over five acres of computer rooms.[5] The inventory grew rapidly over the next decades as we and our male colleagues worked with many vendors to drive new system development to meet our big data processing needs.

Our stories may also provide some insight to companies today that struggle to recruit and retain women in tech. In contrast to corporations and institutions in various other sectors, NSA did a lot right over a 50-to-60-year period to recruit, develop, and retain their computing women. They had learned from previous experience with the Code Girls during World War II that women were a valuable asset to their mission. They invested in us through training, intern programs, and advanced degrees, paid equal starting salaries for men and women, gave women responsibility and credit, promoted many women to senior management and technical positions, and provided a good work/life balance. Fortunately for us, most of the men we worked with were supportive as well. Of course, there were some struggles along the way, including a class-action lawsuit over fair promotion in the 1970s,[6] but we prevailed. The Queens of Code made a daring leap into a new career field of computer science and found innovative, exciting, and rewarding careers that contributed to the high-tech world we live in today

The rest of this article highlights some of the experiences I, and many other women, shared working on our first computers.

FIRST ENCOUNTERS OF A BINARY KIND

If you grew up in the 1980s or 1990s as part of the millennial generation, your first experience on a computer might have been with a personal computer at home or at school. You might have learned to program in basic using my Micro Adventure books[7] on an Apple II, Radio Shack TRS-80, Atari, or IBM PC. If you are part of Generation Z, you probably played games on your first computer tablet or smart phone maybe as early as a toddler. E-books, apps, and online shopping and learning are things you took for granted.

That was not the case when the Queens of Code were young. The ARPANET (the early version of the Internet that had just begun to come online in 1969) only connected some dozens of government agencies, universities, and other research organizations, and the World Wide Web had not been invented. Back in the early 1970s, one of our offices did have a terminal that we could use a modem to dial into the National Bureau of Standards' ARPANET. From NBS, we could connect to the Stanford Research Institute—and it took dozens of steps to send a line of text along with manually calculated checksums (a digit that was the sum of the other digits in a piece of data used to detect errors).

When we were growing up, there was little digital computing technology in the schools we attended before college. Pocket-size calculators made their debut in the 1970s. Before then, in high school or college, we used a slide rule (the manual device invented in 1620) for math, chemistry, or physics courses.

Many of us were 18-to-22-year olds when we met our first computer, perhaps an IBM 1620, 1401, or even 360 (after its release in 1964) at our college or university. Often the Queens of Code's first computing experiences were on their initial assignment at NSA or at college. These computer installations could be huge and expensive, especially those in NSA's extensive basement sometimes taking up spaces as big as a couple of basketball courts, cooled by water under the floors to make the rooms so cold that you had to wear a heavy sweater or jacket when working there.

Some of our first computing experiences were on computers with limited capacity and programming done in assembly language or even octal, and that was not easy. We had to be crafty to make the programs work within the constraints. FORTRAN, the first commercially available computer language to use a compiler was released in 1957. A compiler meant that the code could be written with higher level and easier to manage commands that would automatically generate the assembly language or machine code needed for a specific machine. FORTRAN was designed to provide a language for the scientific community, and NSA certainly fit in that box. As computer technology advanced and memory size increased and became less expensive, programmers could write code with less computer specific restrictions.

Dottie Blum, a legendary computing woman at our agency, was using FORTRAN as early as 1954 even as it was being developed by John Backus and team at IBM. At first, people wondered if using a compiler would produce code as efficient as writing in assembly language. But over time, computer speed and memory size increased and convenience won out. Another benefit was that programs could be ported (moved over) to run on other machines that had a FORTRAN compiler. It was a big improvement over having to rewrite programs in another assembly language every time a new computer came into our collection.[8]

Programming was a little like cooking. You had input (like your ingredients) and then steps that processed the ingredients. If all worked well, you have something to eat for supper. Fortunately, I was a better programmer than a cook.

