First:_____ Last:_____ Instructor (Circle):**JV MT RY AC**

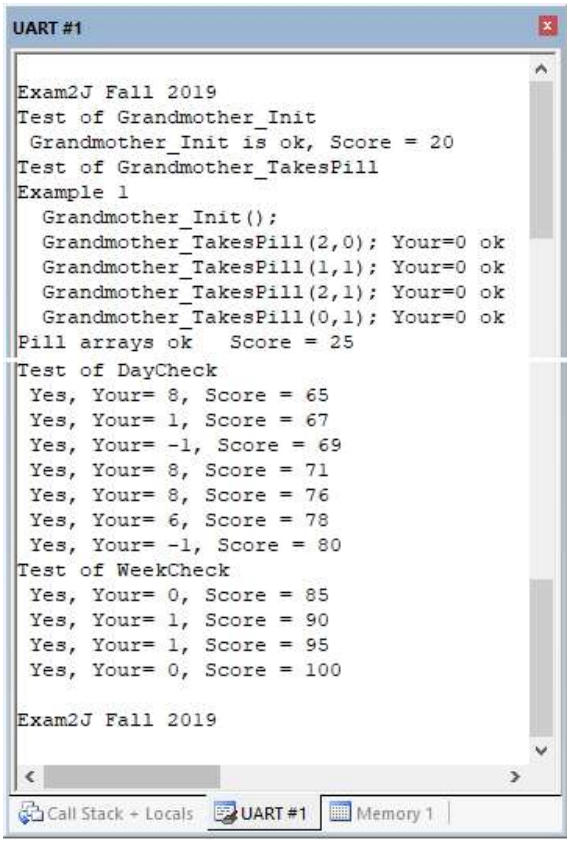
Survey: Have you had any formal programming instruction (like high school comp sci) other than EE306, EE319K or EE312? _____yes / no _____

Scoring The correct output values are shown in the figure on the right. Your grade will be based both on the numerical results returned by your program and on your programming style. In particular, write code that is easy to understand, easy to debug, easy to change. Please employ good labels, pretty structure, and good comments.

Performance Score= Run by TA		TA:

I promise to follow these rules

This is a closed book exam. You must develop the software solution using the **Keil uVision** simulator. **You have about 60 minutes of programming**, so allocate your time accordingly. You must bring a laptop and are allowed to bring only some pens and pencils (no books, cell phones, hats, disks, CDs, or notes). Each person works alone (no groups). You have full access to **Keil uVision**, with the **Keil uVision** help. You may use the Window's calculator. You sit in front of a computer and edit-build-run-debug the programming assignment on the simulator. You do NOT have access the book, internet or manuals. You may not access your network drive or the internet. You are not allowed to discuss this exam with other EE319K students until Saturday.



The following activities occurring during the exam will be considered scholastic dishonesty:

1) running any program from the PC other than **Keil uVision**, or the Window's calculator,
2) communicating with **anyone else** except for the instructors **by any means** about this exam until Saturday.
3) using material/equipment other than a pen/pencil,
4) hard-coding so it outputs answers that give points without actually solving the problem,
5) modifying anything other than **Exam2CPart.c and Exam2ASMPart.s**

Students caught cheating will be turned to the Dean of Students. Your signature is your promise that you have not cheated and will not cheat on this exam, nor will you help others to cheat on this exam:

Signed: _____ November 2019

Procedure

First, you will log onto the computer and download files from the web as instructed by the TAs.

Web site: http://users.ece.utexas.edu/~valvano/Volume1/Exam2J/

User: ***4C123*** Password: ***7799***

UNZIP the folder placing it **ON THE DESKTOP**. Within **Keil uVision** open the project, put your name on the first comment line of the two files **Exam2CPart.c** and **Exam2ASMPart.s**. Before writing any code, please build and run the system. You should get output like the figure above (but a much lower score). You may create backup versions of your program. If you wish to roll back to a previous version, simply open one of the backup versions. Some of the test cases are listed below

