Problem set 5

TDT 4205 - Sigurd Strømsem

1 Assembly exercise: (compiles and returns correct result)

```
.section .data

str:
    .string "%ld is a prime factor\n"

.section .text
.globl main
factor:
    pushq %rbp              # save base pointer
    movq %rsp, %rbp         # set new base pointer
    subq $16, %rsp          # allocate space for n and f
    movq %rdi, -8(%rbp)     # save only argument
    sarq $1, %rdi           # bitshift by one (= divide by two)
    movq %rdi, -16(%rbp)    # move f to stack

loop:
    movq -8(%rbp), %rax     # move arg to new scratch register
    movq $0, %rdx           # move 0 to upper part of rdx:rax
    idivq -16(%rbp)         # divide by f
    cmpq $0, %rdx           # remainder is stored in dx
    je if_statement         # jump to if when equal to 0
    decq -16(%rbp)          # f--
    jmp loop                # next iteration

if_statement:
    cmpq $1, -16(%rbp)
    je else                 # if equal to one, jump to else
    movq -16(%rbp), %rdi
    call factor
    movq -8(%rbp), %rax
    movq $0, %rdx
    idivq -16(%rbp)         # divide by f
    movq %rax, %rdi         # result from division to rdi
    call factor
    jmp return

else:
    cmpq $1, -8(%rbp)
    je return
    movq $str, %rdi         # load string in rdi
    movq -8(%rbp), %rsi     # load n in rsi, arg 2
    call printf             # call compiled object printf
    jmp return
```

```
return:
    leave
    ret

main:
    pushq %rbp
    movq %rsp, %rbp
    movq $1836311903, %rdi
    call factor
    leave
    ret
```

2 See attached coded. The vsl compiler should be fully implemented.