ASSIGNMENT- SECOND YEAR STUDENTS

--SHIRSHAKK PURKAYASTHA

I.I.S.E.R. Bhopal

TEMPORAL DIFFERENCE LEARNING -ALGORITHM

Q-Learning CODE(with outputs):

```
import gym
import numpy as np

from gym.envs.registration import register
register(
    id='Deterministic-4x4-FrozenLake-v0', #new environment
    entry_point='gym.envs.toy_text.frozen_lake:FrozenLakeEnv',
    kwargs={'map_name': '4x4', 'is_slippery': False} # argument passed to the env
)
env = gym.make('Deterministic-4x4-FrozenLake-v0')
my_desk = [
    "GFFFF",
    "FFFFF",
    "FFFFG",
    "FFFFF",
    "FGFFG"
]


import gym

class CustomizedFrozenLake(gym.envs.toy_text.frozen_lake.FrozenLakeEnv):
    def __init__(self, **kwargs):
        super(CustomizedFrozenLake, self).__init__(**kwargs)
```

```
        for state in range(self.nS): # for all states
            for action in range(self.nA): # for all actions
                my_transitions = []
                for (prob, next_state, _, is_terminal) in self.P[state][action]:
                    row = next_state // self.ncol
                    col = next_state - row * self.ncol
                    tile_type = self.desc[row, col]
                    if tile_type == b'F':
                        reward = -1
                    elif tile_type == b'G':
                        reward = 10
                    #else:
                        #reward = 0

                    my_transitions.append((prob, next_state, reward, is_terminal))
                self.P[state][action] = my_transitions

from gym.envs.registration import register

register(
    id='Stochastic-5x5-FrozenLake-v0',
    entry_point='gym.envs.toy_text.frozen_lake:FrozenLakeEnv',
    kwargs={'desc': my_desk, 'is_slippery': False})
env = gym.make('Stochastic-5x5-FrozenLake-v0')
env.render()
```

OUTPUT:

```
GFFFF
FFFFF
FFFFG
FFFFF
FGFFG
```

```
env.reset()
env.render()

print("Action Space {}".format(env.action_space))
print("State Space {}".format(env.observation_space))
```

OUTPUT:

```
GFFFF
FFFFF
FFFFG
FFFFF
FGFFG
Action Space Discrete(4)
State Space Discrete(25)
```

```
"""ACTIONS DEFINED VIA:
    0 = SOUTH
    1 = NORTH
    2 = EAST
    3 = WEST
"""

state=env.s
if state in range (0,14):
  print("State:", state)
elif state in range (14,20):
  print("State:", state+1)
elif state in range (20,22):
  print("State:", state+2)

env.render()
```

```
State: 0
```

OUTPUT:

```
GFFFF
FFFFF
FFFFG
FFFFF
FGFFG
```

```
env.P[state][1]
```

OUTPUT:

```
[(1.0, 0, 0, True)]
```

```
m=int(input("Enter State numnber for start:"))

if m in range (0,14):
  env.s = m
elif m in range (14,20):
  env.s = m+1
elif m in range (20,22):
  env.s = m+2
 # set environment to illustration's state
env.render()

print("...................Learning Starts.......................")

epochs = 0
penalties, reward = 0, 0

frames = [] # for animation

done = False

while not done:
```

```
    action = env.action_space.sample()
    state, reward, done, info = env.step(action)

    if reward == -1:
        penalties += 1

    # Put each rendered frame into dict for animation
    frames.append({
        'frame' : env.render(mode='human'),
        'state': state,
        'action': action,
        'reward': reward
        }
    )

    epochs += 1


print("Timesteps taken: {}".format(epochs))
print("Penalties incurred: {}".format(penalties))
print("\n")
print(frames)
```