(20) Question 1 **Grandmother_Init()**

(20) Question 2 **Grandmother_TakesPill**

```
;Example 1
  Grandmother_Init();
  Grandmother_TakesPill(2,0); // should return 0, takes pill 2 on day 0
  Grandmother_TakesPill(1,1); // should return 0, takes pill 1 on day 1
  Grandmother_TakesPill(2,1); // should return 0, takes pill 2 on day 1
  Grandmother_TakesPill(0,1); // should return 0, takes pill 0 on day 1
; Pill0 should be {0,1,0,0,0,0,0} // took on day 1
; Pill1 should be {0,1,0,0,0,0,0} // took on day 1
; Pill2 should be {1,1,0,0,0,0,0} // took on days 0,1
;Example 2
  Grandmother_Init();
  Grandmother_TakesPill(0,0); // should return 0, takes pill 0 on day 0
  Grandmother_TakesPill(1,0); // should return 0, takes pill 1 on day 0
  Grandmother_TakesPill(2,1); // should return 0, takes pill 2 on day 1
  Grandmother_TakesPill(1,2); // should return 0, takes pill 1 on day 2
  Grandmother_TakesPill(0,3); // should return 0, takes pill 0 on day 3
  Grandmother_TakesPill(2,4); // should return 0, takes pill 2 on day 4
  Grandmother_TakesPill(2,5); // should return 0, takes pill 2 on day 5
  Grandmother_TakesPill(1,6); // should return 0, takes pill 1 on day 6
  Grandmother_TakesPill(0,6); // should return 0, takes pill 0 on day 6
; Pill0 should be {1,0,0,1,0,0,1} // took on days 0,3,6
; Pill1 should be {1,0,1,0,0,0,1} // took on days 0,2,6
; Pill2 should be {0,1,0,0,1,1,0} // took on days 1,4,5
;Example 3
  Grandmother_Init();
  Grandmother_TakesPill(2,4); // returns 0 (first taking pill 2 on day 4 is ok)
  Grandmother_TakesPill(2,4); // returns +1 (second taking pill 2 on day 4 is bad)
; Pill0 should be {0,0,0,0,0,0,0} // took none
; Pill1 should be {0,0,0,0,0,0,0} // took none
; Pill2 should be {0,0,0,0,2,0,0} // took two pill 2 on day 4
;Example 4
  Grandmother_Init();
  Grandmother_TakesPill(3,0); // should return -1 because pill=3 is illegal
  Grandmother_TakesPill(1,7); // should return -1 because day=7 is illegal
; Pill0 should be {0,0,0,0,0,0,0} // took none
; Pill1 should be {0,0,0,0,0,0,0} // took none
; Pill2 should be {0,0,0,0,0,0,0} // took none
```

(20) Question 3 **Grandmother_NeedsPill** *examples 5-8 use a correct version of Grandmother_TakesPill and use your version of Grandmother_NeedsPill*

```
;Example 5
  Grandmother_Init();
  Grandmother_TakesPill(0,0);
  Grandmother_TakesPill(1,0);
  Grandmother_TakesPill(2,0);
  Grandmother_TakesPill(1,1);
  Grandmother_TakesPill(0,1);
  Grandmother_TakesPill(2,1);
  Grandmother_TakesPill(2,2);
  Grandmother_TakesPill(1,2);
  Grandmother_TakesPill(0,2);
  Grandmother_NeedsPill(1); // should return 3, completed all 3 pills on day 1
  Grandmother_NeedsPill(2); // should return 3, completed all 3 pills on day 2
  Grandmother_NeedsPill(0); // should return 3, completed all 3 pills on day 0
;Example 6
  Grandmother_Init();
  Grandmother_TakesPill(2,6);
  Grandmother_TakesPill(0,6);
  Grandmother_NeedsPill(6); // should return 1 because needed pill 1
;Example 7
  Grandmother_Init();
  Grandmother_TakesPill(0,3);
  Grandmother_TakesPill(1,3);
  Grandmother_TakesPill(2,3);
  Grandmother_TakesPill(1,4);
  Grandmother_TakesPill(0,4);
  Grandmother_TakesPill(2,4);
  Grandmother_TakesPill(2,5);
  Grandmother_TakesPill(1,5);
  Grandmother_TakesPill(0,5);
  Grandmother_TakesPill(2,6);
  Grandmother_TakesPill(1,6);
  Grandmother_TakesPill(0,6);
  Grandmother_NeedsPill(4); // should return 3, completed all 3 pills on day 4
  Grandmother_NeedsPill(5); // should return 3, completed all 3 pills on day 5
  Grandmother_NeedsPill(6); // should return 3, completed all 3 pills on day 6
  Grandmother_NeedsPill(3); // should return 3, completed all 3 pills on day 3
;Example 8
   Grandmother_Init();
   Grandmother_TakesPill(0,3);
   Grandmother_TakesPill(2,3);
   Grandmother_TakesPill(1,4);
   Grandmother_TakesPill(0,4);
   Grandmother_TakesPill(2,5);
   Grandmother_TakesPill(1,5);
   Grandmother_TakesPill(2,6);
   Grandmother_TakesPill(1,6);
   Grandmother_NeedsPill(4); // should return 2, needs pill 2 on day 4
   Grandmother_NeedsPill(5); // should return 0, needs pill 0 on day 5
   Grandmother_NeedsPill(6); // should return 0, needs pill 0 on day 6
   Grandmother_NeedsPill(3); // should return 1, needs pill 1 on day 3
```

