Definition of complexity strings $\mathrm{K}(x)$:-

If *x* be binary string, then the minimal description and descriptive complexity of *x*'s are $d(x)$ and $\mathrm{K}(x)$ respectively. Turing machine *M* and small string *w* we get minimal description is $\langle M, w\rangle$. From several of such shorter strings we select lexicographically among them then we can get descriptive complexity of such strings $\mathrm{K}(x) = |d(x)|$.

Now we must prove that $\mathrm{K}(x)$ is not a computable function.

Let y_n be the lexicographical first-string s that satisfies $n < K(y)$. Let M be the Turing machine such that if the input is n, the binary strings that are generated are $x_0, x_1, x_2, x_3 \ldots$.

The Turing machine M computes $\mathrm{K}(x_i)$ and if $\mathrm{K}(x) > n$, then the Turing Machine will generate x_i as the output and the machine will halt.

If the machine does not halt, the machine will examine the next lexicographical string x_{i+1}.

The Turing machine M will come across a string x which satisfies $\mathrm{K}(x) > n$ because K is unbounded.

For input n, the output generated is y_n but the length of the input n is $\log_2(n)$. So, $K(n) < \log_2(n) + c$, where c is the constant.

So, for all n, $n < K(y_n)$, so it can be said that $n < \log_2(n) + c$ $n < \log_2(n) + c$ but this is false, if n is large.

This is a contradiction because M cannot compute K(x) so, it is not a computable function.