OUTPUT:

```
Enter State numnber for start:14


GFFFF
FFFFF
FFFFG
FFFFF
FGFFG
....................Learning Starts........................
  (Down)
GFFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GFFFF
FFFFF
FFFFG
FFFFF
FGFFG
Timesteps taken: 2
Penalties incurred: 0


[{'frame': None, 'state': 20, 'action': 1, 'reward': 0.0}, {'frame': None,
'state': 21, 'action': 2, 'reward': 1.0}]
```

```
from IPython.display import clear_output
from time import sleep
def print_frames(frames):
    for i, frame in enumerate(frames):
        clear_output(wait=True)
        print("frame: ",frame)
        print(f"Timestep: {i + 1}")
        #print(f"State: {frame['state']}")
```

```
        print(f"Action: {frame['action']}")
        print(f"Reward: {frame['reward']}")
        sleep(.1)

print_frames(frames)
```

OUTPUT:

```
frame:  {'frame': None, 'state': 24, 'action': 2, 'reward': 1.0}
Timestep: 1
Action: 2
Reward: 1.0
```

```
"""QLearning"""
import numpy as np
q_table = np.zeros([env.observation_space.n, env.action_space.n])
print(q_table)
```

OUTPUT:

```
[[0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]]
```

```
%%time
#Training the Agent

import random
from IPython.display import clear_output

# Hyperparameters
alpha = 0.1
gamma = 0.6
epsilon = 0.1

# For plotting metrics
all_epochs = []
all_penalties = []

for i in range(1, 10001):
    state = env.reset()
```

```
    epochs, penalties, reward, = 0, 0, 0
    done = False

    while not done:
        if random.uniform(0, 1) < epsilon:
            action = env.action_space.sample() # Explore action space
        else:
            action = np.argmax(q_table[state]) # Exploit learned values

        next_state, reward, done, info = env.step(action)

        old_value = q_table[state, action]
        next_max = np.max(q_table[next_state])

        new_value = (1 - alpha) * old_value + alpha * (reward + gamma * next_max)
        q_table[state, action] = new_value

        if reward == -1:
            penalties += 1

        state = next_state
        epochs += 1

    if i % 100 == 0:
        clear_output(wait=True)
        print(f"Episode: {i}")

print("Training finished.\n")
```

OUTPUT:

```
Episode: 100000
Training finished.

CPU times: user 7.5 s, sys: 2.33 s, total: 9.83 s
Wall time: 9.18 s
```

```
m=int(input(print("Enter state value:")))

for m in range (0,14):
  print(q_table[m])
for m in range (14,20):
  print(q_table[(m+1)])
for m in range (20,22):
  print(q_table[(m+2)])
```

OUTPUT:

```
Enter state value:
20
[0. 0. 0. 0.]
[1.   0.36 0.36 0.6 ]
[0.6        0.21599983 0.21599977 0.35999992]
[0.35999997 0.03954002 0.01053266 0.0582606 ]
[0.08836117 0.         0.         0.        ]
[0.59999976 0.35999989 0.35999997 1.        ]
[0.6        0.2159974  0.21599977 0.59999917]
[0.36       0.02348849 0.02607496 0.0684     ]
[0.16088968 0.         0.         0.        ]
[0. 0. 0. 0.]
[0.26911731 0.09340281 0.10309805 0.6       ]
```

```
[0.35999985 0.00175724 0.02009013 0.036     ]
[0.15117515 0.         0.         0.        ]
[0.00815816 0.         0.         0.        ]
[0.01549376 0.         0.         0.29322828]
[0.03047873 0.         0.         0.        ]
[0. 0. 0. 0.]
[0. 0. 0. 0.]
[0. 0. 0. 0.]
[0. 0. 0. 0.]
[0. 0. 0. 0.]
[0. 0. 0. 0.]
'q_table[20]\n'
```

```
q_table[20] #Checking Qvalue for any random state
```

