TEST CASES

Inputs:

```
3,10,7,9,4,11
2,5,14,6,8,16
10,5,48,3,89,90
1,6,2,3,7,4,5
7,15,26,35,85,3
6,5,3,4,2,1,100
```

Outputs:

```
Longest:  3 7 9 11        size =  4
Longest:  2 5 6 8 16      size =  5
Longest:  10 48 89 90     size =  4
Longest:  1 2 3 4 5       size =  5
Longest:  7 15 26 35 85   size =  5
Longest:  3 4 100         size =  3
```

Inner Results:

```
3 10 11       size =  3
3 10       size =  2
3 7 9 11       size =  4
3 7 9       size =  3
3 7 11       size =  3
3 7       size =  2
3 9 11       size =  3
3 9       size =  2
3 4 11       size =  3
3 4       size =  2
3 11       size =  2
3       size =  1
10 11       size =  2
10       size =  1
7 9 11       size =  3
7 9       size =  2
7 11       size =  2
7       size =  1
9 11       size =  2
9       size =  1
4 11       size =  2
4       size =  1
11       size =  1
0       size =  0

Longest:  3 7 9 11       size =  4
```

```
1 6 7       size =  3
1 6       size =  2
1 2 3 7       size =  4
1 2 3 4 5       size =  5
1 2 3 4       size =  4
1 2 3 5       size =  4
1 2 3       size =  3
1 2 7       size =  3
1 2 4 5       size =  4
1 2 4       size =  3
1 2 5       size =  3
1 2       size =  2
1 3 7       size =  3
1 3 4 5       size =  4
1 3 4       size =  3
1 3 5       size =  3
1 3       size =  2
1 7       size =  2
1 4 5       size =  3
1 4       size =  2
1 5       size =  2
1       size =  1
6 7       size =  2
6       size =  1
```

```
2 3 7       size =  3
2 3 4 5       size =  4
2 3 4       size =  3
2 3 5       size =  3
2 3       size =  2
2 7       size =  2
2 4 5       size =  3
2 4       size =  2
2 5       size =  2
2       size =  1
3 7       size =  2
3 4 5       size =  3
3 4       size =  2
3 5       size =  2
3       size =  1
7       size =  1
4 5       size =  2
4       size =  1
5       size =  1
0       size =  0

Longest:  1 2 3 4 5       size =  5
```

```
2 5 14 16       size =  4
2 5 14       size =  3
2 5 6 8 16       size =  5
2 5 6 8       size =  4
2 5 6 16       size =  4
2 5 6       size =  3
2 5 8 16       size =  4
2 5 8       size =  3
2 5 16       size =  3
2 5       size =  2
2 14 16       size =  3
2 14       size =  2
2 6 8 16       size =  4
2 6 8       size =  3
2 6 16       size =  3
2 6       size =  2
2 8 16       size =  3
2 8       size =  2
2 16       size =  2
2       size =  1
5 14 16       size =  3
5 14       size =  2
5 6 8 16       size =  4
5 6 8       size =  3
5 6 16       size =  3
5 6       size =  2
5 8 16       size =  3
5 8       size =  2
5 16       size =  2
5       size =  1
14 16       size =  2
14       size =  1
6 8 16       size =  3
6 8       size =  2
6 16       size =  2
6       size =  1
8 16       size =  2
8       size =  1
16       size =  1
0       size =  0

Longest:  2 5 6 8 16       size =  5
```

```
7 15 26 35 85       size =  5
7 15 26 35       size =  4
7 15 26 85       size =  4
7 15 26       size =  3
7 15 35 85       size =  4
7 15 35       size =  3
7 15 85       size =  3
7 15       size =  2
7 26 35 85       size =  4
7 26 35       size =  3
7 26 85       size =  3
7 26       size =  2
7 35 85       size =  3
7 35       size =  2
7 85       size =  2
7       size =  1
15 26 35 85       size =  4
15 26 35       size =  3
15 26 85       size =  3
15 26       size =  2
15 35 85       size =  3
15 35       size =  2
15 85       size =  2
15       size =  1
26 35 85       size =  3
26 35       size =  2
26 85       size =  2
26       size =  1
35 85       size =  2
35       size =  1
85       size =  1
3       size =  1
0       size =  0

Longest:  7 15 26 35 85       size =  5
```

```
10 48 89 90       size =  4
10 48 89       size =  3
10 48 90       size =  3
10 48       size =  2
10 89 90       size =  3
10 89       size =  2
10 90       size =  2
10       size =  1
5 48 89 90       size =  4
5 48 89       size =  3
5 48 90       size =  3
5 48       size =  2
5 89 90       size =  3
5 89       size =  2
5 90       size =  2
5       size =  1
48 89 90       size =  3
48 89       size =  2
48 90       size =  2
48       size =  1
3 89 90       size =  3
3 89       size =  2
3 90       size =  2
3       size =  1
89 90       size =  2
89       size =  1
90       size =  1
0       size =  0

Longest:  10 48 89 90       size =  4
```

```
6 100       size =  2
6       size =  1
5 100       size =  2
5       size =  1
3 4 100       size =  3
3 4       size =  2
3 100       size =  2
3       size =  1
4 100       size =  2
4       size =  1
2 100       size =  2
2       size =  1
1 100       size =  2
1       size =  1
100       size =  1
0       size =  0

Longest:  3 4 100       size =  3
```

