

[illegible]

### # 접합 T1·T2와 T2·T1에 대한 결과를 출력

[illegible]

```

printf("\n\t\t\t\t\t 문자열 T2에 대해 3개의 문자열을 입력하라.\n")
numbers = ["첫", "두", "세"]
for i in range(0, 3):
    printf("\t\t\t\t\t %s번째 문자열 : %s\n", numbers[i])
    printf("\r\r\t\t\t\t\t")
    t2.append(stdin.readline().strip("\n"))

# T2에 대한 Image를 출력
printf("\n\t\t\t\t\t Image of T2\n")
for i in range(0, 3):
    printf("\t\t\t\t\t <xs>\n" % t2[i])

# 접합 T1·T2를 계산하는 페인트공 알고리즘의 실행 시간으로 출력하고 접합을 텍스트 파일에 저장
printf("\n\t\t\t\t\t 페인트공 알고리즘 : T1\u25cfT2")

printf("\n\t\t\t\t\t \"pointer_result.txt\" 파일이 생성되었다.\n")
start = time.perf_counter()

for g in range(0, 100):
    for j in range(0, 3):
        for i in range(0, 5):
            t1t2.append(t1[i] + t2[j])

end = time.perf_counter()

for i in range(0, 15):
    fp2.write("\t\t\t\t\t %s\n" % t1t2[i])

printf("\t\t\t\t\t 실행 시간 : %.3f(ms)\n", (end-start)*1000.0)

# 접합 T1·T2를 계산하는 '반환값을 주는 포인터 없는 알고리즘'의 실행 시간으로 출력하고 접합을
# 텍스트 파일에 저장
printf("\n\t\t\t\t\t 반환값을 주는 포인터 없는 알고리즘 : T1\u25cfT2")
printf("\n\t\t\t\t\t \"notpointer_result.txt\" 파일이 생성되었다.\n")
start = time.perf_counter()

for g in range(0, 100):
    for j in range(0, 3):
        for i in range(0, 5):
            for l in range(0, 20001):
                try:
                    if t1[i][l] == "\000":
                        break;
                except IndexError:
                    break;
            t1t2.append(t1[i] + t2[j])

end = time.perf_counter()
for i in range(0, 15):
    fp3.write("\t\t\t\t\t %s\n" % t1t2[i])
printf("\t\t\t\t\t 실행 시간 : %.3f(ms)\n", (end-start)*1000.0)

# 생성된 파일들을 닫음
fp1.close()
fp2.close()
fp3.close()

stdin.readline()

