프로그래밍 실습 1

```
from __future__ import print_function
from sys import stdin

def printf(str, *args):
	print(str % args, end='')

def my_opening():
	printf("# 지금부터 여러분을 신비한 마술의 세계로 초대합니다.₩n₩n₩n")
	printf("1 - 31의 숫자중 마음에 드는 숫자를 생각해 보세요. ₩n₩n₩n")

def my_card(index):
	cardname = chr(ord('A') + index - 1)
	while True:
		printf("    --------- %s 카드 ---------₩n" % cardname )
		for i in range(0, 4):
			for j in range(0, 4):
				printf("%7d" % card[index-1][4*i + j])
			printf("₩n")
		printf("    -------------------------₩n")
		printf("₩n₩n")
		printf("%s 카드에 생각한 숫자가 있다면 YES(1번),₩n" % cardname)
		printf("없다면 NO(0번)을 선택하여 주시기 바랍니다. : ")
		result = int(stdin.readline())
		printf("₩n₩n")
		if result>-1 and result<2:
			return result

def my_res(a,b,c,d,e):
	res = (e*2*2*2*2) + (d*2*2*2) + (c*2*2) + (b*2) + (a)
	printf("당신이 마음에 드는 숫자는 %d입니다.₩n₩n₩n" % res);
	printf("어때요? 신기하죠!!₩n");
	printf("지금까지 마술의 세계였습니다.₩n₩n");
card = [
	[1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31],
	[2,3,6,7,10,11,14,15,18,19,22,23,26,27,30,31],
	[4,5,6,7,12,13,14,15,20,21,22,23,28,29,30,31],
	[8,9,10,11,12,13,14,15,24,25,26,27,28,29,30,31],
	[16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31]
	]
my_opening()

printf("생각하셨어요? 시작하려면 엔터를 누르시오.")
stdin.readline()

print("₩n₩n")
a = my_card(1)
b = my_card(2)
c = my_card(3)
d = my_card(4)
e = my_card(5)

my_res(a,b,c,d,e)

stdin.readline()
```

프로그래밍 실습 2

```
from __future__ import print_function
import sys

def printf(str, *args):
        print(str % args, end='')

printf("입력(Input)에 대한 소수 판별을 해드립니다.₩n")
printf("소수 판별하기 위한 N값을 입력해 주세요 : ")
inputvalue = int(sys.stdin.readline())

for i in range(2, inputvalue):
        if inputvalue%i == 0 and inputvalue != i:
                printf("₩ninput number -->%5d  Not Prime number!!₩n" % inputvalue)
                input_enter(0)
                sys.exit(0);

if inputvalue>1:
        printf("₩ninput number -->%5d    Prime number!!₩n" % inputvalue)
else:
        printf("₩ninput number -->%5d    Not Prime number!!₩n" % inputvalue)

sys.stdin.readline()
```

프로그래밍 실습 3

```
from __future__ import print_function
from sys import stdin

def printf(str, *args):
        print(str % args, end='')

arr_1 = [[1,2,3,4],[4,3,2,1],[2,5,7,9],[6,3,2,1]]
arr_2 = [[1,5,6,7],[8,3,1,7],[6,2,8,3],[9,2,1,2]]

# print Arr_1
printf("배열_1 : ₩n")
for i in range(0, 4):
        printf("            ")
        for j in range(0, 4):
                printf("%4d" % arr_1[i][j])
        printf("₩n")

printf("₩n₩n")

# print Arr_2
printf("배열_2 : ₩n")
for i in range(0, 4):
        printf("            ")
        for j in range(0, 4):
                printf("%4d" % arr_2[i][j])
        printf("₩n")

printf("₩n₩n")

# init res
res = [[0 for _ in range(4)] for _ in range(4)]

# multi Arr_1 Arr_2 with ( i , j , k )
printf("배열_1 * 배열_2 (i,j,k 순서로) : ₩n")
for i in range(0, 4):
        for j in range(0, 4):
                for k in range(0, 4):
                        res[i][j] += arr_1[i][k] * arr_2[k][j]

# print res
for i in range(0, 4):
        printf("                                    ");
        for j in range(0, 4):
                printf("%4d" % res[i][j])
        printf("₩n")

# ------------------------------------------------------------
# init res
res = [[0 for _ in range(4)] for _ in range(4)]

# multi Arr_1 Arr_2 with ( i , k , j )
printf("배열_1 * 배열_2 (i,k,j 순서로) : ₩n")
for i in range(0, 4):
        for k in range(0, 4):
                for j in range(0, 4):
                        res[i][j] += arr_1[i][k] * arr_2[k][j]
```

```
# print res
for i in range(0, 4):
        printf("                                        ");
        for j in range(0, 4):
                printf("%4d" % res[i][j])
        printf("₩n")

printf("₩n₩n결과는 같다. 그러나 실행시간은 차이가 난다.₩n")

stdin.readline()
```

