

San José State University
Department of Computer Engineering

CMPE 152 Compiler Design

Section 1
Fall 2019
Instructor: Ron Mak

Assignment #1

Assigned: Tuesday, August 26
Due: Tuesday, September 3 at 11:59 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:

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

Program 1: Hello, world

Write a Pascal program that prints “Hello, world” 10 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 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 by reference to a **Merge** routine. The routine merges the two arrays and stores the merged values (which should be sorted) into the third array.
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)
- The president's last name

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

You may assume that the presidents are sorted by year. But you do not know in advance how many president records there are. The last president record is followed by a dummy 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 Pascal record. Store all the records in an array of records 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`

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

Output

After reading in the presidents and storing them in an array of records, print the array.

Then for each search year, search the array for the matching presidents and print the search year and the 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 **eofln** 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.)
- **Do not use the built-in string type.** Represent a string as a "packed" array of characters, as in classic Pascal. For example:

TYPE

```
    Name = PACKED ARRAY [1..12] OF char;
```

A string value can only be read one character at a time from the input and entered into the character array. However, an entire string (the entire array) can be written all at once. After reading a name, blank-fill the rest of the character array.

What to submit to Canvas

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

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

This is an individual assignment. Your program must be your own work.

Rubric

Your program will be graded according to these criteria:

Criteria	Max points
Good output	30
Good program design <ul style="list-style-type: none">• Procedures and/or functions.• Arrays.• Record data type.• Parameters passed by reference.	40 <ul style="list-style-type: none">• 10• 10• 10• 10
Good program style <ul style="list-style-type: none">• Descriptive variable names and meaningful comments.• Good use of control statements.• In general, well written.	30 <ul style="list-style-type: none">• 10• 10• 10