8. (2

7. Let A be an array containing positive integers. The following C program computes the length of the longest increasing sequence of numbers in A. For example, for

```
A = {1, 9, 2, 5, 8, 6, 4}
```

the increasing sequences of length 1 are {1}, {9}, {2}, {5}, {8}, {6}, {4}; of length 2 are {1, 9}, {2, 5}, {5, 8}; and of length 3 are {2, 5, 8}. There is no increasing sequence of length 4 or higher. So the length of the longest increasing sequence is 3. Hence the result is 3.

Complete the program by filling up the missing lines in the code. [5 * 1 = 5]

```
#include <stdio.h>

int main() {
    int A[] = { 2, 3, 5, 2, 3, 4, 7, 9, 1, 3, 2, 5, 7, 9 }; // Input array
    const int n = 14;      // Number of elements in the array
    int i = 0;             // Index variable to iterate over the array
    int len = 1;           // Length of the current increasing sequence
    int maxlen = 1;        // Length of the longest increasing sequence so far

    while (                ) {    // Iterate on array A. Note that the last
           ---------------        // element does not have a next element

        if (               )      // Check if the sequence is increasing
            ---------------

                                  // Update current increasing
                                  // sequence length
            ---------------

        else {
            if (len > maxlen)     // Update length of the
                maxlen = len;     // longest sequence so far

                                  // Re-init (Reset) current
                                  // increasing sequence length
            ---------------
        }

                                  // Move to next element

        ---------------
    }
    if (len > maxlen)      // Update length of the
        maxlen = len;      // longest sequence so far

    printf("Length of longest increasing sequence = %d\n", maxlen);

    return 0;
}
```