```

### [프로그래밍 실습 3]

```
from __future__ import print_function
from sys import stdin

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

class State:
    # 상태를 표현하기 위한 멤버 변수 선언
    def __init__(self):
        self.m = 0
        self.c = 0
        self.p = ''
    # 상태를 초기화하기 위한 함수
    def state_init(self):
        self.m = 2
        self.c = 2
        self.p = 'r'
# 상태 전이를 진행하는 함수
def state_moving(self, m, c):
    if self.p == 'r': # 배가 오른쪽에 있을 때
        # 현재 상태가 (2, 2, R)일 때
        if self.m == 2 and self.c == 2:
            # 상태 (2, 2, R)로부터 입력 (0, 1)에 따른 상태 (2, 1, L)로의 전이
            if m == 0 and c == 1:
                self.m = 2
                self.c = 1
                self.p = 'l'
            # 상태 (2, 2, R)로부터 입력 (0, 2)에 따른 상태 (2, 0, L)로의 전이
            elif m == 0 and c == 2:
                self.m = 2
                self.c = 0
                self.p = 'l'
            # 상태 (2, 2, R)로부터 입력 (1, 0)에 따른 상태 종결이 불가능한 경우
            elif m == 1 and c == 0:
                printf("식인종들이 선교사(들)을 잡아먹음!")
                return -1
            # 상태 (2, 2, R)로부터 입력 (1, 1)에 따른 상태 (1, 1, L)로의 전이
            elif m == 1 and c == 1:
                self.m = 1
                self.c = 1
                self.p = 'l'
            # 상태 (2, 2, R)로부터 입력 (2, 0)에 따른 상태 (0, 2, L)로의 전이
            elif m == 2 and c == 0:
                self.m = 0
                self.c = 2
                self.p = 'l'
        # 현재 상태가 (2, 1, R)일 때
        elif self.m == 2 and self.c == 1:
            # 상태 (2, 1, R)로부터 입력 (0, 1)에 따른 상태 (2, 0, L)로의 전이
            if m == 0 and c == 1:
                self.m = 2
                self.c = 0
                self.p = 'l'
            # 상태 (2, 1, R)로부터 입력 (0, 2)에 따른 상태 종결이 불가능한 경우
            elif m == 0 and c == 2:
                printf("식인종들이 선교사(들)을 잡아먹음!")
                return -1
```

```

# 상태 (2, 1, R)로부터 입력 (1, 0)에 따른 상태 (1, 1, L)로의 전이
elif m == 1 and c == 0:
    self.m = 1
    self.c = 1
    self.p = 'l'
# 상태 (2, 1, R)로부터 입력 (1, 1)에 따른 상태 (1, 0, L)로의 전이
elif m == 1 and c == 1:
    self.m = 1
    self.c = 0
    self.p = 'l'
# 상태 (2, 1, R)로부터 입력 (2, 0)에 따른 상태 (0, 1, L)로의 전이
elif m == 2 and c == 0:
    self.m = 0
    self.c = 1
    self.p = 'l'
# 현재 상태가 (2, 0, R)일 때
elif self.m == 2 and self.c == 0:
    # 상태 (2, 0, R)로부터 입력 (0, 1)에 따른 상태 종결이 불가능한 경우
    if m == 0 and c == 1:
        printf("식인종들이 선교사(들)을 잡아먹음!")
        return -1
    # 상태 (2, 0, R)로부터 입력 (0, 2)에 따른 상태 종결이 불가능한 경우
    elif m == 0 and c == 2:
        printf("식인종들이 선교사(들)을 잡아먹음!")
        return -1
    # 상태 (2, 0, R)로부터 입력 (1, 0)에 따른 상태 (1, 0, L)로의 전이
    elif m == 1 and c == 0:
        self.m = 1
        self.c = 0
        self.p = 'l'
    # 상태 (2, 0, R)로부터 입력 (1, 1)에 따른 상태 종결이 불가능한 경우
    elif m == 1 and c == 1:
        printf("식인종들이 선교사(들)을 잡아먹음!")
        return -1
    # 상태 (2, 0, R)로부터 입력 (2, 0)에 따라 상태 (0, 0, L)로 전이하고 종결
    elif m == 2 and c == 0:
        self.m = 0
        self.c = 0
        self.p = 'l'
        printf("무사히 왼쪽 강가에 도달함. : \t축하합니다!!!")
        return 1
# 현재 상태가 (1, 1, R)일 때
elif self.m == 1 and self.c == 1:
    # 상태 (1, 1, R)로부터 입력 (0, 1)에 따른 상태 (1, 0, L)로의 전이
    if m == 0 and c == 1:
        self.m = 1
        self.c = 0
        self.p = 'l'
    # 상태 (1, 1, R)로부터 입력 (0, 2)에 따른 상태 종결이 불가능한 경우
    elif m == 0 and c == 2:
        printf("식인종들이 선교사(들)을 잡아먹음!")
        return -1
    # 상태 (1, 1, R)로부터 입력 (1, 0)에 따른 상태 (0, 1, L)로의 전이
    elif m == 1 and c == 0:
        self.m = 0
        self.c = 1
        self.p = 'l'
    # 상태 (1, 1, R)로부터 입력 (1, 1)에 따라 상태 (0, 0, L)로 전이하고 종결

