Lab 2 – WS2024/25

Legacy Software Engineering with PL/1 and COBOL

Stefan Strobl, Mario Pilz

November 24, 2024

1 Introduction

The focus of this exercise is to write a functional program, with associated JCL script, that performs a simple task and writes out its results to the printer. Please note that the lab is intended to be solved individually.

2 Exercise

In this exercise, you will develop a program that calculates the elements of a Fibonacci-like sequence and prints out its members that are in a certain range.

The program gets passed four parameters: i_0, i_1, min, and max. Based on these four numbers, it should generate a sequence by the following rules:

- The first element f_0 is i_0,
- the second element f_1 is i_1,
- every other element f_{n+2} is the sum of the preceding two elements f_n and f_{n+1}.

In other words, the sequence is defined identically to the Fibonacci sequence, except for the initial elements.

The program should generate this sequence and print the elements f_n that are in the interval $[min, max]$, that is, the numbers which are greater or equal to min and less than or equal to max.

At the end, it should print how many elements were printed and how many were calculated in total, including the given initial numbers. The program should be able to work with numbers in the range $[-2,147,483,647; 2,147,483,647]$.

The parameter values should be passed to the program from JCL via the PARM string. On the example of the Lucas sequence [1] between 42 and 76, for PL/1 programs, the

[1] `https://en.wikipedia.org/wiki/Lucas_number`

format should be:

PARM='I0=2 I1=1 MIN=42 MAX=76;' [2]

For COBOL, the format should be:

PARM='000002,000001,0000000042,0000000076' [3]

For these given example parameters, a run should produce the following output:

```
      47
      76
Numbers in interval found: 2
Total numbers calculated: 10
```

3 Submission

The submission is due **on Friday 2025-01-10 at 23:59** via TUWEL.

Please submit the following items in an archive, either Zip or Tar:

- The source code of your implementation (including all complementary information and files, such as JCL)
- A sample execution of your program, consisting of
  - Input parameters (please select non-trivial input)
  - All output. Please separate the output of your program from the remaining job output.
- A document explaining your solution in either English or German language. Document at least:
  - the names and purposes of the required data sets
  - an explanation how your program solves the posed problem
  - the steps you performed to execute the program and obtain the output

  The core of this assignment is the implementation of the described functionality with parameter passing from JCL to the program. The submission document should focus on these aspects. While navigation and usage of the development environment should still be described, it can be assumed that the reader is already familiar with that and the description does therefore not need to be as detailed as in Lab 1.

[2] This format was chosen because it is easy to write and understand, and PL/1 has a specialized facility to handle it.

[3] This format was chosen because COBOL does not have a facility for convenient parsing of argument strings as the specified for the PL/1 version. COBOL is mostly equipped to deal with fixed size data. Delimiters were included to give the format a minimum of readability.

- Any known issues or limitations you are aware of.

4 Tipps

Do not hesitate to ask questions on the discussion forum or after lectures. If really stuck it is also possible to contact us via mail.

We also want to encourage you to engage in discussion with your colleagues.

If you encounter any restrictions that you think are due to the use of the (very) old compilers (as present in the TurnKey systems), please document as part of your solution.