OUTPUT:

```
array([0., 0., 0., 0.])
```

```
"""Evaluating Agent's performance after Q-learning"""

total_epochs, total_penalties = 0, 0
episodes = 520148

for _ in range(episodes):
    state = env.reset()
    epochs, penalties, reward = 0, 0, 0

    done = False

    while not done:
        action = np.argmax(q_table[state])
        state, reward, done, info = env.step(action)

        if reward == -1:
            penalties += 1

        epochs += 1

    total_penalties += penalties
    total_epochs += epochs

print(f"Results after {episodes} episodes:")
print(f"Average timesteps per episode: {total_epochs / episodes}")
print(f"Average penalties per episode: {total_penalties / episodes}")
```

OUTPUT:

```
Results after 520148 episodes:
Average timesteps per episode: 1.0
Average penalties per episode: 0.0
```

****

SARSA CODE(with outputs):

```
import gym

import numpy as np

from gym.envs.registration import register

register(

    id='Deterministic-4x4-FrozenLake-v0', # name given to this new environment

    entry_point='gym.envs.toy_text.frozen_lake:FrozenLakeEnv', # env entry point

    kwargs={'map_name': '4x4', 'is_slippery': False} # argument passed to the env

)

"""We specify the start state at 1..

This can be reconfiguired as per our requirements """

env = gym.make('Deterministic-4x4-FrozenLake-v0') # load the environment

my_desk = [

    "GSFFF",

    "FFFFF",

    "FFFFG",

    "FFFFF",

    "FGFFG"

]
```

```
import gym

class CustomizedFrozenLake(gym.envs.toy_text.frozen_lake.FrozenLakeEnv):

    def __init__(self, **kwargs):

        super(CustomizedFrozenLake, self).__init__(**kwargs)


        for state in range(self.nS): # for all states

            for action in range(self.nA): # for all actions

                my_transitions = []

                for (prob, next_state, _, is_terminal) in self.P[state][action]:

                    row = next_state // self.ncol
```

```
                    col = next_state - row * self.ncol
                    tile_type = self.desc[row, col]
                    if tile_type == b'F':
                        reward = -1
                    elif tile_type == b'G':
                        reward = 10
                    else:
                        reward = 0


                    my_transitions.append((prob, next_state, reward, is_terminal))
                self.P[state][action] = my_transitions


from gym.envs.registration import register


register(
    id='Stochastic-5x5-FrozenLake-v0',
    entry_point='gym.envs.toy_text.frozen_lake:FrozenLakeEnv',
    kwargs={'desc': my_desk, 'is_slippery': False})
env = gym.make('Stochastic-5x5-FrozenLake-v0')
env.render()
print(env.action_space.n)
print(env.observation_space.n)
```

OUTPUT:

```
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
4
25
```

```
#Parameters
epsilon = 0.9
total_episodes = 5500
max_steps = 100
alpha = 0.70
```

```
gamma = 0.75
```

```
#Initializing the Q-matrix
Q = np.zeros((env.observation_space.n, env.action_space.n))
#print(Q)

#Function to choose the next action
def choose_action(state):
  action=0
  if np.random.uniform(0, 1) < epsilon:
    action = env.action_space.sample()
  else:
    action = np.argmax(Q[state, :])
  return action

#Function to learn the Q-value
def update(state, state2, reward, action, action2):
  predict = Q[state, action]
  target = reward + gamma * Q[state2, action2]
  Q[state, action] = Q[state, action] + alpha * (target - predict)

#print(Q)
```

```
#Initializing the reward
reward=0

# Starting the SARSA learning
for episode in range(total_episodes):
  t = 0
  state1 = env.reset()
  action1 = choose_action(state1)

  while t < max_steps:
    #Visualizing the training
```

```
    env.render()

    #Getting the next state
    state2, reward, done, info = env.step(action1)

    #Choosing the next action
    action2 = choose_action(state2)

    #Learning the Q-value
    update(state1, state2, reward, action1, action2)

    state1 = state2
    action1 = action2

    #Updating the respective vaLues
    t += 1
    reward += -1

    #If at the end of learning process
    if done:
      break
```

