```python
# Airport security Simulation

# Goal - Average Wait - 15 min or less

# Steps:
# arrive at airport; mean rate = 0.2 minute
# get through ID/boarding-pass queue; mean rate = 0.75 minutes
# personal scanner; mean rate = (0.5-1) minutes

# Start defining simulation

import simpy
import random
import statistics

wait_times = []


class Airport(object):
    def __init__(self, env, num_ID_queue, num_personal_check_queue):
        self.env = env
        self.ID_queue = simpy.Resource(env, num_ID_queue)
        self.personal_check_queue = simpy.Resource(env, num_personal_check_queue)


    def boarding_pass_queue(self, passengers): # personal scanner; mean rate = (0.5-1) minutes given
                                                    # this will trigger an event after the timeout period.
        yield self.env.timeout(random.uniform(0.5, 1))

    def personal_check_queue(self, passengers): # ID/boarding-pass queue; mean rate = 0.75 minutes given
        yield self.env.timeout(0.75)


def arrive_at_airport(env, passengers, airport):
    # arrives at airport
    arrival_time = env.now

    # go through ID/boarding pass queue
    with airport.ID_queue.request() as request:
        yield request
        yield env.process(airport.boarding_pass_queue(passengers))


    wait_times.append(env.now - arrival_time)

# define a function that runs the simulation.

def run_airport(env, num_ID_queue, num_personal_check_queue):
    airport = Airport(env, num_ID_queue, num_personal_check_queue)

    for passengers in range(1):
        env.process(arrive_at_airport(env, passengers, airport))

    while True:
        yield env.timeout(0.20) # 5 new arrivals per minute

        passengers += 1
        env.process(arrive_at_airport(env, passengers, airport))

def calculate_wait_times(wait_times):
    average_wait = statistics.mean(wait_times)

    # pretty print results:
    minutes, frac_minutes = divmod(average_wait, 1)
    seconds = frac_minutes * 60
    return round(minutes), round(seconds)

# Create input fields at runtime for the user
def get_user_input():
    num_ID_queue = input("Input # of ID_queues working: ")
    num_personal_check_queue = input("Input # of personal_check_queues working: ")


    params = [num_ID_queue, num_personal_check_queue]

    if all(str(i).isdigit() for i in params):
        params = [int(x) for x in params]
    else:
        print(
            "Could not parse input. The simulation will use default values",
            "\n1 ID_queue, 1 personal_check_queue.",
        )
        params = [1, 1, 1]
    return params

def main():
    # Setup
    random.seed(42)
    num_ID_queue, num_personal_check_queue = get_user_input()

    # Run the simulation
    env = simpy.Environment()
    env.process(run_airport(env, num_ID_queue, num_personal_check_queue))
    env.run(until=90)

    # View the results
    mins, secs = calculate_wait_times(wait_times)
    print(
        "Running simulation...",
        f"\nThe average wait time is {mins} minutes and {secs} seconds.",
    )

if __name__ == '__main__':
    # Possible Solution:
    # 10 ID_queues
    # 6 personal_check_queues

    main()
```