

Report for Programming Assignment (#4)

-Regular Expression Library-

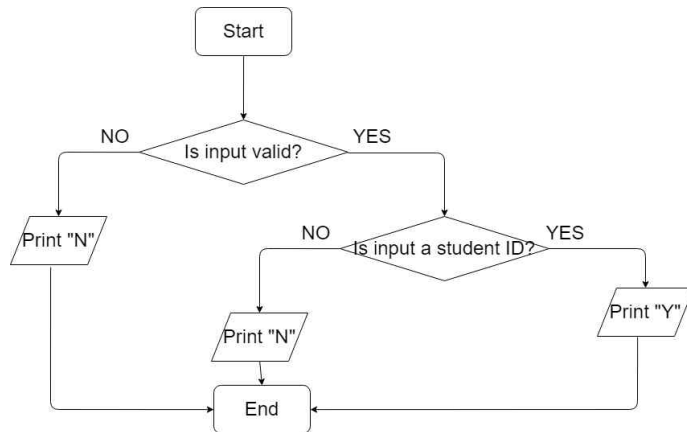
Submission Date	2018.05.15
Professor in Charge	Lee Hee-Jo (이희조 교수님)
Subject Name	Theory of Computation
Department	Computer Science&Engineering
Student Number	2018320161
Student Name	Song Dae Sun (송대선)

1. environment:

Windows 10 pro, Python 3.7.3, Pycharm Community Edition Version 2019.1

2. algorithm:

problem1:



input validation is 10 numbers

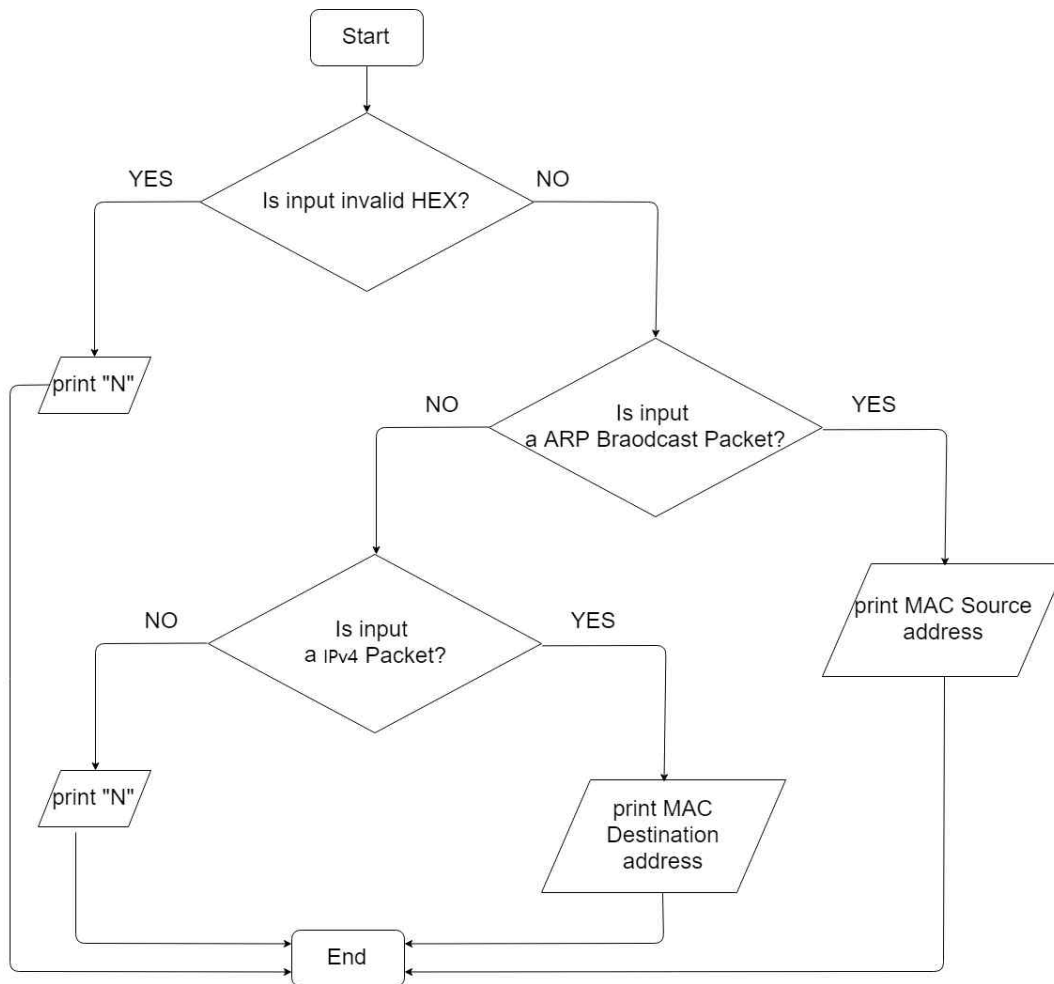
student ID is restricted:

