

1. The Fibonacci numbers are  $F_{i+1} = F_i + F_{i-1}$  where  $F_1=1$  and  $F_0=0$ . Using  $N=64$ , for example, write:
  - a. Main program that calls `genFibTable(FIB,N)`, `lookupFib(FIB,13)` and `findFibLE(FIB,180)`. FIB is a label.
  - b. Subroutine **`lookupFib(X,Y)`**. Given a table of Fibonacci numbers starting at address  $X$  in  $r4$  and a value  $Y$  in  $r5$ , where  $0 \leq Y < N$ , find  $F_Y$  in the table and return it in register  $r2$ .
  - c. Subroutine **`findFibLE(X,Z)`**. Given a table of Fibonacci numbers starting at address  $X$  in  $r4$ , and a value  $Z$  in  $r5$ , where  $0 \leq Z < F_{N-1}$ , find the largest Fibonacci number  $F_i \leq Z$  and return it in register  $r2$ .
  - d. Subroutine **`genFibTable(X,Y)`**. Compute the first  $Y$  Fibonacci numbers and store them *in ascending order* in memory. Start the table at location  $X$  in register  $r4$ , and value  $Y$  is in register  $r5$ .

```
.equ    N, 64
.global _start:
.text
/* ***** */
_start:
```

```
/* ***** */
lookupFib:
```

```
/* ***** */
findFibLE:
```

```
/* ***** */
genFibTable:
```

```
.data
FIB:
.skip 4*N
.end
```