Our first programs were either assignments at school or "toy" programs we were assigned to learn how the computer worked. On the earliest computers from the 1950s like the special purpose ones built in house, there was only the basic documentation, so you had to figure things out for yourself. Sarah, one of our first programmers, used to say that when she started at NSA the computers took up a whole room and you were lucky to find a small notebook with instructions. By the time she retired, the computers were small enough to fit on your desk, and you had a bookcase of manuals and online documentation.

In our environment, the programming process worked as follows. The first step was to define the problem. In our case for application programs, this meant talking to the analyst to understand the problem that needed solving. The problem was often to automate a time-intensive manual process such as an attack on a cryptographic code we had collected by analyzing signals or language translation. NSA processed tons of data to produce intelligence reports for government decision makers including the President and the military. NSA was doing "big data," long before the phrase was coined in 2005. Programs were written to support requirements at the time.

The programmer would then break the process down into small steps that would provide a solution. Programmers often use flowcharts to block out the steps that need to be taken. We used plastic templates back then to draw the flow charts.[9] Now there are many software tools and applications to help with program design.

Next, we had write the code in a programming language like FORTRAN, PL1, or C or in assembly language in the earlier days. Then, we had to debug it, resolving all the problems that we could find. After that, we tested with our real-world users; and, when all was working, officially declared the program live. Of course, there would always be more bugs that popped up, and we had to fix those in a timely manner.

At the agency, system programmers who worked on the operating systems and networking were in the C (for Computer) Organization (which was later reorganized and renamed T (for Telecommunications and Computer) Organization. It seemed that every three or four years we have a major reorganization, sometimes corresponding to a new Director's arrival. Some application programmers started out as part of C, but later moved out to sit with the users in the production organizations. All the reorganizations and reassignments were confusing. One of my bosses had a sign in his office that read, "Perfect reorganization is only achieved by groups on the verge of collapse."

LEGACY QUEEN'S FIRST ENCOUNTER WITH A COMPUTER

Dottie Toplitzky Blum, 1950

Dottie had worked with the Electronic Adding Machines (EAM) equipment and the Army's version of the BOMBE, an electromechanical device developed by Joe Desch of National Cash Register during WWII to decode Enigma messages. Another of Dottie Blum's earliest binary encounters was with the Standards Eastern Automatic Computer (SEAC), which was built in Washington, DC, USA, for the National

Bureau of Standards. The SEAC was one of the first U.S. stored program computers. Dottie then worked for AFSA, the Armed Forces Security Agency, NSA's predecessor. AFSA did not have their own computer but the support organization did manage a number of calculating and cipher machines including the Navy's and Army's BOMBEs that were used in the war effort. These earlier devices were not actually computers since they lacked memory or ability to do anything outside of their limited computational functions such as compiling and comparing text, searching for cribs (plain text), or calculating statistics.

It was 1950 and Dottie and Sam Snyder, one of her coworkers whose computer history writings documented this story, got an urgent request from the Navy's Communications Security Division. The job required the verification of a few hundred involutory 4 x 4 matrices[10] that were used in the Navy callsign system. The SEAC's memory was only 512 (45 bit) words, which was pretty limiting. Back then, they had to negotiate time on the SEAC to debug the program, and the only time NBS would allow them to purchase (at $24 an hour) was after midnight or on Sunday afternoons.

The program was written but they needed test data. Dottie, who was working for the Machine Production Organization as an IBM specialist, produced thousands of random numbers on punched tape to be used to test the application. The SEAC took between 8 and 15 seconds to process each matrix and then printed out the ones that met the "useful" criteria. With a lot of work and some late nights and weekends, Dottie and Sam got the information to the Navy in a timely manner to help solve the problem. Sam said, "Those who participated in this task found the experience 'frustrating, exhilarating sense of accomplishment and participation in making history.'"[11]

