**주차장 모의 프로그램 소스코드 32212390 신휘**

# 인트로

def intro() :

underbar()

print("이 프로그램은 주차장 모의 프로그램입니다.")

print("해당 프로그램의 주차장은 B1, B2, B3으로 구성되어 있습니다.")

print("주차장의 자리는 기본적으로 각각 %d개, %d개, %d개로 설정되어 있으며 여러 메뉴를 통해 주차장을 조작할 수 있습니다." %(B1, B2, B3))

# 프로그램 설명

def explain() :

underbar()

print("먼저 차량 입차 메뉴에 대한 설명입니다.")

print("입차 위치를 고른 후 차량의 수를 입력하면, 기본으로 설정된 주차 공간을 넘지 않는 선에서 차량을 입차합니다.")

print()

print("다음은 차량 출차 대기 메뉴에 대한 설명입니다.")

print("출차하는데 기다릴 시간을 입력하면, 분 당 1대의 차량이 각 위치에서 랜덤으로 출차합니다.")

print("만약 입차된 차량보다 많은 대기 시간을 입력하면 입차되어 있는 차량을 모두 출차합니다.")

print()

print("다음은 주차 공간 확인 메뉴에 대한 설명입니다.")

print("주차 공간 계산 함수를 통해서 주차 공간 그림에서 입차되어 있는 차량의 수가 남아있는 차량의 수와 일치시킵니다.")

print("그림에서의 주차 공간은 0이 비어있는 공간, 1이 주차되어 있는 공간을 의미합니다.")

underbar()

re()

# 메뉴 선택

def options() :

print()

underbar()

print("다음은 주차장 조작 메뉴입니다.")

print("1. 프로그램 설명")

print("2. 차량 입차")

print("3. 차량 출차 대기")

print("4. 주차 공간 확인")

print("5. 종료")

underbar()

n = int(input("원하시는 메뉴를 선택해주세요 : "))

if n == 1 :

explain()

elif n == 2 :

entrance\_options() # 입차 위치 선택 함수 호출

elif n == 3 :

wait() # 출차 대기 함수 호출

elif n == 4 :

view() # 주차 공간 확인 함수 호출

elif n == 5 :

print("프로그램을 종료합니다.")

quit()

else :

print("메뉴에 해당하는 번호를 선택해주세요.")

options()

# 입차 위치 선택 함수

def entrance\_options() :

underbar()

print("다음은 차량 입차 위치 메뉴입니다.")

print("1. B1")

print("2. B2")

print("3. B3")

print("4. 다른 메뉴 선택하기")

underbar()

n = int(input("차량을 입차할 위치를 선택해주세요 : "))

# 입차 위치에 따른 계산

if n == 1 :

entrance\_local("B1") # 입차 차량 수 입력 함수 호출

global b1ent

if(car > B1-b1ent) : # 입차 차량이 남아있는 공간보다 많을 때

entrance\_over() # 입차 차량 오버 함수 호출

else :

b1ent = entrance(car, b1ent, "B1", B1) # 적절하면 입차 함수 호출

re()

elif n == 2 :

entrance\_local("B2")

global b2ent

if(car > B2-b2ent) :

entrance\_over()

else :

b2ent = entrance(car, b2ent, "B2", B2)

re()

elif n == 3 :

entrance\_local("B3")

global b3ent

if(car > B3-b3ent) :

entrance\_over()

else :

b3ent = entrance(car, b3ent, "B3", B3)

re()

elif n == 4 :

options()

else :

print("메뉴에 해당하는 번호를 선택해주세요.")

entrance\_options()

# 입차 차량 수 입력 함수

def entrance\_local(n) :

global car

car = int(input("%s에 입차할 차량의 수를 입력해주세요 : " %n))

if(car < 0) :

print("차량 수를 다시 입력해주세요.")

entrance\_local(n)

# 입차 차량 오버 함수

def entrance\_over():

print()

print("입차할 차량의 수가 현재 남아있는 주차 공간보다 많습니다.")

print("메뉴를 다시 선택해주세요.")

entrance\_options()

# 입차 함수

def entrance(car, rest, local, num) :

rest += car

underbar()

print("%d대의 차량을 %s에 입차하였습니다." %(car, local))

print("%s에 현재 주차 공간 : %d / %d" %(local, rest, num))

underbar()

return rest

# 출차 대기 함수

def wait() :

global b1ent, b2ent, b3ent, b1out, b2out, b3out

time = int(input("차량이 출차하는데 기다릴 시간을 분 단위로 입력해주세요 : "))

