TURING THEOREM

THE CONCEPT OF ALGORITHM

An algorithm is a finite sequence of mechanical, deterministic instructions based on a finite alphabet, i.e. a computer program.

We shall call a Turing machine a “perfect” computer, that is to say with infinite memory which can never break down.

We want to understand which tasks can be accomplished by a Turing machine, for example in arithmetics.

PRIME NUMBERS

It is possible, for example, to consider a Turing machine that provides us the list of all the prime numbers. In fact, deciding whether or not a natural number is a prime number is a matter of computing.

N=[0, 1, 2, 3 …] → Turing Mac. → [2, 3, 5, 7 …]

The actions of Turing machines can represent some problems of the arithmetic. For example the arithmetical problem:

"Is the set of prime numbers a finite or an infinite set?"

can be also expressed as:

"Will the Turing machine which generates the set of prime numbers ever stop?"

It can be proved that the answer to this question is no, as the set of the prime numbers is an infinite set.

This is a solved solvable problem.

PERFECT NUMBERS

A number is perfect if it is equal to the sum of its divisors, such as:

6 = 1+2+3 ;

28 = 1+2+4+7+14.

Even in this case, there exists an algorithm which can decide whether or not a natural number is perfect.

PERFECT NUMBERS

We can consider the following Turing machine:

N = [0, 1, 2, 3 …] → Turing Mac. → [6, 28, 496 …]

It hasn't been proved yet whether the set of perfect numbers is finite or infinite, so we do not know yet whether or not this Turing machine stops.

This is an unsolved solvable problem.

Let S be an increasing sequence of infinitely many natural numbers generated by an algorithm

TURING THEOREM:

There exists no algorithm whose output is the list of all these S.