OUTPUT:

```
Streaming output truncated to the last 5000 lines.
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
```

```
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
```

```
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
```

```
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
```

```
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
```

```
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
```

```
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
```

```
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
```

```
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
```

```
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
```

```
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
```

```
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
```

```
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
```

```
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
```

```
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
```

```
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
```

```
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
```

```
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
```

```
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
```

```
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
```

```
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
```

```
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
```

```
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
```

```
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
```

```
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
```

```
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
```

```
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
```

```
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
```

```
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
```

```
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
```

```
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
```

```
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
```

```
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
```

```
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
```

```
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
```

```
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
```

```
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
```

```
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
```

```
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
```

```
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
```

```
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
```

```
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
```

```
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
```

```
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
```

```
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
```

```
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
```

```
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
```

```
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
```

```
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
```

```
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
```

```
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
```

```
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
```

```
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
```

FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG

```
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
```

FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF

```
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
```

```
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
```

```
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
```

```
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
```

```
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
```

```
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
```

```
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
```

```
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
```

```
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
```

```
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
```

FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF

```
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
```

```
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
```

```
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
```

```
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
```

```
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
```

```
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
```

```
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
```

FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)

```
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
```

```
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
```

```
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
```

```
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
```

FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF

```
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
```

```
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
```

```
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
```

```
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
```

```
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Down)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
```

```
FFFFG
FFFFF
FGFFG
  (Right)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Up)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
  (Left)
GSFFF
FFFFF
FFFFG
FFFFF
FGFFG

GSFFF
FFFFF
FFFFG
FFFFF
FGFFG
```

```
#Evaluating the performance

print ("Performace : ", reward/total_episodes)

#Visualizing the Q-matrix

print(Q)

"A positive performance is highly acceptable, given that with every step a penalty
 of -1 is incured"
```

OUTPUT:

```
Performace :  0.0
[[0.         0.         0.         0.        ]
 [1.         0.13274721 0.07648872 0.56953974]
 [0.54072524 0.12122495 0.15814572 0.0489783 ]
 [0.07776374 0.19327976 0.4114731  0.30618066]
 [0.08640925 0.3275145  0.17826196 0.08731207]
 [0.39577389 0.11300707 0.09764022 1.        ]
 [0.12497262 0.36135288 0.09676008 0.19409453]
 [0.13187797 0.08586729 0.36476933 0.07487011]
 [0.10040456 0.70236679 0.74703373 0.09673464]
 [0.2501247  1.         0.69773896 0.11994224]
 [0.13446335 0.20802134 0.18088891 0.56541349]
 [0.23756802 0.2693264  0.06477107 0.20374487]
 [0.05947002 0.08085113 0.08264882 0.12529833]
 [0.07471843 0.37871433 1.         0.41217226]
 [0.         0.         0.         0.        ]
```

```
[0.42556701 0.63598816 0.59889383 0.32766487]
[0.24501957 1.         0.30190637 0.11578603]
[0.10318084 0.21278995 0.15782044 0.09638   ]
[0.36341704 0.58674952 0.70785036 0.2871282 ]
[0.13864157 1.         0.605381   1.        ]
[0.58651346 0.49714581 1.         0.44345222]
[0.         0.         0.         0.        ]
[1.         0.28576564 0.3188376  0.09962918]
[0.58024371 0.26075519 1.         0.41009241]
[0.         0.         0.         0.        ]]
```

❖ **QUESTION:**

Conceptual Question:

Difference between On-Policy and Off-Policy:

On-Policy TD Control Algorithm: SARSA (State-Action-Reward-State-Action)

- In SARSA, the agent learns the optimal policy and behaves also per the same greedy policy.
- Update policy is same as the Behaviour Policy.

Off-Policy TD Control Algorithm: Q-Learning

- IN Q-Learning, the agent learns the optimal policy using the absolute Greedy Policy but behaves according to other greedy policies.
- Update Policy is different than the Behaviour Policy.

****