Spring 2025, MIS 102 – COMPUTER PROGRAMMING

Midterm Exam

姓名:________________ 學號:________________ 系級:________________

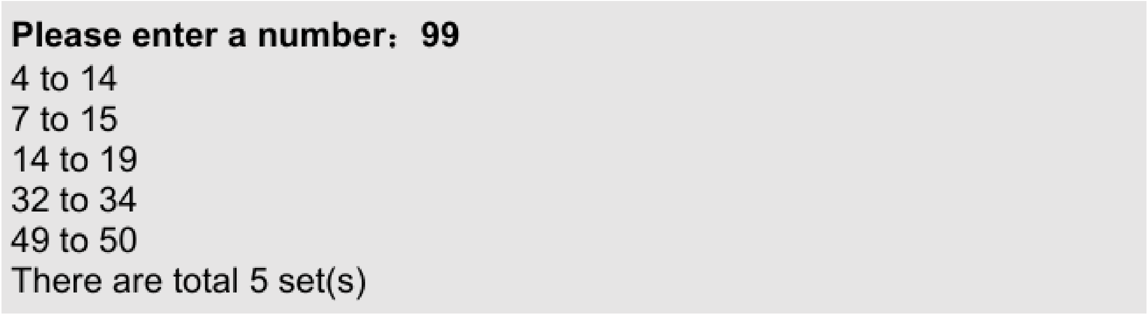
Part ＩＩ. [40 pts]

1. **[10 pts]** An integer number n may be represented by sums of a set (or multiple sets) of consecutive numbers. Please write a C program that prints and counts these set(s). For example, if we enter “9”, the output of the program is "2”, as there are 2 sets:

   2+3+4

   4+5

   Below is a sample output of your program.



```
Please enter a number：99
4 to 14
7 to 15
14 to 19
32 to 34
49 to 50
There are total 5 set(s)
```

2. [10 pts] Print All Numbers Yielding Reversible Sum Palindromes

A number n is said to yield a **reversible sum palindrome** if, when added to its digit-reversal, the resulting sum is a palindrome. For example:

- For i=12, its reversal is 21 and 12+21=33 (33 is a palindrome), so 12 is valid.
- For i=107, its reversal is 701 and 107+701=808 (808 is a palindrome), so 107 is valid.

Notes:

1. Only consider numbers with two or more digits (i.e., i ≥ 10).

2. If the number itself is a palindrome (e.g., 11, 22, etc.), skip it.

3. To avoid duplicate mirrored pairs (e.g., 12 and 21 yielding the same result), only consider a number if i<reverse(i).

4. There is no restriction on the number of digits in the sum; as long as i+reverse(i) is a palindrome, it is accepted.

Your Task:

Write a C program that:

1. **Input:**
   - Prompts the user to enter a positive integer n (this n is the upper bound for the search; assume n≥10).

2. **Processing:**
   - Iterates through each number i from 10 to n (inclusive).
   - For each i, calculates its digit reversal.
   - If the following conditions are met, print the equation:
     (a) i is not a palindrome;
     (b) i<reverse(i);
     (c) i+reverse(i) is a palindrome.
   - Print every valid equation in the given range.

3. **Output:**
   - For each valid iii, print the equation in the format:
     `i + reversal(i) = sum`
   - Finally, print the total count of numbers that satisfy the condition.

Sample Execution:

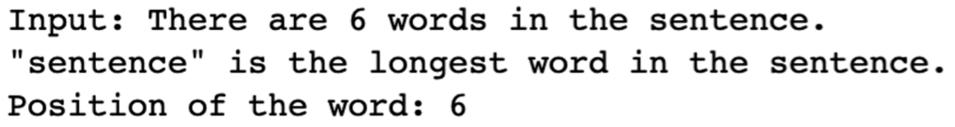
```
Please enter a positive integer: 20

Numbers yielding reversible sum palindromes:

12 + 21 = 33

13 + 31 = 44

14 + 41 = 55

15 + 51 = 66

16 + 61 = 77

17 + 71 = 88

18 + 81 = 99

Total count: 7
```

3.[10 pts] Please write a C program with a function int longestWord(char *sentence) which return the position of the longest word in a sentence. The program should print out both of the longest word and its position in the sentence.

Hints:

1. You can use this sentence “There are 6 words in the sentence” as the input of the function.
2. You can predefine a number to limit the length of the longest word.
3. If there are two or more words in the same length, please output the first one.

Expected output would be looked like:



```
Input: There are 6 words in the sentence.
"sentence" is the longest word in the sentence.
Position of the word: 6
```

4. [10 pts] Check Divisibility After Truncating the Leftmost Digit

Problem Description: Define a number as a "truncated divisible number" if it satisfies the following conditions:

1. The number has at least three digits (i.e., i≥100).

2. Remove (truncate) the leftmost digit of the number to obtain a new number (ignoring any leading zeros).

3. If the new number is not zero and divides the original number evenly (i.e., i%(truncated)==0), then the number meets the condition.

For example:

- For i=105, removing the leftmost digit yields 05 (which is 5). Since 105%5==0, 105 meets the condition and should be printed as:

  105 / 5 = 21

- For i=120, removing the leftmost digit yields 20. Since 120%20==0, 120 meets the condition, and you should print:

120 / 20 = 6

- For i=100, removing the leftmost digit yields 00 (considered 0); since division by 0 is invalid, skip this number.

Your Task:

Write a C program that:

1. **Input:**
    - Prompt the user to enter a positive integer N (this N is the upper bound for the search; assume N≥100).
2. **Processing:**
    - For each number i in the range 100 ≤ i ≤ N :
        - Determine the number of digits in i and compute the divisor corresponding to $10^{(number\ of\ digits-1)}$.
        - Use this divisor to remove the leftmost digit from iii (i.e., compute truncated=i%divisor).
        - If the truncated number is not zero and divides the original number evenly (i.e., i%truncated==0), record the number and its quotient.
3. **Output:**
    - For each number that meets the condition, print the equation in the format:

      i / truncated = quotient
    - Finally, print the total count of numbers that satisfy the condition.

    - If no number in the range can be expressed in this way, print an appropriate message.

Sample Execution:

```
Please enter a positive integer: 110

The numbers that satisfy the truncated divisibility condition are:

101 / 1 = 101

102 / 2 = 51

104 / 4 = 26

105 / 5 = 21

110 / 10 = 11

Total count: 5
```

Note: For example, for 105, the leftmost digit (1) is removed, leaving 05 (which is 5). Since 105 is divisible by 5, the equation is printed.