Administrative instructions: Please turn in sheets to Ugo Marzolino before the exercise session. A complete solution will be worth ten points (fewer on this first sheet, as there is less than one week to complete it). Solutions should be submitted in groups of three. Check the web site for occasional hints and updates.

[P1] We'll start softly. The proofs asked for in this problem can easily be found in standard textbooks or online. Your creativity will be needed later – here the goal is that everybody understands the fundamental concepts employed.

(1) Write out Cantor's *diagonal argument* for the fact that the set of real numbers is uncountable. The *power set* $\mathcal{P}(S)$ of a set S is the set of all subets of S. Prove that $\mathcal{P}(\mathbb{N})$ is uncountable.

(2) Recall that a *language* $\mathcal{L}$ is a subset of the set $\{0,1\}^*$ of binary strings (e.g. we have encountered the language of strings of odd parity). A Turing machine T *decides* the language $\mathcal{L}$ if (*i*) T takes bit strings as input, (*ii*) for every input x, T halts after finitely many steps, (*iii*) T will write "1" to a designated cell of its output tape if $x \in \mathcal{L}$, and it will write "0" to that cell if $x \notin \mathcal{L}$. Using just the results of (1), give a very short proof that there are Turing-undecidable languages.

(3) Problem (2) shows that there *exist* uncomputable problems, but it gives no hint as to what they might be. Let

$$\mathcal{L}_H = \{\langle \alpha, x\rangle \,|\, \text{TM with Turing number } \alpha \text{ halts on input of } x\}$$

be the language consisting of "halting combinations" of Turing machines α and an inputs x. Show that $\mathcal{L}_H$ is Turing-undecidable. (Hint: aim for a proof by contradiction. Show that if $\mathcal{L}_H$ were decidable, one could arrive at the paradoxical situation of being able to specify a program that halts exactly if it doesn't halt.) (6 P.)