프로그래밍 실습 4

```
from __future__ import print_function
from sys import stdin, exit

def printf(str, *args):
        print(str % args, end='')

MAX = 3

def matrixout(mx):
        print("₩n")
        for i in range(0, MAX):
                for j in range(0, MAX):
                        printf("%15E " % mx[i][j])
                printf("₩n")

mxa = [[0 for _ in range(MAX)] for _ in range(MAX)]
inversemx = [[0 for _ in range(MAX)] for _ in range(MAX)]

printf("please input 3 by 3 matrix : ₩n");
printf("ex) 1 2 3₩n      4 5 6₩n      7 8 9₩n₩n");
# read matrix
list = []
while len(list) < MAX * MAX:
        list += stdin.readline().strip("₩n").split()
for i in range(0, MAX):
        for j in range(0, MAX):
                mxa[i][j] = float(list[i*MAX + j])

# write images of matrix A
printf("%5cmatrix A₩n" % ' ')
matrixout(mxa)

# determinant
sumx = 0
sumy = 0
determins = 0
for j in range(0, MAX):
        l = (j+1) % MAX
        n = (j+2) % MAX
        sumx += mxa[0][j]*mxa[1][l]*mxa[2][n]
        sumy += mxa[0][j]*mxa[1][n]*mxa[2][l]
determins = sumx - sumy
printf("₩n")
printf("%5cdeterminant%7c:%15E₩n" %(' ',' ',determins) )

# inverse matrix
if determins == 0:
        printf("%5cinverse matrix does not exist₩n" % ' ')
        exit(1)
```

```
for i in range(0, MAX):
        for j in range(0, MAX):
                if i == j:
                        if i==0:
                                inversemx[0][0] = mxa[1][1]*mxa[2][2]-mxa[1][2]*mxa[2][1]
                        elif i==1:
                                inversemx[1][1] = mxa[0][0]*mxa[2][2]-mxa[0][2]*mxa[2][0]
                        elif i==2:
                                inversemx[2][2] = mxa[0][0]*mxa[1][1]-mxa[0][1]*mxa[1][0]
                else:
                        l = i+j
                        l = {1:2, 2:1, 3:0}.get(l)
                        inversemx[i][j]=(mxa[i][l]*mxa[l][j]-mxa[i][j]*mxa[l][l])*1.0

for i in range(0, MAX):
        for j in range(0, MAX):
                inversemx[i][j] /= determins
printf("₩n%5cinverse matrix₩n" % ' ')
matrixout(inversemx)
```

프로그래밍 실습 5

```
from __future__ import print_function
from sys import stdin

def printf(str, *args):
        print(str % args, end='')

maxvalue = 3

c = [[False for _ in range(maxvalue)] for _ in range(maxvalue)]
d = [[False for _ in range(maxvalue)] for _ in range(maxvalue)]
e = [[False for _ in range(maxvalue)] for _ in range(maxvalue)]

def mat_compute(a, b):
        for i in range(0, maxvalue):
                for j in range(0, maxvalue):
                        for k in range(0, maxvalue):
                                c[i][j] = a[i][j] and b[i][j]
                                d[i][j] = a[i][j] or b[i][j]
                                if k == 0:
                                        e[i][j] = a[i][k] * b[k][j]
                                else:
                                        e[i][j] = e[i][j] or (a[i][k] and b[k][j])

def print_matrix(x):
        for i in range(0, maxvalue):
                printf("₩n")
                for j in range(0, maxvalue):
                        printf("%2d" % x[i][j])
        printf("₩n")

a = [[False for _ in range(maxvalue)] for _ in range(maxvalue)]
b = [[False for _ in range(maxvalue)] for _ in range(maxvalue)]

printf("₩n")
printf("부울 행렬 A(3*3)를 입력하세요.₩n(0과 1만 사용하세요)₩n")
printf("ex) 0 1 0₩n      0 0 0₩n      1 0 1₩n₩n")

list = []
while len(list) < maxvalue * maxvalue:
        list += stdin.readline().strip("₩n").split()
for i in range(0, maxvalue):
        for j in range(0, maxvalue):
                a[i][j] = bool(int(list[i*maxvalue + j]))
print(a)
printf("₩n 부울 행렬 B(3*3)을 입력하세요.₩n")

list = []
while len(list) < maxvalue * maxvalue:
        list += stdin.readline().strip("₩n").split()
for i in range(0, maxvalue):
        for j in range(0, maxvalue):
                b[i][j] = bool(int(list[i*maxvalue + j]))
print(b)
mat_compute(a,b)
printf("₩n")
```

```
printf("A MEET B₩n")
print_matrix(c)
printf("₩n")

printf("A JOIN B₩n")
print_matrix(d)
printf("₩n")

printf("A Boolean product B₩n")
print_matrix(e)

stdin.readline()
```