years -> 1905~2019

major -> 320

classnumber -> 000~199

problem2:

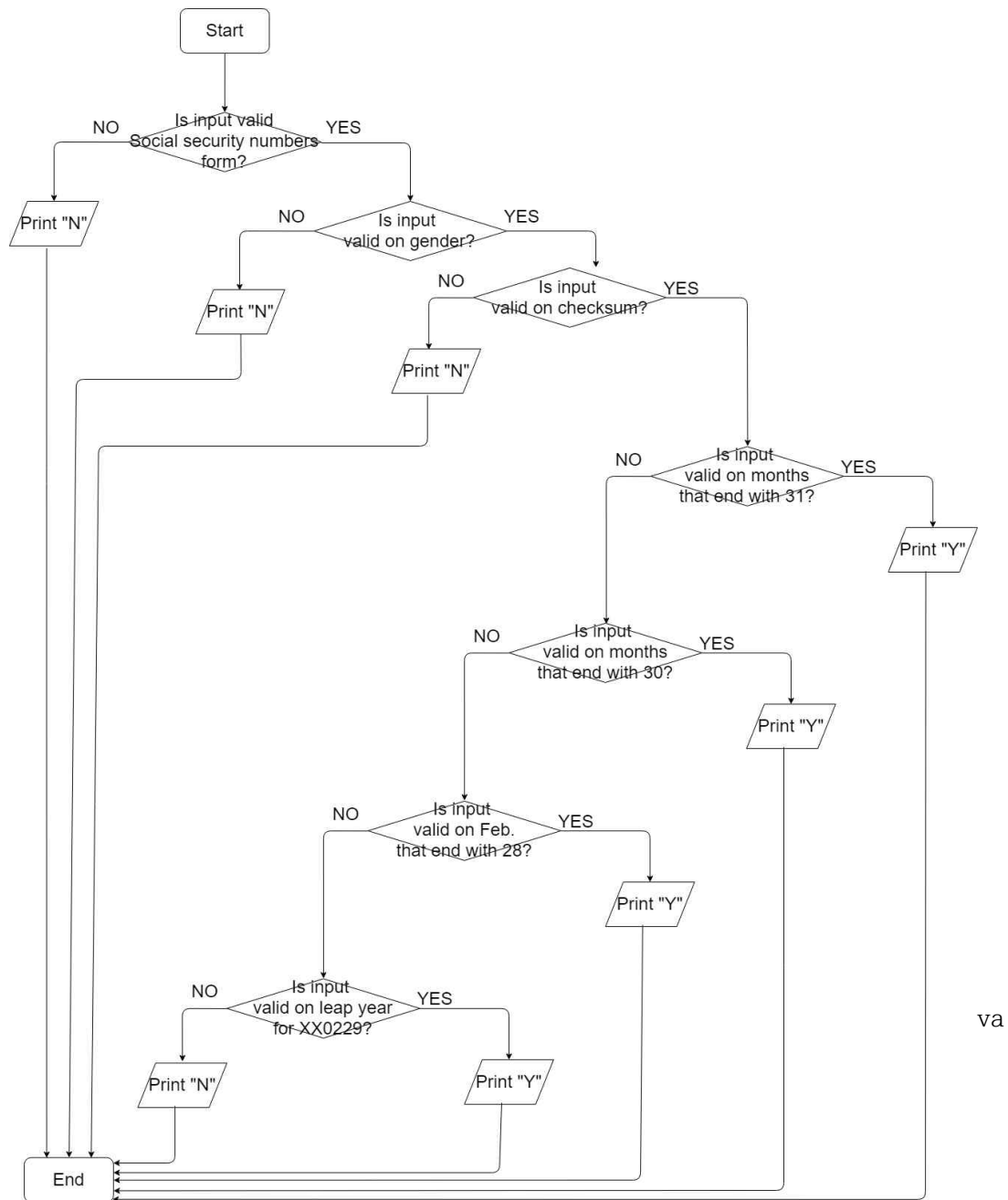


invalid HEX: means a string that has any character(s) belonging to g(G) to z(Z) and non-word character(s).