```

상태에 도달

상태에 도달

```
elif m == 1 and c == 1:
    self.m = 0
    self.c = 0
    self.p = 'l'
    printf("무사히 왼쪽 강가에 도달함. : \t축하합니다!!!")
    return 1
# 상태 (1, 1, R)로부터 입력 (2, 0)에 따른 상태 종결이 불가능한 경우
elif m == 2 and c == 0:
    printf("식인종들이 선교사(들)을 잡아먹음!")
    return -1
# 현재 상태가 (0, 2, R)일 때
elif self.m == 0 and self.c == 2:
    # 상태 (0, 2, R)로부터 입력 (0, 1)에 따른 상태 (0, 1, L)로의 전이
    if m == 0 and c == 1:
        self.m = 0
        self.c = 1
        self.p = 'l'
    # 상태 (0, 2, R)로부터 입력 (0, 2)에 따라 상태 (0, 0, L)로 전이하고 종결
```

상태에 도달

```
elif m == 0 and c == 2:
    self.m = 0
    self.c = 0
    self.p = 'l'
    printf("무사히 왼쪽 강가에 도달함. : \t축하합니다!!!")
    return 1
# 상태 (0, 2, R)로부터 입력 (1, 0)에 따른 상태 종결이 불가능한 경우
elif m == 1 and c == 0:
    printf("식인종들이 선교사(들)을 잡아먹음!")
    return -1
# 상태 (0, 2, R)로부터 입력 (1, 1)에 따른 상태 종결이 불가능한 경우
elif m == 1 and c == 1:
    printf("식인종들이 선교사(들)을 잡아먹음!")
    return -1
# 상태 (0, 2, R)로부터 입력 (0, 0)에 따른 상태 종결이 불가능한 경우
elif m == 2 and c == 0:
    printf("식인종들이 선교사(들)을 잡아먹음!")
    return -1
# 현재 상태가 (0, 1, R)일 때
elif self.m == 0 and self.c == 1:
    # 상태 (0, 1, R)로부터 입력 (0, 1)에 따라 상태 (0, 0, L)로 전이하고 종결
```

