Steps:

1. ask user for DNA sequence [tip: use function *input()*]
2. Test if the DNA sequence is RNA [tip: use conditional expression to test for the presence of "U"]
3. Test if the DNA sequence contains non-DNA characters [tip: you could use a loop through all letters in DNA]
4. Compute and print nucleotide percentages [tip: you could use a loop through all four nucleotides and then use *count()* function]
5. Generate a dictionary of dinucleotides [tip use a loop]
6. Move along the sequence one nucleotide at a time to extract all overlapping 2-base subsequences and count dinucleotides [tip: to extract 2bp-subsequences use the notation "DNA[pos_i:pos_i+1]"]
7. Define dictionary with restriction enzymes [tip: see class slides #17]
8. Search sequence with *find* to locate restriction sites [tip:use find and the position argument]

Algorithm 1 Assigment 2 pseudocode

```
DNA ← userinput
if "U" in DNA then
    Print Warning
    Replace "U" by "T"
end if
for each letter in DNA do
    if letter not in DNA then
        Print Errormessage
        Exit
    end if
end for
for each nucleotide in "A","C","G","T" do
    NuclPercent ← Count nucleotide in DNA/length DNA
    Print NuclPerecnt
end for
for each nucl1 in "A","C","G","T" do
    for each nucl2 in "A","C","G","T" do
        DiNuc[nucl1 + nucl2] ← 0
    end for
end for
for each position in 1 to length(DNA)-1 do
    DiNuc[position : position + 2] ← DiNuc[position : position + 2] + 1
end for
for each DiNuc.key in key(DiNuc) do
    Print DiNuc[Dinuc.key]/(length(DNA)-1)
end for
REdic ← {"EcoRI" : "GAATTC","BamHI" : "GGATCC","HindIII" :
"AAGCTT","NotI" : "GCGGCCGC"}
for each Enz in key(REdic) do
    EnzPos ← 0
    if REdic[Enz] in DNA then
        while not(End.of.Seq) do
            CutPos ← Find REdic[Enz] in DNA after EnzPos
            Print CutPos
            if CutPos<0 then
                End.of.Seq==True
            end if
        end while
    end if
end for
```