MORE FIRST-HAND ENCOUNTERS FROM OUR QUEENS OF CODE

Carol McWilliams, 1967–1970

My first programming experience was assembly language on a CP818 (UNIVAC 1224) for field installation. We "wrote" our programs on a Kleinschmidt—something like a typewriter, but it produced punched paper tape with one instruction per line (e.g., "clear register"). You could fix an error by wrapping Scotch tape over the holes in the line and repunching the line! Fortunately, the readers were not sensitive to the opacity of the tape, just the holes. The resulting paper tape was wrapped butterfly style in a figure eight with a paper clip in the center and stored until you had time on the computer.[12]

The first programs I wrote were standalone processes. I had data input from magnetic tape, ran the program, and produced data output. I scheduled computer time and, when it was my time, I took my paper tape and mag data tape to the computer room. After loading the paper tape into memory, I put my magnetic tape on the spool and initiated the program. When I was done, I took my program tape, mag tape, and results off the computer, cleaned the heads for the next programmer, and took everything back to my desk to assess the results and debug my program. Very much a hands-on process!

Eileen Buckholtz, 1968-1969

I had transferred to Ohio State University (OSU) as a math major and was taking a fourth course in calculus while struggling with the theoretical proofs in the class. I remember the professor covering a big blackboard with the proof of the Heine–Borel Theorem. He got near the end, realized that he had made a mistake and started to erase half his scribblings. My eyes were glazing over. What was I doing here?

Later that afternoon, I heard that OSU was opening their computer science department and they were looking for students. My boyfriend Howard was in engineering and he heard the same thing. Turns out they were offering degrees in both the Arts and Sciences Department, where I was enrolled as well as an engineering computer science degree. We both signed up, became OSU's first computer science graduates, and have been computing together for over 50 years.

There were only several dozen students in the first computer science classes. It was love at first byte for me when I took my first programming course. The initial assignment was a simple sort. The next was to use a random number generator to simulate shuffling a deck of cards and a matrix for holding all the hands. The idea that you could learn a language like FORTRAN and make an enormous computer do your bidding with structured commands was just fun. As we got into more advanced programs, it became challenging as well.

An IBM 360 installation including CPU, tape drives, IO controllers, and other peripherals were housed in a big computer room that took up much of an upper floor of the engineering building. We could see into the computer room through big windows but we were not allowed to go in. After class, we would design our programs and then punch up Hollerith cards (one line of code per card) on keypunch machines, submit them over the counter, and then wait 5 or 6 hours for them to run and get our output back. If we made an error, we had to correct that and submit again. No wonder the four women in the computer science program were all dating guys also in the program. Who else would want to spend Friday and Saturday nights debugging programs?

Elaine Mills, 1965

As part of a work-study program at Towson State College (known today as Towson University), I was studying to become an elementary school teacher. I was privileged to be assigned to a special project to "computerize" all the records for the five Maryland state colleges. Using Hollerith cards and becoming proficient in writing FORTRAN programs in the mid-1960s was a "blast" that actually proved to be a tremendous personal boost a few years later at NSA.

Kathleen Jackson, 1967

My orientation started with a tour of the "basement," a huge area where the computer I would be using, the UNIVAC 1108, was located. The system was so large that it nearly filled the whole computer room, since it had several printers and other pieces of peripheral equipment attached to it. My job was to remove computer printouts from one of the printers, review the data, look for data "anomalies," and adjust the FORTRAN software as needed to fix them. It seemed challenging and interesting at first but, as the days and weeks rolled by, reviewing rows and rows of 1s and 0s became a little tedious to put it mildly. However, I persisted. After completing my tour, I looked forward to my next assignment. Over the years, I sometimes thought about that initial assignment, and how different it was from all the other work I had done at the Agency.

Several years later, I came across a report that contained Agency historical information. It included information about the data that was processed on that UNIVAC 1108 computer I was supporting during my first assignment in 1967. This report identified how critical those data were to national security at that time. Suddenly, I became acutely aware that the many hours I had spent reviewing those 1's and 0's were definitely worth the challenge of the task. In the end, I determined that this work was probably some of the most significant work I did during my entire career at the Agency. I took pride in knowing that this work was very important to the security of our nation.