# 시간이 입차되어 있는 공간보다 더 많을 때

if(b1ent + b2ent + b3ent < time) :

b1ent, b2ent, b3ent = wait\_over(b1ent, b2ent, b3ent) # 출차 오버 함수 호출

# 시간이 마이너스일 때

elif(time <= 0) :

print("시간을 다시 입력해주세요.")

wait()

else :

wait\_random(time) # 적절하면 출차 랜덤 함수 호출

b1ent -= b1out

b2ent -= b2out

b3ent -= b3out

underbar()

print("B1에 %d대의 차량이 출차하였습니다. 현재 주차 공간 : %d / %d" %(b1out, b1ent, B1))

print("B2에 %d대의 차량이 출차하였습니다. 현재 주차 공간 : %d / %d" %(b2out, b2ent, B2))

print("B3에 %d대의 차량이 출차하였습니다. 현재 주차 공간 : %d / %d" %(b3out, b3ent, B3))

underbar()

re()

# 출차 랜덤 함수, 분 당 차량 1대가 출차한다고 했을 때, 각 위치에서 랜덤으로 자리 추출

def wait\_random(time) :

global b1ent, b2ent, b3ent, b1out, b2out, b3out

b1out = randrange(0, time+1) # 0부터 time까지 랜덤

b2out = randrange(0, time+1)

b3out = randrange(0, time+1)

if(b1out > b1ent) : # 출차하는 차량이 입차 되어 있는 차량보다 더 많으면 출차 차량을 입차 되어 있는 차량과 같도록 계산

b1out = b1ent

if(b2out > b2ent) :

b2out = b2ent

if(b3out > b3ent) :

b3out = b3ent

while(b1out + b2out + b3out != time): # 랜덤으로 time과 분 당 차량 1대가 일치하는 출차 차량이 나올 때까지 계산

wait\_random(time)

# 출차 오버 함수, 시간이 입차 되어 있는 공간보다 많을 때, 모든 공간을 출력하고 0으로 설정

def wait\_over(b1ent, b2ent, b3ent) :

global b1out, b2out, b3out

b1out = b1ent

b2out = b2ent

b3out = b3ent

return 0, 0, 0

# 주차 공간 확인 함수

def view() :

underbar()

print("B1에 현재 주차 공간 : %d / %d" %(b1ent, B1))

rest\_view(b1rest\_view, b1ent, B1) # 주차 공간 계산 함수 호출

underbar()

print("B2에 현재 주차 공간 : %d / %d" %(b2ent, B2))

rest\_view(b2rest\_view, b2ent, B2)

underbar()

print("B3에 현재 주차 공간 : %d / %d" %(b3ent, B3))

rest\_view(b3rest\_view, b3ent, B3)

print()

print("그림에서의 주차 공간은 0이 비어있는 공간, 1이 주차되어 있는 공간을 의미합니다.")

underbar()

re()

# 주차 공간 계산 함수

def rest\_view(view, ent, loc) :

if(view.count(1) < ent) : # 입차 되어 있는 공간이 그림 속의 1 공간보다 많으면, 수가 일치할 때까지 그림 속의 랜덤한 0 공간을 1 공간으로 설정

while(ent != view.count(1)) :

i = randrange(loc)

if(view[i] == 0) :

view[i] = 1

elif(view.count(1) > ent) : # 그림 속의 1 공간이 입차 되어 있는 공간보다 많으면, 수가 일치할 때까지 그림 속의 랜덤한 1 공간을 0 공간으로 설정

while(ent != view.count(1)) :

i = randrange(loc)

if(view[i] == 1) :

view[i] = 0d

print(view)

# 테두리 설정 함수

def underbar() :

print("================================================================================")

# 메뉴로 돌아가기 함수

def re() :

q = input("아무 키나 입력하시면 메뉴로 돌아갑니다 : ")

if(q != "") :

options()

# 메인 함수

from random import \*

B1 = 30 # 기본 설정 주차 공간

B2 = 50

B3 = 100

b1ent = 0 # 입차되어 있는 차량 수

b2ent = 0

b3ent = 0

b1rest\_view = [0 for i in range(B1)] # 주차 공간 수만큼의 사이즈로 0을 리스트로 설정

b2rest\_view = [0 for i in range(B2)]

b3rest\_view = [0 for i in range(B3)]

intro()

options()