상태에 도달

```
if m == 0 and c == 1:
    self.m = 0
    self.c = 0
    self.p = 'l'
    printf("무사히 왼쪽 강가에 도달함. : \t축하합니다!!!")
    return 1
# 상태 (0, 1, R)로부터 입력 (0, 2)에 따른 상태 종결이 불가능한 경우
elif m == 0 and c == 2:
    printf("식인종들이 선교사(들)을 잡아먹음!")
    return -1
# 상태 (0, 1, R)로부터 입력 (1, 0)에 따른 상태 종결이 불가능한 경우
elif m == 1 and c == 0:
    printf("식인종들이 선교사(들)을 잡아먹음!")
    return -1
# 상태 (0, 1, R)로부터 입력 (1, 1)에 따른 상태 종결이 불가능한 경우
elif m == 1 and c == 1:
    printf("식인종들이 선교사(들)을 잡아먹음!")
```

```

        return -1
        # 상태 (0, 1, R)로부터 입력 (2, 0)에 따른 상태 종결이 불가능한 경우
        elif m == 2 and c == 0:
            printf("식인종들이 선교사(들)을 잡아먹음!")
            return -1
    elif self.p == 'l':
        # 배가 왼쪽에 있을 때
        # 현재 상태가 (2, 1, L)일 때
        if self.m == 2 and self.c == 1:
            # 상태 (2, 1, L)로부터 입력 (0, 1)에 따른 상태 (2, 2, R)로의 전이
            if m == 0 and c == 1:
                self.m = 2
                self.c = 2
                self.p = 'r'
            # 상태 (2, 1, L)로부터 입력 (0, 2)에 따른 상태 종결이 불가능한 경우
            elif m == 0 and c == 2:
                printf("식인종들이 선교사(들)을 잡아먹음!")
                return -1
            # 상태 (2, 1, L)로부터 입력 (1, 0)에 따른 상태 종결이 불가능한 경우
            elif m == 1 and c == 0:
                printf("식인종들이 선교사(들)을 잡아먹음!")
                return -1
            # 상태 (2, 1, L)로부터 입력 (1, 1)에 따른 상태 종결이 불가능한 경우
            elif m == 1 and c == 1:
                printf("식인종들이 선교사(들)을 잡아먹음!")
                return -1
            # 상태 (2, 1, L)로부터 입력 (2, 0)에 따른 상태 종결이 불가능한 경우
            elif m == 2 and c == 0:
                printf("식인종들이 선교사(들)을 잡아먹음!")
                return -1
        # 현재 상태가 (2, 0, L)일 때
        elif self.m == 2 and self.c == 0:
            # 상태 (2, 0, L)로부터 입력 (0, 1)에 따른 상태 (2, 1, R)로의 전이
            if m == 0 and c == 1:
                self.m = 2
                self.c = 1
                self.p = 'r'
            # 상태 (2, 0, L)로부터 입력 (0, 2)에 따른 상태 (2, 2, R)로의 전이
            elif m == 0 and c == 2:
                self.m = 2
                self.c = 2
                self.p = 'r'
            # 상태 (2, 0, L)로부터 입력 (1, 0)에 따른 상태 종결이 불가능한 경우
            elif m == 1 and c == 0:
                printf("식인종들이 선교사(들)을 잡아먹음!")
                return -1
            # 상태 (2, 0, L)로부터 입력 (1, 1)에 따른 상태 종결이 불가능한 경우
            elif m == 1 and c == 1:
                printf("식인종들이 선교사(들)을 잡아먹음!")
                return -1
            # 상태 (2, 0, L)로부터 입력 (2, 0)에 따른 상태 종결이 불가능한 경우
            elif m == 2 and c == 0:
                printf("식인종들이 선교사(들)을 잡아먹음!")
                return -1
        # 현재 상태가 (1, 1, L)일 때
        elif self.m == 1 and self.c == 1:
            # 상태 (1, 1, L)로부터 입력 (0, 1)에 따른 상태 종결이 불가능한 경우
            if m == 0 and c == 1:
                printf("식인종들이 선교사(들)을 잡아먹음!")
                return -1

```

```

# 상태 (1, 1, L)로부터 입력 (0, 2)에 따른 상태 종결이 불가능한 경우
elif m == 0 and c == 2:
    printf("식인종들이 선교사(들)을 잡아먹음!")
    return -1
# 상태 (1, 1, L)로부터 입력 (1, 0)에 따른 상태 (2, 1, R)로의 전이
elif m == 1 and c == 0:
    self.m = 2
    self.c = 1
    self.p = 'r'
# 상태 (1, 1, L)로부터 입력 (1, 1)에 따른 상태 (2, 2, R)로 전이
elif m == 1 and c == 1:
    self.m = 2
    self.c = 2
    self.p = 'r'
# 상태 (1, 1, L)로부터 입력 (2, 0)에 따른 상태 종결이 불가능한 경우
elif m == 2 and c == 0:
    printf("식인종들이 선교사(들)을 잡아먹음!")
    return -1
# 현재 상태가 (0, 2, L)일 때
elif self.m == 0 and self.c == 2:
    # 상태 (0, 2, L)로부터 입력 (0, 1)에 따른 상태 종결이 불가능한 경우
    if m == 0 and c == 1:
        printf("식인종들이 선교사(들)을 잡아먹음!")
        return -1
    # 상태 (0, 2, L)로부터 입력 (0, 2)에 따른 상태 (2, 2, R)로의 전이
    elif m == 0 and c == 2:
        self.m = 2
        self.c = 2
        self.p = 'r'
    # 상태 (0, 2, L)로부터 입력 (1, 0)에 따른 상태 종결이 불가능한 경우
    elif m == 1 and c == 0:
        printf("식인종들이 선교사(들)을 잡아먹음!")
        return -1
    # 상태 (0, 2, L)로부터 입력 (1, 1)에 따른 상태 종결이 불가능한 경우
    elif m == 1 and c == 1:
        printf("식인종들이 선교사(들)을 잡아먹음!")
        return -1
    # 상태 (0, 2, L)로부터 입력 (2, 0)에 따른 상태 (2, 2, R)로의 전이
    elif m == 2 and c == 0:
        self.m = 2
        self.c = 2
        self.p = 'r'
# 현재 상태가 (0, 1, L)일 때
elif self.m == 0 and self.c == 1:
    # 상태 (0, 1, L)로부터 입력 (0, 1)에 따른 상태 (0, 2, R)로의 전이
    if m == 0 and c == 1:
        self.m = 0
        self.c = 2
        self.p = 'r'
    # 상태 (0, 1, L)로부터 입력 (0, 2)에 따른 상태 종결이 불가능한 경우
    elif m == 0 and c == 2:
        printf("식인종들이 선교사(들)을 잡아먹음!")
        return -1
    # 상태 (0, 1, L)로부터 입력 (1, 0)에 따른 상태 (0, 2, R)로의 전이
    elif m == 1 and c == 0:
        self.m = 1
        self.c = 1
        self.p = 'r'
    # 상태 (0, 1, L)로부터 입력 (1, 1)에 따른 상태 종결이 불가능한 경우

