```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/mman.h>
#include <sys/shm.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <sys/wait.h>
#include <string.h>
#include <time.h>
///////////////gcc -o m multiprocess_1.c -lrt
float grade1A[] = {2.5,4,4,4,2.5,2.5};//15,6
float weight1A[] = {3,3,2,1,3,3,3};
float sum1A=0;

float grade1B[] = {4,3,2.5,3.5,4,4,3.5,4};//20,8
float weight1B[] = {3,3,3,1,1,3,3,3};
float sum1B=0;

float grade2A[] = {4,2.5,3,4,3.5,4,4,3}; //20,8
float weight2A[] = {3,3,3,3,1,3,1,3};
float sum2A=0;

float grade2B[] = {4,4,3.5,4,3.5,3.5,2,2,2.5}; //21,9
float weight2B[] = {3,1,3,1,1,3,3,3,3};
float sum2B=0;

float grade3A[] = {4,4,4,4,4,4,4,4,4}; //21,9
float weight3A[] = {3,1,3,3,1,3,3,3,1};
float sum3A=0;
float gradeAll=0;

void delay(int num){
        int milli_sec = 1000*num;
        clock_t start_time = clock();
        while(clock() < start_time + milli_sec);
}
void DoWorkInChild(){
        int i,j;
        for(i=0; i<6; i++){ //15,6
                sum1A+=grade1A[i]*weight1A[i];
        }
        sum1A = sum1A/15;

        for(i=0; i<8; i++){ //20,8
                sum1B+=grade1B[i]*weight1B[i];
```

```
47          }
48          sum1B = sum1B/20;
49
50          for(i=0; i<8; i++){ //20,8
51                  sum2A+=grade2A[i]*weight2A[i];
52          }
53          sum2A = sum2A/20;
54
55          for(i=0; i<9; i++){ //21,9
56                  sum2B+=grade2B[i]*weight2B[i];
57          }
58          sum2B = sum2B/21;
59
60          for(i=0; i<9; i++){ //21,9
61                  sum3A+=grade3A[i]*weight3A[i];
62          }
63          sum3A = sum3A/21;
64
65          delay(2000);
66  }
67  void DoWorkInParent(){
68          int i;
69
70          printf("Grade AVG: %.2f \n",gradeAll);
71          delay(2000);
72  }
73  void main (){
74          pid_t pids[10];
75          int i;
76          int n = 1;
77          for (i = 0; i < n; ++i) {
78                  pids[i] = fork();
79                  if (pids[i] < 0) {
80                          perror("fork");
81                          abort();
82                  } else {
83                          if (pids[i] == 0) {
84                                  DoWorkInChild();
85                                  delay(2000);
86                                  exit(0);
87                          }
88                  }
89          }
90      DoWorkInParent();
91      delay(2000);
```