Code for Round Robin Scheduling :-

```
# Python3 program for implementation of RR scheduling

# Function to find the waiting time for all processes
def findWaitingTime(processes, n, bt, wt, quantum):
	rem_bt = [0] * n

	# Copy the burst time into rt[]
	for i in range(n):
		rem_bt[i] = bt[i]
	t = 0 # Current time

	# Keep traversing processes in round robin manner until all of them are not done.
	while(1):
		done = True

		# Traverse all processes one by one repeatedly
		for i in range(n):

		# If burst time of a process is greater than 0 then only need to process further
			if (rem_bt[i] > 0) :
				done = False # There is a pending process

				if (rem_bt[i] > quantum) :

				# Increase the value of t i.e. shows how much time a process has
				  been processed
					t += quantum

					# Decrease the burst_time of current
					# process by quantum
					rem_bt[i] -= quantum

		# If burst time is smaller than or equal to quantum. Last cycle for this process
				else:

					# Increase the value of t i.e. shows how much time
					# a process has been processed
					t = t + rem_bt[i]

					# Waiting time is current time minus time used
					# by this process
					wt[i] = t - bt[i]

					# As the process gets fully executed
					# make its remaining burst time = 0
					rem_bt[i] = 0

		# If all processes are done
		if (done == True):
```

```
                        break

# Function to calculate turn around time
def findTurnAroundTime(processes, n, bt, wt, tat):

        # Calculating turnaround time
        for i in range(n):
                tat[i] = bt[i] + wt[i]


# Function to calculate average waiting and turn-around times.
def findavgTime(processes, n, bt, quantum):
        wt = [0] * n
        tat = [0] * n

        # Function to find waiting time of all processes
        findWaitingTime(processes, n, bt, wt, quantum)

        # Function to find turn around time for all processes
        findTurnAroundTime(processes, n, bt, wt, tat)

        # Display processes along with all details
        print("Processes Burst Time  Waiting","Time Turn-Around Time")
        total_wt = 0
        total_tat = 0
        for i in range(n):

                total_wt = total_wt + wt[i]
                total_tat = total_tat + tat[i]
                print(" ", i + 1, "\t\t", bt[i], "\t\t", wt[i], "\t\t", tat[i])

        print("\nAverage waiting time = %.5f "%(total_wt /n) )
        print("Average turn around time = %.5f "% (total_tat / n))

# Driver code
if __name__ =="__main__":

        # Process id's
        proc = [1, 2, 3]
        n = 3

        # Burst time of all processes
        burst_time = [10, 5, 8]

        # Time quantum
        quantum = 2;
        findavgTime(proc, n, burst_time, quantum)
```