Kathleen Reading, 1982

"Oh great, another girl!" Imagine hearing those words upon meeting your supervisor for the first time. I was 21 years old and just beginning my 34-year career in the Information Assurance Directorate (IAD), in the Agency's print shop. I was taken aback by my boss's comment, but did not say anything as I was just starting a new job and did not know what to expect. I do remember thinking I was going to do everything in my power to change my boss's mind about what "girls" could do.

My job title was "Reproduction Worker," and I was one of three women working in the shop. I found that job title pretty funny. I first started working in the bindery, and then also ran a printing press, large Xerox machines and printers, and eventually worked in the Electronic Printing and Publishing (EPP) branch. In the EPP branch, for the first time, documents to be printed were sent electronically via computer by Agency customers; and documents also were sent electronically to the printers for printing. One of the documents printed on the night shift was a daily report that was couriered each day to the White House.

As it turns out, I did prove to my boss that women are good workers. I was promoted several times, and was also one of the first women selected to participate in the Agency's first ever production trade program.

Mary Clulow, 1977

"I will rule these machines; they will not rule me." Quietly determined, I spoke these words late one evening at work while trying to complete a typing task. I was using an IBM Magnetic Card Selectric (aka MagCard) typewriter. This was quite a bold statement for an entry level Clerical Assistant at the National Security Agency. However, I had been challenged by this machine more than once since learning to use it two years prior, in 1977.

For those who may not be familiar with the MagCard, it was quite state of the art for its time and was the upgrade to its predecessor, the Magnetic Tape typewriter. Basically, after typing on bond paper, it recorded one page at a time, provided you inserted a magnetic card (much like an IBM card, but Mylar and magnetic) and pressed Record, *before* turning the machine off—or else it was not saved. Once recorded, the file could be edited by picking the related card to the desired page from the labeled envelope, inserting it into the card reader, and then pressing Read. The file then could be played back one page, one line, one word, or one character at a time. The playback was rather quick, making it easy to miss the mark. Sometimes, the paper ripped during a return motion. This was one of those moments; blame it on user error, but I finally thought, enough was enough, and enrolled at the local community college.

This was the beginning of my journey into advanced learning, leading to a BS degree in information management, but definitely not the end of using other machines that would enter NSA's workspaces. They provided more word processing technologies, office automation, and then advanced further into end user computing.

Maureen McHugh, 1969

I graduated from Marywood College in May, 1969. My experience with computers was limited. I was a Math major in a college whose curriculum focused mainly on training teachers. I did not want to be a teacher. It was obvious in 1969 that computer work would be an exciting field and I could get in on the ground floor. As a senior in college, I took a FORTRAN II class.

The teacher was a business professional who taught a few night classes at the college. He had a customized van in which he had a card punch machine, a printer, and sorter. It was only accessible a few hours a week including class time. Writing and debugging our "toy" programs was difficult, to say the least. We submitted our punch card deck to the teacher, and he would return it the following week after running it on a computer back in his office. One turnaround per week! A single typo could set you back two weeks. I think I got a B in the course, but certainly did not feel as though I had mastered FORTRAN.

Peggy Strader, 1969

During my intern tour, I was introduced to the UNIVAC 494 and the SPRYE assembly language. This was an octal-based system so I learned and became proficient in reading dumps in octal. On this system I honed my skills in SPRYE, FORTRAN, and ALGOL. I believe this group was responsible for the first Information Storage and Retrieval System named TIPS (Technical Information Processing System) and its retrieval language named TIPS Interrogation Language. On this system, we were able to put our queries on model 35 teletypes and it would search magnetic tapes of data or magnetic drums for the information requested. I became the user/customer interface for these systems, often teaching the Boolean logic and constructs necessary to retrieve the information needed.

