The DNA molecule consists of a long sequence of four nucleotide bases: adenine (A), cytosine (C), guanine (G) and thymine (T). Since this molecule contains all the genetic information of a living organism, geneticists are interested in understanding the roles of the variuos DNA sequence patterns that are continuously being discovered worldwide. One of the most common methods to identify the role of a DNA sequence is to compare it with other DNA sequences, whose functionality is already known. The more similar such DNA sequences are, the more likely it is that they will function similarly.

Your task is to write a C program, called `dna.c`, that reads three DNA sequences from a file called `dna_input.dat` and prints the results of a comparison between each pair of sequences to the file `dna_output.dat`. The input file `dna_input.dat` consists of three lines. Each line is a single sequence of characters from the set {**A**, **C**, **G**, **T**}, that appear without spaces in some order, terminated by the end of line character **\n**. You *can assume* that the three lines contain the same number of characters, and that this number is at most 241 (including the character **\n**). Here is a sample input file:

```
ACGTTTTAAGGGCTGAGCTAGTCAGTTCATCGCGCGCGTATATCCTCGATCGATCATTCT
CTCTAGACGTTTTAAGGGCTGAGCTAGTCAGTTC
ACGTTTTAAGGGCTTAGAGCTTATGCTAATCGCGCGCGTATATCCTCGATCGATCATTCT
CTCTAGACGTTTTAAGGGCTAAGGCGCGTAATTA
TCGTTTGAAGGGCTTAGTTAGTTAGTTCATCGGCGGCGTATATCCTCGATCGATCATTCT
CTCTAGACGTTTTAAGGGCTGAGCCGGTCAGTTA
```

Each of the three lines (shown with wrap-around above) consists of 95 characters: the 94 letters from {**A**, **C**, **G**, **T**} and the character **\n** (not shown). The output file `dna_output.dat` must be structured as follows. For each pair of sequences #i and #j, with $i, j \in \{1, 2, 3\}$ and $i > j$, you should print:

- A single line, saying "**`Comparison between sequence #i and sequence #j:`**"
- The entire sequence #i in the *first row*, and the entire sequence #j in the *third row*.
- The comparison between the two sequences in the *second (middle) row*. This should be printed as follows. For each position, if the two bases are the *same* in both sequences then the corresponding base letter (one of **A**, **C**, **G**, **T**) should be printed; otherwise a blank **" "** should be printed.
- A single line, saying "**`The overlap percentage is x%`**" where x is a floating-point number which measures the percentage of letters that match in the two sequences. This number should be printed with a single digit of precision after the decimal point.

Each line in the output file `dna_output.dat` should contain at most 61 characters, including the end of line character **\n**. If the DNA sequences are longer than that, then each of the three rows mentioned

above should be split across several lines, with the first few lines containing *exactly* 60 letters, and the last containing the rest of the letters. Here is a sample file `dna_output.dat` which results upon processing the file `dna_input.dat` above:

```
Comparison between sequence #1 and sequence #2:
ACGTTTTAAGGGCTGAGCTAGTCAGTTCATCGCGCGCGTATATCCTCGATCGATCATTCT
ACGTTTTAAGGGCT AG    T  G T ATCGCGCGCGTATATCCTCGATCGATCATTCT
ACGTTTTAAGGGCTTAGAGCTTATGCTAATCGCGCGCGTATATCCTCGATCGATCATTCT

CTCTAGACGTTTTAAGGGCTGAGCTAGTCAGTTC
CTCTAGACGTTTTAAGGGCT AG      A TT
CTCTAGACGTTTTAAGGGCTAAGGCGCGTAATTA

The overlap percentage is 80.9%


Comparison between sequence #1 and sequence #3:
ACGTTTTAAGGGCTGAGCTAGTCAGTTCATCGCGCGCGTATATCCTCGATCGATCATTCT
 CGTTT AAGGGCT AG TAGT AGTTCATCG   GCGTATATCCTCGATCGATCATTCT
TCGTTTGAAGGGCTTAGTTAGTTAGTTCATCGGCGGCGTATATCCTCGATCGATCATTCT

CTCTAGACGTTTTAAGGGCTGAGCTAGTCAGTTC
CTCTAGACGTTTTAAGGGCTGAGC  GTCAGTT
CTCTAGACGTTTTAAGGGCTGAGCCGGTCAGTTA

The overlap percentage is 88.3%


Comparison between sequence #2 and sequence #3:
ACGTTTTAAGGGCTTAGAGCTTATGCTAATCGCGCGCGTATATCCTCGATCGATCATTCT
 CGTTT AAGGGCTTAG    T  G T ATCG   GCGTATATCCTCGATCGATCATTCT
TCGTTTGAAGGGCTTAGTTAGTTAGTTCATCGGCGGCGTATATCCTCGATCGATCATTCT

CTCTAGACGTTTTAAGGGCTAAGGCGCGTAATTA
CTCTAGACGTTTTAAGGGCT AG CG   A TTA
CTCTAGACGTTTTAAGGGCTGAGCCGGTCAGTTA

The overlap percentage is 79.8%
```

Notes:

- As part of the solution, you are required to declare, define, and call the following functions. In these functions, you can assume that **`input`** and **`output`** are *global variables* of type **`FILE *`**.
  - The function **`read_DNA(char sequence[])`** that reads a DNA sequence from **`input`**, stores it in the array **`sequence[]`**, and returns the number of letters read, as an **`int`**.
  - The function **`compare_DNA(char seq1[], char seq2[], char seq3[], int n)`** that stores in the array **`seq3[]`** the comparison sequence of the two DNA sequences stored in **`seq1[]`** and **`seq2[]`**. The length of these DNA sequences is assumed to be **`n`**. The function returns, as a **`double`**, the percentage of overlap between the two DNA sequences.
  - The function **`print_DNA(char seq1[], char seq2[], char seq3[], int n)`** that prints to **`output`** the DNA sequences stored in **`seq1[]`** and **`seq2[]`**, as well as their comparison sequence stored in **`seq3[]`**, according to the rules explained above. The length of all these sequences is assumed to be **`n`**. The function does not return a value.
- The numbers 241 and 60, used above, should be defined as symbolic constants **`MAX_IN_LENGTH`** and **`OUT_LENGTH`**, using the **`#define`** compiler directive. The program should keep working correctly if the values of these symbolic constants are changed (within a reasonable range).