(20) Question 4 **DayCheck**

```
Inputs                                  Expected return value
 aDay={5,{3,0,4,4,3},{3,0,4,4,3};       8 (all pills taken as prescribed)
 aDay={3,{1,1,2},    {1,2,2};           1 (too few pill 1)
 aDay={3,{1,2,3},    {1,2,2};           -1 (too many pill 2)
 aDay={0,{ },        { };               8 (no pills needed)
```

(20) Question 5 **WeekCheck**

```
Inputs                                  Expected return value
 aWeek={{5,{3,1,4,4,3},{3,1,4,4,3}},
        {4,{3,2,4,4}  ,{3,2,4,4}},
        {5,{3,1,4,4,3},{3,1,4,4,3}},
        {4,{3,2,4,4}  ,{3,2,4,4}},
        {5,{3,0,4,4,3},{3,0,4,4,3}},
        {4,{3,1,4,4}  ,{3,1,4,4}},
        {3,{3,4,4}    ,{3,4,4}}};     0 and status={0,0,0,0,0,0,0}(all taken ok)
 aWeek={{1,{2},       ,{2}},
        {0,{}         ,{}},
        {3,{3,1,4}    ,{3,1,3}},      too few on day 2
        {4,{3,2,4,4}  ,{3,3,4,4}},    too many on day 3
        {1,{2}        ,{2}},
        {0,{}         ,{}},
        {1,{2}        ,{2}}};    1 and status={0,0,1,1,0,0,0} (days 2 3 improper}
 aWeek={{1,{2},       ,{2}},
        {3,{3,1,4}    ,{5,1,3}},  too many and too few on day 1
        {0,{}         ,{}},
        {1,{2}        ,{2}},
        {0,{}         ,{}},
        {4,{3,2,4,4}  ,{3,3,4,4}}, too many on day 5
        {1,{2}        ,{2}}};    1 and status={0,1,0,0,0,1,0} (days 1 5 improper)
```

Our main program will call your functions multiple times, and will give your solution a performance score of 0 to 100. *You should not modify our main program or our example data.* Each time you add a block of code, you should run our main program, which will output the results to the **`UART#1`** window. After you are finished, raise your hand and wait for a TA. The TA will direct you on how to complete the submission formalities. The TA will run your program in front of you and record your performance score on your exam cover sheet. The scoring page will not be returned to you.

Important Notes:

- Your functions should work for all cases given to it by the grader.
- The description of functions can be found in the exam comments.
- ***The three assembly functions are related.***
- ***The two C functions are related.***
- ***Since the assembly functions are not related to the C functions, you can answer assembly part and C part in either order.***

You may not use any functions from any C libraries other than **stdint.h**.

Submission Guidelines:

- Log onto Canvas and submit your **Exam2CPart.c** and **Exam2ASMPart.s** source files into the Exam2 submission link. Be careful because only one submission will be allowed.