```



```

        elif m == 1 and c == 1:
            printf("식인종들이 선교사(들)을 잡아먹음!")
            return -1
        # 상태 (0, 1, L)로부터 입력 (2, 0)에 따른 상태 (1, 1, R)로의 전이
        elif m == 2 and c == 0:
            self.m = 1
            self.c = 1
            self.p = 'r'
        # 현재 상태가 (0, 0, L)일 때
        elif self.m == 0 and self.c == 0:
            # 상태 (0, 0, L)로부터 입력 (0, 1)에 따른 상태 (0, 1, R)로의 전이
            if m == 0 and c == 1:
                self.m = 0
                self.c = 1
                self.p = 'r'
            # 상태 (0, 0, L)로부터 입력 (0, 2)에 따른 상태 (0, 2, R)로의 전이
            elif m == 0 and c == 2:
                self.m = 0
                self.c = 2
                self.p = 'r'
            # 상태 (0, 0, L)로부터 입력 (1, 0)에 따른 상태 (1, 0, R)로의 전이
            elif m == 1 and c == 0:
                self.m = 1
                self.c = 0
                self.p = 'r'
            # 상태 (0, 0, L)로부터 입력 (1, 1)에 따른 상태 (1, 1, R)로의 전이
            elif m == 1 and c == 1:
                self.m = 1
                self.c = 1
                self.p = 'r'
            # 상태 (0, 0, L)로부터 입력 (2, 0)에 따른 상태 (2, 0, R)로의 전이
            elif m == 2 and c == 0:
                self.m = 2
                self.c = 0
                self.p = 'r'

        return 0

res = 0
s = State()          # 클래스로 선언된 상태 객체를 선언
s.state_init()       # 시작 상태를 초기화
list = []            # 선교사와 식인종의 총 인원수를 위한 변수(사용자 입력)
# 인식 상태 (0, 0, L)에 도달하거나 종결이 불가능한 상태에 도달할 때까지 반복

while res == 0:
    printf("\n현재 상태 : (%d,%d,%c)\n" %(s.m, s.c, s.p) )
    # 사용자로부터 알파벳을 입력받음
    printf("\n승선 인원수를 입력하라. : ")
    list += stdin.readline().strip('\n').split()
    input_m = int(list.pop(0))
    input_c = int(list.pop(0))
    printf("입력 : (%d,%d)\n" %(input_m, input_c) )
    # 입력된 알파벳에 따라 전이하고 결과를 반환
    res = s.state_moving(input_m, input_c)

# 마지막 상태를 출력
printf("\n\n상태 : (%d,%d,%c)\n" %(s.m, s.c, s.p) )
stdin.readline()

```