Lois Gutman, 1970

I had no real computer or programming experience other than creating small card decks for overnight runs on a cardpunch machine in a summer job at Johns Hopkins University. NSA's high-level programming language at the time was IMP,[13] running on the operating system FOLKLORE, NSA's homegrown time-sharing system, developed by the Institute for Defense Analyses in Princeton, NJ, USA. Everyone in my office used CDC-6600 computers and sat in a large open tube room, a room full of Cathode-ray tube (CRT) terminals connected to a mainframe where programmers could work on their code in the NSA Headquarters building basement. Operators hung large magnetic tape reels for users. We stacked the reels on our desks (under sheets of black cloth for security) and made hanging decorations from colored plastic write rings.

Toby Merriken, 1970

Fresh out of college, I went to work for NSA in 1966. I started out in the Cryptanalysis Intern Program and became certified as a professional in that field. Shortly after that, in 1970, I joined a newly created branch dedicated to using computer science for the first time for cryptanalytic applications. With no computer training or experience, I wrote programs in FORTRAN and learned a lot on the job.

I wrote each program in longhand and took it to a staff of key punch operators to transfer to keypunch cards. I then took the cards to the remote job entry (RJE) room housing a printer and a card reader, into which I manually fed the cards. The input went to the computer mainframe and was printed out on the printer in the RJE room. The computers were high tech for the time but not interactive. A great deal of time transpired between the writing of a program and the implementation of it. I left this branch in 1974 to return to cryptanalysis and eventually became a linguist.

Marie Rowland, 1970

When I started at NSA, I can honestly say I knew nothing about computers. I had gone to an all-women's college and majored in math. The only exposure to computers we experienced came one afternoon when a guest speaker explained to us that we would all have to learn a language called FORTRAN. So, when I landed at the Agency with my group of new mathematicians, management decided that with my background I should start at the beginning.

I was assigned to the Research organization, where they handed me a small box and explained they were doing research on how small computers might one day be used to schedule jobs for large computers. My task was to teach the small box to tell time. Now, I was naïve enough to believe this and did not realize that it was actually a good exercise for me to learn about how to program computers and really understand them from the inside out.

I started reading the manuals and some library books. Each morning I plugged in my little box and got it going with a paper tape from a teletype, starting a heartbeat interrupt, a periodic signal that the hardware generated to indicate its working or to synchronize other parts of the system. I could count these beats and get up to a second, then a minute and so forth, and thus tell time. The little machine only had a few instructions such as *load, compare,* and *store* so it seemed much easier than the FORTRAN description. The biggest headache was working with the teletype and paper tape. An "all thumbs" affliction was to plague me through card decks and keyboards, through all my years of writing code.

The library books said I could name my variables anything I wanted. I took this to heart and called them names from the book I was reading, *The Hobbit*. Thus, "BILBO" became the second counter. Eventually, the person guiding me looked at my work and gently mentioned that it was traditional to name the variables after the function they performed so other people could follow the program. DUH! Later, as I was finishing the project, I asked if he thought I should put in a routine to handle Y2K—something I had discovered in my library research. I do not know how he kept a straight face when he replied that it probably was not necessary for the purpose of this project. I often remembered this Y2K-innocence when it struck with a vengeance years later.

As it turned out, teaching this little machine to tell time was a very good introduction to the world of computers. Programming it illustrated the "edge" of the hardware and software divide and left me completely fearless to wade into all kinds of hardware–software issues. I realized what a leap it was from the early computer greats made in the 1940s and 1950s when their research allowed them to move from purpose-built machines to building machines that kept both the instructions and the data that the instructions worked on in the same form. I went on to write many programs and eventually received an MS in Computer Science from Johns Hopkins, but I was always able to view the complexity of tasks through the lens of my first project.

REFLECTION

We all had memorable encounters with our first computers and went on to have rewarding careers in technology. Our Queens of Code are good examples of how women were working with early computer technology. NSA gave us opportunities to excel in this exciting new career field.

Over the past 50 years, women have continued to bring their talents and skills to the technology revolution. We hope our project will encourage other women computer pioneers in both the public and private sectors to step forward and tell their stories. While much has been written on the low percentage of women graduating with computer science degrees[14] and problems with retaining female technical employees, it is critical for the future of tech that women's ideas and points of view be part of future developments. We hope our stories will inspire more women to pursue STEM careers.