If first 6bytes are "ff ff ff ff ff ff" and last 2bytes are "08 06",
It is the ARP Broadcast packet.

If last 2bytes are "08 00"
It is the Ipv4 packet.

problem3:



valid social security number form is

“(6-digits) + (single space) + ‘-’ + (single space) + (7-digits)” as given

Validity on gender gives 1 or 2 from 1918 to 1999, and 3 or 4 from 2000 to 2017.

Validity on checksum calculates the checksum for the input using "getChecksum," a function for obtaining checksum, and checks whether it is valid.

Validity on months that end with 31 days, if input belongs to those months, check whether input is within 31 days.

Validity on months that end with 30 days, if input belongs to those months, check whether input is within 30 days.

Validity on Feb, if input belongs to February, check whether input is within 28 days.

Validity on leaf year checks if input is leap year or not when input is 29 February. The criteria for the leap year is the **modified algorithm** specified in ppt.
-> if (year is divisible by 4) then (it is a leap year)

3. explanation

#This function is used to obtain checksum.

function getChecksum (string) **returns** checksum

Declare mask **as** "234567 - 892345"

Declare mulSum **as** 0

For each i **in** range(0,15) **except** 6, 7, 8

 mulSum+=string[i]*mask[i]

 #following exactly what PPT requires.

Obtain a remainder by dividing the mulSum by 11.

Obtain checkSum by 11 subtract the reminder.

if checkSum>=10 **then return** checkSum%10

else then return checkSum

Read an input in a single line.

Read the number of inputs for each problem

#problem 1

For i **in** problem1 **do**

if i is in a valid format **then**

if i is a valid study number **then**

print "Y"

else then

print "N"

else then

print "N"

#problem 2

for i **in** problem2 **do**

if i is configured with characters other than HEX **then**

print "N"

else then

if i is an ARP Broadcast Packet **then**

print Source MAC Address

else if i is an IPv4 Packet **then**

print Destination MAC Address

else then

print "N"

```

#problem 3
for i in problem3 do
    if i is a social security numbers in a valid format then
        if i is valid on gender then
            calculate checksum of i by function getChecksum(line[i])
            if checksum in i is valid then
                if i is in months that ends with 31 days then
                    print "Y"
                else if i is in months that ends with 30 days then
                    print "Y"
                else if i is in Feb. that ends with 28 days then
                    print "Y"
                else if i is a leap year for Feb. 29th then
                    print "Y"
                else then
                    print "N"
            else then
                print "N"
        else then
            print "N"
    else then
        print "N"

# Shut down the program when you receive any input
# it prevents the console from shutting down immediately
input "엔터키를 눌러 종료합니다."

```

4. regex

: This program uses “Python re”

5. result

input	output
10	
2011320200234	N
2017322001	N
2008870121	N
2011320003	Y
2017322001	N
2008870121	N
2018320088	Y
2011000000	N
2013333111	N
2015320168	Y
5	
ffffffffffff00089f79bde00806000108000604 000100089f79bde0a398a2300000000000000 a398a258	00089f79bde0
ffffffffffff00089f79bde00806000108000604 000100089f79bde0a398a2300000000000000 a398a 258	N
d05099a54f5064ae0c4132c0080045000028 b4274000f106a9ed36c0af2aa398a23701bb e6062b9cd09bff5b31495011007f6efb	d05099a54f50
af2aa398a23764ae0c4132c0080645000028 b4274000f106a9ed36c0af2aa398a23701bb e6062b9cd09bff5b31495011007f6efb	N
79bde0a398a200089f79bde0084200010800 0604000100089f79bde0a398a230000000000 0000a398a258	N
11	
190228 - 3234563	Y
111231 - 3111220	Y
180101 - 3365986	Y
920229 - 1239536	Y
190228 - 3234567	N
971130 - 3325808	N
910125-1289204	N
170101 - 1365983	N
111231 - 3111220	Y
111231 - 3111221	N
180101 - 3365986	Y