Pseudocodes:

```
Main:
        Open input file
        Open output file
        for n=0 to 6 do
                Read 1 line from input file
                Convert char to int and copy to array
                Find longest subsequence
                Print longest subsequence to console
                Convert int to char and copy to buffer
                Print buffer to output file
        End for
        Close input file
        Close output file
        End Program
End Main

FindLongestSub:
        if array index >= array size
                if temp array size >= max size
                        max size= temp array size
                        Copy temp to longest
                end if
                Print temp to console
        end if

        if temp size ==0 || current element > last element of temp array
                Add current element to temp array
                FindLongestSub( increase array index )
                Remove last element of temp array
        end if
        FindLongestSub( increase array index )
End FindLongestSub

IntToCharConverter:
        for i=0 to array size do
                divide array[i] by 10
                get reminder and quotient
                add reminder 48
                add quotient 48
                copy quotient and reminder to buffer
                add space to buffer
        end for
End IntToCharConverter

CharToIntConverter:
        temp=0
        for i=0 to buffer size do
                if buffer[i] != ‘,’
                        multiply temp by 10
                        substract 48 from buffer[i]
                        temp = temp + buffer[i]
                else
                        copy temp to array
                end if

                if buffer[i] != ‘\n’
                        return
                end if
        end for
End CharToIntConverter
```

Explanation Of Searching Algoritm:

While the program proceeds by checking whether it is greater than the last element of the temp array from the first element of the array, it adds it to the temp array and goes to the last element.

When it comes to the end of the array, it removes the last element of the temp array and performs the operations again. When the subarray starting with the first element of the array is finished, it finds the subarrays starting with the second element of the array.

Explanation Of Reading File:

When reading character by character, it checks whether it is less than 48.

If it is greater than 48, it subtracts 48 and adds it to the sequence according to the next character, or subtracts 48 from the character it reads and multiplies the number it keeps in memory by 10 and then adds it with that number.

If it is less than 48, the character it reads is a comma or new line character. Adds the number it keeps in memory to the array.

Explanation Of Writing File:

It first checks if the element in the array is less than 10.

If it is less than 10, it adds 48 and saves it to the buffer.

If it is greater than 10, it divides by 10. If the section is greater than 10, it divides 10 again, first adds 48 to the section and saves it to the buffer, then adds 48 to the remaining section and saves it to the buffer.

Time Complexity Of Searching Algoritm:

```
FindLongestSub:
        if array index >= array size                                        O(2*k)
                if temp array size >= max size                              O(k)
                        max size= temp array size                           O(1)
                        Copy temp to longest            S(k)                O(k) k elements in temp
                end if
                Print temp to console                                       O(k) k elements in temp
        end if

        if temp size ==0 || current element > last element of temp array    T(n-1)
                Add current element to temp array                           Θ(1)
                FindLongestSub( increase array index )                      T(n-1)
                Remove last element of temp array                           Θ(1)
        end if
        FindLongestSub( increase array index )                              T(n-1)
End FindLongestSub
```

$T(n)=2*T(n-1)+O(2*k)$

$T(n-1)=2*T(n-2)+O(2*(k-1))$ Space Complexity : S(k)

.

\+ $T(1)=2*T(0)+O(2)$

$T(n)=2^n+O(2*k)$