This condition is true if either
\( i > n \), or \( \text{found} = \text{true} \), or both.

A variation on the **sequential search**
algorithm looks for a given number
(\( \text{key} \)) from a list \( a_1, \ldots, a_n \) of
numbers. If a match is found, the
index \( i \) for which \( a_i = \text{key} \) is
returned, otherwise \( 0 \) is returned.

**Given:** \( n \geq 1, a_1, \ldots, a_n, \text{key} \)

**Find:** The first index \( i \) such that
\( a_i = \text{key} \). If no match, return \( 0 \).

1.) \text{set } i \text{ to } 1
2.) \text{set } \text{found} \text{ to } \text{false}
3.) \text{repeat until } i > n \text{ or } \text{found}
4.) \text{if } a_i = \text{key}
5.) \text{set } \text{found} \text{ to } \text{true}
6.) \text{else}
7.) \text{set } i \text{ to } i + 1
8.) \text{end of loop}
9.) \text{if } \text{found} \text{ is not found}
10.) \text{set } i \text{ to } 0
11.) \text{print } i
12.) \text{stop}

Ex. trace with \((3, -1, 2, 5, 12)\), \( n = 5 \), \( \text{key} = 2 \)
How can we modify either version to find the 2nd (or i-th) occurrence of key? To find every occurrence?

Another common problem is to find the largest element in a list of numbers.

Find-largest Algorithm

Given: n ≥ 2, and n distinct numbers a₁, ..., aₙ
Find: the largest value in the list and its location (i.e., the index where it occurs.)

1.) Set MAX to a₁
2.) Set LOCATION to 1
3.) Set i to 2
4.) Repeat until i > n
5.) If aᵢ > MAX
6.) Set MAX to aᵢ
7.) Set LOCATION to i
8.) Set i to i + 1
9.) End loop
10.) Print MAX
11.) Print LOCATION
12.) STOP
EX. BRIEFLY TRACE EXECUTION ON
(2, 1, -7, 5, 5, 11, 0)

How does this algorithm behave when given a list of non distinct numbers?
E.g. (1, 3, 0, 10, 3, 5, 10)

How can the algorithm be modified to return the number of times the minimum occurs?
(Hint: Combine with sequential search.)

How can we modify it to find the minimum?

Another classical problem in computer science is pattern matching. There are many versions of this problem, dealing with patterns in graphics, sound, pictures, and other types of data.

We will consider a simple form of this problem: finding patterns in text data.
More precisely, given \( n \) characters of text

\[ T_1, T_2, \ldots, T_n \]

and given a pattern of \( m \) characters, where \( m \leq n \)

\[ P_1, P_2, \ldots, P_m \]

find every occurrence of the pattern within the text, i.e., find each index position in the text at which the pattern begins.

\[ \text{Ex. } n = 18, \ m = 2 \]

\[ \text{TEXT: } \ TO\_RE\_OR\_NOT\_TO\_BE \]

\[ \text{PATTERN: } \ BE \]

\[ \text{Answer: } \ 4, 17 \]

\[ \text{Ex} \]

\[ \text{TEXT: } \ xxxaaaaxx \]

\[ \text{PATTERN: } \ aaa \]

\[ \text{Answer: } \ 4, 5, 6 \]
Notice that if \( m = 1 \), i.e. the pattern was just one character, then this problem is not essentially different from searching a list. We may expect that our algorithm will show some similarity to sequential search.

Ex. \( n = 7 \), \( m = 3 \)

- \( T_0 \, T_1 \, T_2 \, T_3 \, T_4 \, T_5 \, T_6 \, T_7 \)
  - \( l = 1 \)
  - \( P_1 \, P_2 \, P_3 \)

\( \rightarrow \) Slide over

- \( T_1 \, T_2 \, T_3 \, T_4 \, T_5 \, T_6 \, T_7 \)
  - \( l = 2 \)
  - \( P_1 \, P_2 \, P_3 \)

- \( T_1 \, T_2 \, T_3 \, T_4 \, T_5 \, T_6 \, T_7 \)
  - \( l = 3 \)
  - \( P_1 \, P_2 \, P_3 \)

- \( T_1 \, T_2 \, T_3 \, T_4 \, T_5 \, T_6 \, T_7 \)
  - \( l = 4 \)
  - \( P_1 \, P_2 \, P_3 \)

- \( T_1 \, T_2 \, T_3 \, T_4 \, T_5 \, T_6 \, T_7 \)
  - \( l = 5 \)
  - \( P_1 \, P_2 \, P_3 \)

Possible answers are indices \( l \) ranging from 1 to \( n - m + 1 \).
PATTERN MATCHING ALGORITHM

Given: $n \geq 1$, $m \geq 1$, $m \leq n$

$T_1T_2\ldots T_n$, $P_1P_2\ldots P_m$

Find: All indices $i$ such that $P_1\ldots P_m$

MATCHES $T_i\ldots T_{i+m-1}$

1.) Set $i$ to 1
2.) REPEAT UNTIL $i > n - m + 1$
3.) Set $i$ to 1
4.) Set MATCH to TRUE
5.) REPEAT UNTIL $i > m$ OR MATCH = FALSE
6.) IF $P_j \neq T_{i+j-1}$
7.) Set MATCH to FALSE
8.) ELSE
9.) Set $i$ to $i+1$
10.) END LOOP
11.) IF MATCH = TRUE
12.) PRINT 'MATCH FOUND AT POSITION:
13.) PRINT $i$
14.) Set $i$ to $i+1$
15.) END LOOP
16.) STOP
Ex.
Hand the band to Andy and I

Search for: and, and, and, and, and, and, and, and, and, and, and,

HW 2 Due 10/1 Wed 10/2 Thu 10/4
Chap. 2 p. 61
# 1, 6, 9, 12, 13, 15, 17, 18.