Alan Turing and the Turing test

John MacCormick

(appeared in Dickinson Science Magazine, volume 1 issue 2, p31)

If you're a fan of the British TV show *Sherlock*, you might be interested to hear that the lead actor in that show, Benedict Cumberbatch, will be starring as the legendary English scientist Alan Turing in a movie to be released this fall. The movie is called *The Imitation Game*. Why? We'll get back to that in a minute.

First, who was Alan Turing, and why would anyone make a movie about him? He's sometimes described as a mathematician, sometimes as an engineer, and sometimes as a computer scientist. All are true. His university degrees were in mathematics; he essentially founded the field of theoretical computer science with a breakthrough paper in the 1930s; and he later helped to engineer some of the earliest real computers. But for something worthy of the big screen, we can turn to Turing's contribution during the Second World War: he was a key member of the team that cracked the German naval Enigma code. It's generally accepted that his work saved many lives and considerably shortened the war. And it also makes for an exciting movie script, especially when you consider that the work was done on a shoestring budget by a secret team operating at an English country mansion called Bletchley Park. (Many Dickinson students have visited Bletchley Park as part of the Norwich science program.)

But where does the "imitation game" come into it? Well, it turns out that Turing didn't stop after his seminal work on the theory of computation and cryptography. In the 1950s, he also wrote one of the founding documents of artificial intelligence. In a paper called *Computing Machinery and Intelligence*, he gave a beautiful description of what it might mean for a machine to "think". At the heart of the description is the imitation game, which we now know as the "Turing test": a computer plays a game in which it tries to fool a human interrogator into thinking it's human.

The Turing test remains strikingly relevant today. I was reminded of this a couple of years ago when I was lucky enough to be in Cambridge, England for the centenary celebrations of Alan Turing's birth. Two of the talks I saw there focused on interesting variants of the Turing test: in one, we ask if a computer could imitate a mathematician; in the other, we ask if a computer could imitate a computer programmer. These are profound questions that take us far beyond the slippery chat bots that dominate attempts to pass the generic Turing test. Imagine, instead, if a computer could prove interesting new mathematical theorems, or write useful new computer programs. These are imitation games that could really change the world.