San José State University
Department of Computer Engineering

CMPE 152
Compiler Design
Section 1
Spring 2021
Instructor: Ron Mak

Assignment #1

Assigned: Thursday, January 28
Due: Thursday, February 4 at 2:30 PM
Individual assignment, 100 points max

Write simple Pascal programs
The purpose of this assignment is to give you a crash course in Pascal programming.

Use one of the following online Pascal development environments (I like the third one):

- http://rextester.com/l/pascal_online_compiler
- https://www.tutorialspoint.com/compile_pascal_online.php
- https://www.jdoodle.com/execute-pascal-online

Program 1: Go, Spartans!
Write a Pascal program that prints "Go, Spartans!" ten times to the standard output
(such as to your laptop screen).

Program 2: Merge two arrays

Write a Pascal program that does the following:

1. Read the following two lines of integers from the standard input (such as your keyboard) into two integer arrays:

   ```
   1 5 9 12 16 21 25
   3 7 12 13 14 15 16 19 20 26
   ```

   The numbers are separated by one or more blanks. Keep reading integers until the end of the line. You may assume that there are no errors in the input data and each line of integers is properly sorted in ascending order. Each array should be large enough to hold up to 20 values.

2. Pass each array by reference to a **Print** routine that prints the array values.

3. Pass the two arrays and a third initially empty array <u>by reference</u> to a **Merge** routine. The routine merges the two arrays and stores the merged values (which should be sorted) into the third array. The third array should be large enough to hold up to 40 values.

4. Pass the merged array to the **Print** routine.

Program 3: President records

Input file **presidents.txt** contains records in the form of text lines, one per U.S. president. Each president record contains:

- The starting year of the president's term (4 digits)
- The ending year of the president's term (4 digits)
- The president's first name
- The president's middle initials with periods (not all presidents have middle initials)
- The president's last name

One or more blanks separate the years and the first and last names and middle initials. There are no blanks inside the first and last names. If a president has more than one initial, they are written together with periods but no blanks.

You must assume that the presidents not sorted. Your program should not know in advance how many president records there are. The last president record is followed by a sentinel record that has -1 in place of the starting year.

Read and store each president's years, first name, middle initials (if any), and last name in a <u>Pascal record</u>. Store all the records in an <u>array of records</u> that's large enough to hold 20 records.

Following the dummy record, there is a list of search years terminated by -1. For each search year, find the president who was in office during that year (there may be more than one president).

Input file `presidents.txt`

```
2009 2017 Barack Obama
1961 1963 John Kennedy
1981 1989 Ronald Reagan
1993 2001 Bill Clinton
1989 1993 George H.W. Bush
1977 1981 Jimmy Carter
2017 2021 Donald Trump
1963 1969 Lyndon Johnson
1969 1974 Richard Nixon
1974 1977 Gerald Ford
2001 2009 George W. Bush
-1
1975 1963 1989 2019 2001 2030 2016 -1
```

Output

Read the input file and store the presidents' data into the array of records, sort the array into ascending order by starting year, and print the presidents in sorted order.

Then for each search year, search the array for the matching presidents and print the search year and the matching presidents' names. If there is no match, print "no match".

Text input hints

- If `ch` is a character variable, then `read(ch)` will read the next character from the standard input.

- If `n` is an integer variable, then `read(n)` will read the next integer value from the standard input.

- A call to `readln` will skip the rest of the current input line. Then the next call to `read` will read starting on the next input line.

- Boolean functions `eoln` and `eof` without parameters test whether the standard input is at the end of the current line or at the end of file, respectively. (If you're typing input data into a Windows command window, a line containing only control-Z is the end-of-file marker. On Linux and the Mac, it's control-D.)

- **Use Pascal's built-in string type.** Read online how to use the string type:

    - https://wiki.freepascal.org/String
    - https://www.freepascal.org/docs-html/rtl/sysutils/stringfunctions.html

    To use Pascal string functions, you may need to add the line

    Uses sysutils;

    at the beginning of your program.

What to submit to Canvas

- Your reasonably commented Pascal source files copied into text files.
- Text files of your programs' outputs.

Zip the files together and submit the zip file into Canvas: **Assignment #1**

> **This is an <u>individual assignment</u>. Your program must be your <u>own</u> work.**

Rubric
Your submission will be graded according to these criteria:

Criteria	Max points
Good output	**30**
Good program design	**40**
• Procedures and/or functions.	• 10
• Arrays.	• 10
• Record data type.	• 10
• Parameters passed by reference.	• 10
Good program style	**30**
• Descriptive variable names and meaningful comments.	• 10
• Good use of control statements.	• 10
• In general, well written.	• 10