Task 2 Analysis Report **Regina So (62104047)**

The scheme I implemented, or have attempted to implement, works as follow:

First, the user will be prompted to input the fixed utilization value (U), and the number of tasks (n) involved. Seeing as the utilization should be 1 at max, utilization is stored as a double. The program then proceeds to generate a random number between 0 and 1 (more precisely, it should've been the range (0,1], as there wouldn't be any tasks with 0 utilization (then the task would not have existed)). The first random guess will be saved into an array and be considered the utilization of the first task (U_1) within the task set with n number of tasks. The next random value will be taken from 0 to U-U_1, then proceed to be stored onto the array as U_2. This will continue on until the n-1 task. Each utilization are kept on record and a sum of the total is available to calculate the final task, which will simply be U-sum. After several tests, it appears the values generated are quite random (rand is fed with current calendar time using time() to seed and initialize the rand() generator (srand).