Many years before I was born, in an interview he gave in 1995, Steve Jobs famously said: “Everybody should learn how to program a computer – should learn a computer language, because it teaches you how to think.” I could not agree with it more strongly: software engineering is, without doubt, the most exciting thing I know. Having discovered it at the age of 10 by learning Pascal at my school’s extracurricular programming club, within just a few months I had rapidly progressed from the obligatory “Hello World” to writing my own version of the 1970s classic arcade game, “Space Invaders”, complete with an adjustable shooting angle and other additional features.

As my school in Saint Petersburg specialised in Maths and Computer Science, I had ample opportunities to continue to develop and hone my abilities. At first, I focused on the challenge of learning the fundamentals of C; although my initial progress was slow and hard-won, I gradually came to terms with its syntactic rigour, the difference between pointers and references, and the intricacies of dynamic memory allocation. I continued to build on this foundation by studying object-oriented languages like C++ and Java, using them to explore new data structures and algorithms. I wrote an implementation of an AVL tree from scratch, solidifying my grasp of recursion in the process, then used my newfound understanding of binary trees in Huffman coding. Intrigued by lossless data compression techniques, I studied the LZ77 / LZ78 and DEFLATE algorithms, which in turn led to me investigating their lossy counterparts and becoming fascinated by the kaleidoscopic world of audio and video codecs. Keen to develop a well-rounded view of the modern software and hardware applications, I then began to study graphics rendering, using the OpenGL API to design and animate a 3D vehicle moving on an inclined slope.

Coming to the UK last year to continue my education had allowed me to focus on the subjects relevant to my future career choice. A-levels in Maths and Further Maths gave me a new perspective on the familiar concepts, such as using logarithms to calculate the orders of operations of binary trees algorithms or learning the relevance of probability distributions and statistical expectations to data compression. I also decided to teach myself the Decision Mathematics modules in my spare time, which gave me an insight into Graph Theory algorithms, such as Prim’s or Kruskal’s, and inspired me to write a version of them in Python, the language I am currently studying. As I researched the heuristic solutions to the Travelling Salesman problem, I learned more about the nature of optimisation and computational trade-offs between quality and speed. Finally, my last A-level subject of Economics offers me the opportunity to understand the nature of global markets, and I use my coding skills and knowledge of mathematical models to write dynamic simulations of these processes.

Like every great technology, computers also give their users unique ways to misuse them. Watching a documentary "The Social Dilemma" made me consider the downsides of an interconnected global society, which lets big tech companies to collect and trade personal data of their users, using it to shape public attitudes and opinions. To investigate this further, I read "Free Software, Free Society" by Richard M. Stallman, which introduced me to the world of FOSS, Free and open-source software. This became the topic of my EPQ, "Linux: an OS you control", in which I demonstrated how by meticulously decomposing every action undertaken in the OS, it becomes possible for a user to control the system, instead of the system controlling the user.

I always strive to grow my knowledge of coding and the algorithms involved therein. Solving challenging programming problems on Project Euler spurred me to create my own bespoke libraries for fundamental mathematical tasks, such as working with large numbers, precise algebraic operations and simplification of fractions, and modular arithmetic. "Behind Deep Blue", a book about the IBM chess supercomputer that defeated the world champion Garry Kasparov in 1997, encouraged me to enrol into a two-week summer course on machine learning at the University of Oxford, where I studied genetic algorithms and the fundamentals of constructing a deep neural network. To improve the quality of my code, I read books like "Clean Code" by Robert C. Martin and Andrew Hunt's "The Pragmatic Programmer," putting their suggestions into practice to make my work more concise, readable, maintainable, and efficient. Last summer, I began to research cryptocurrencies and the blockchain technology, which led me to complete a two-week internship at Euratek GmbH, a Swiss company providing digital currency investment services. Being exposed to the day-to-day operations allowed me to witness first-hand the challenges of dealing with the inherently volatile virtual currency markets, and the importance of effective software solutions to mitigate the unpredictability thereof.

I believe that physical and mental health are intertwined, highlighting the essential nature of keeping active no matter the academic workload at hand. I have been playing tennis for 7 years and have 4 years of MMA experience. I greatly enjoy volunteering, having recently coordinated a fundraiser, “Night Walk for the National Autistic Society”. I am also part of the Student Council at my school, where I have been appointed the digital innovator.