Look for more stories from the Queens of Code and follow our journey on Facebook: https://www.facebook.com/QueensofCode/

Our website: https://Queensofcode.com

On Twitter: @QueensofCode

REFERENCES

1. Liza Mundy, *Code Girls: The Untold Story of the American Women Code Breakers of World War II*. New York, NY, USA: Hachette, 2017.
2. J. Abbate, "Women and gender in the history of computing," *IEEE Annu. History Comput.*, vol. 25, no. 4, pp. 4–8, Oct.–Dec. 2003.
3. Friedman is the subject of two recent books: G. S. Smith's, *A Life in Code: Pioneer Cryptanalyst Elizebeth Smith Friedman*. Jefferson, NC, USA: McFarland & Company, Inc., Publishers, 2017 and J. Fagone's, *The Woman who Smashed Codes*. New York, NY, USA: Dey Street, an imprint of William Morrow, 2017. Agnes Driscoll's mostly unknown life is the subject of K. W. Johnson's, *The Neglected Giant: Agnes Meyer Driscoll*. Ft. George G. Meade, Center for Cryptologic History, 2015. Another little-known early female cryptologist, Genevieve Young Hitt, is the subject of B. R. Smoot's, "An accidental cryptologist," *Cryptologia*, vol. 35, no. 2, pp. 164–175, 2011.
4. Hidden Figures: *The American Dream and the Untold Story of the Black Women Mathematicians Who Helped Win the Space Race*. New York, NY, USA: William Morrow, 2016.
5. NSA's Key Role in Major Developments in Computer Science, Part Two, Accessed: 2019. [Online]. Available: https://www.nsa.gov/Portals/70/documents/news-features/declassified-documents/nsa-early-computer-history/6586785-nsa-key-role-in-major-developments-in-computer-science.pdf
6. R. Predmore-Lynch. Accessed: 2019. [Online]. Available: https://www.nsa.gov/About-Us/Current-Leadership/Article-View/Article/1620951/renetta-predmore-lynch/
7. I was a co-creator of the popular Micro Adventure series (Scholastic 1984, with Ruth Glick). [Online]. Available: http://www.microadventure.net/Home/About/. We had a talented team of mostly women writers and programmers and some teens to create the series. 1984–1986
8. Dottie T. Blum, Hall of Honor Inductee. 2004. [Online]. Available: https://www.nsa.gov/About-Us/Current-Leadership/Article-View/Article/1622398/, Accessed: 2019.
9. Read more about flowcharting templates in the Peggy Aldrich Kidwell's article in the IEEE Annals of the History of Computing (vol. 41, no. 1). Accessed: 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8667955
10. Involutory matrices can be used in visual cryptography to transform the data. A more detailed example of the Hill cipher algorithm can be found at Accessed: 2020. [Online]. Available: https://pdfs.semanticscholar.org/9626/7d1194c8beffc8abcf8b142f9870051bdb7c.pdf
11. S. Snyder, Earliest Application of the Computer at NSA (Snyder), 1973.
12. Details on the history of Punched paper tape. Accessed: 2020. [Online]. Available: https://en.wikipedia.org/wiki/Punched_tape
13. Read more about IMP at: Accessed: 2020. [Online]. Available: http://www.liquisearch.com/imp_programming_language https://www.seas.harvard.edu/courses/cs152/2016sp/lectures/lec05-imp.pdf
14. Only 18% of computer science degrees were earned by women in 2016. Accessed: 2020. [Online]. Available https://www.computerscience.org/resources/women-in-computer-science/

Eileen Buckholtz is a Maryland-based computer scientist and author of 40 books. She received a M.S. degree in computer science from the University of Maryland, a B.S. in computer and information science from Ohio State University, and a Chief Information Officer certificate from the National Defense University. For over 30 years, she had an exciting and distinguished career working on cutting-edge technology for the National Security Agency.