1)main.py

```
# This entrypoint file to be used in development. Start by reading README.md
from RPS_game import play, mrugesh, abbey, quincy, kris, human, random_player
from RPS import player
from unittest import main

play(player, quincy, 1000)
play(player, abbey, 1000)
play(player, kris, 1000)
play(player, mrugesh, 1000)

# Uncomment line below to play interactively against a bot:
#play(human, abbey, 20, verbose=True)

# Uncomment line below to play against a bot that plays randomly:
#play(human, random_player, 1000)


# Uncomment line below to run unit tests automatically
main(module='test_module', exit=False)
```

2)RPS.py

```
def player(prev_play, opponent_history=[]):
    if prev_play:
        opponent_history.append(prev_play)

    # Step 1: Handle first move
    if not opponent_history:
        return 'R'  # Start with Rock
```

```
    # Step 2: Counter Quincy (predictable pattern "RRPPSS")
    quincy_cycle = ["R", "R", "P", "P", "S"]
    if len(opponent_history) % len(quincy_cycle) == 0:
        return "P"  # Paper beats Rock

    # Step 3: Counter Kris (mirrors previous moves)
    if len(opponent_history) >= 1:
        return {'R': 'P', 'P': 'S', 'S': 'R'}[opponent_history[-1]]  # Beat the
mirrored move

    # Step 4: Counter Mrugesh (most frequent move tracking)
    last_ten = opponent_history[-10:]
    if last_ten:
        most_common = max(set(last_ten), key=last_ten.count)
        counter_move = {'R': 'P', 'P': 'S', 'S': 'R'}[most_common]
        return counter_move

    # Step 5: Counter Abbey (pattern-based prediction)
    if len(opponent_history) > 2:
        last_two = opponent_history[-2] + opponent_history[-1]
        abbey_counter = {
            "RR": "P", "RP": "S", "RS": "R",
            "PR": "S", "PP": "R", "PS": "P",
```

```
        "SR": "R", "SP": "P", "SS": "S"
    }
    return abbey_counter.get(last_two, 'R')

# Step 6: Default rotation strategy to stay unpredictable
rotation = ['R', 'P', 'S']
return rotation[len(opponent_history) % 3]
```