

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

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		TA:
Score=		
Run by 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 75 minutes**, 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:

```

UART #1
Exam2StringConversions
***Q1) Test of Octal2String***
Yes, 0 is "0", Score = 2
Yes, 66 is "102", Score = 4
Yes, 123 is "173", Score = 6
Yes, 5123 is "12003", Score = 8
Yes, 16777215 is "77777777", Score = 15
Yes, 16777216 is "", Score = 15
Testing 15 randomly selected valid n
***Q2) Test of String2Octal***
Yes, "5" is 5, Score = 31
Yes, "12" is 10, Score = 32
Yes, "371" is 249, Score = 33
Yes, "05421" is 2833, Score = 34
Yes, "177777" is 65535, Score = 37
Yes, "200000" is 4294967295, Score = 47
Yes, "84" is 4294967295, Score = 47
Yes, "3,1" is 4294967295, Score = 5
Yes, "" is 0, Score = 57
Testing 13 randomly selected valid s
***Q3) Test of AssignStudents2LabTab
Test 0, return ok, Lab seating corre
Test 1, return ok, Lab seating corre
Test 2, return ok, Score = 83
Test 3, return ok, Score = 85
Test 4, return ok, Lab seating corre
Test 5, return ok, Lab seating corre
Test 6, return ok, Lab seating corre
Test 7, return ok, Score = 100
End of Exam2StringConversions
  
```

Signed: _____ April 2018

Valvano, Tiwari, Yerraballi Spring 2018 Exam2

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/Exam2SC/>

User: tvy Password: 999

UNZIP the folder placing it **ON THE DESKTOP**. You are not allowed to archive this exam. 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.

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 here is less detailed than that in the comments above the function in the source files. Refer to them before attempting your solution.
- *In this exam you are being asked to write one routine in assembly and two in C. All routines are INDEPENDENT, you can answer them in any order you want.*

The exam has three parts a) through c), details of which are given in the starter code (**Exam2CPart.c** and **Exam2AsmPart.s**).

Let $b_8, b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0$ be the digits of a 9-bit binary number

The digits of the 3-digit octal number are simply found by grouping 3 binary bits per octal digit

$$d_2 = 4*b_8 + 2*b_7 + b_6 \quad \text{where each octal digit is 0 to 7}$$

$$d_1 = 4*b_5 + 2*b_4 + b_3$$

$$d_0 = 4*b_2 + 2*b_1 + b_0$$

The value of the octal number is $64*d_2 + 8*d_1 + d_0$

In general the value of an octal number is the sum of $8^i * d_i$, where each d_i is 0 to 7

Octal is similar to hex, the value of a hex number is the sum of $16^i * d_i$, where each d_i is 0 to 15

(30 pts) Part a) Write a C function called **Octal2String**, that converts an integer to a null-terminated ASCII string in octal. For example, if the input **n** is 20 then the **retString** has “24”, and the return parameter is 0 (success). You must add the null character `\0` to the end of the string. Valid inputs will convert to less than or equal to 8 octal digits. If the input is greater than 8 digits (e.g., **n > 16777215**) then set the **retString** to “*”, and the return parameter should be 1 (error).

```
int Octal2String( uint32_t n, char retString[ ] );
```

(40 pts) Part b) Write an assembly subroutine called **String2Octal**, that takes as input, the

address of a null-terminated ASCII string that encodes an octal number, and returns the unsigned integer conversion of the string. Octal is base 8 and an octal digit ranges from '0' to '7'. For example, if the input string is "24" then the returned integer value is 20. If there is an error or invalid input, then return a 0xFFFFFFFF. The C prototype for the function is given below; you will write the subroutine in assembly.

```
uint32_t String2Octal(char *input);
```

(30 pts) Part c) Write a function in C called **AssignStudents2LabTables** that assigns students in the class to lab tables based on their groupID.

```
int AssignStudents2LabTables(Student_t const EE319K[],
                             LabTable_t lab[],
                             uint8_t const size);
```

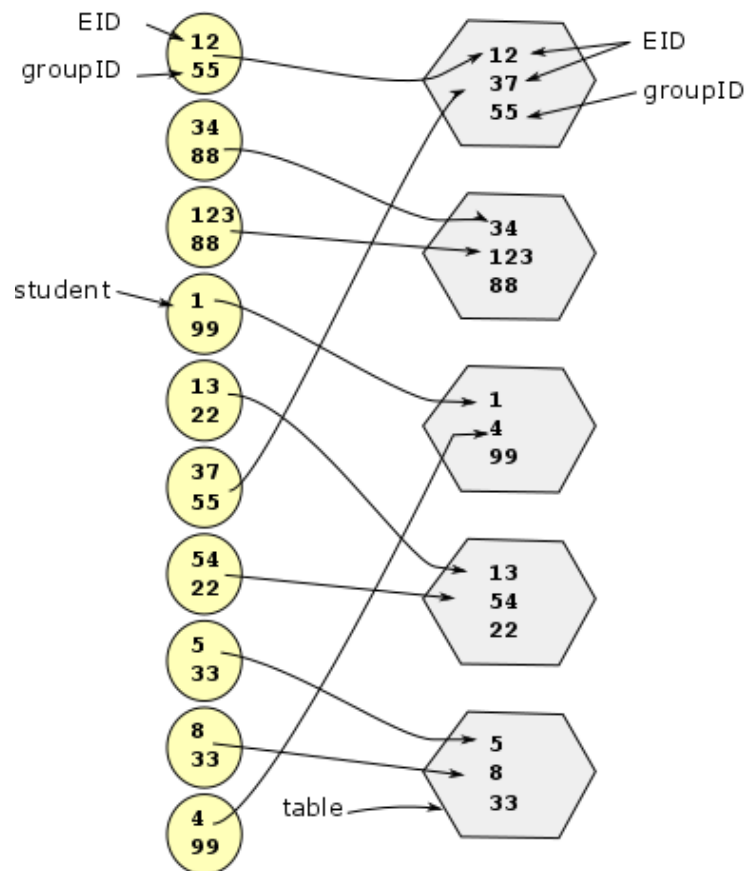
Consider this case with 10 students in **EE319K**. Each student has an EID and a groupID. The left hand side shows the input to your function, and the right side shows one possible correct solution.

Students 12 and 37 are in the same group because they have the same groupID (55). A valid input has an even number of students, and each student has exactly one lab partner. The index into the **lab** array defines the table number. So for this output,

- students 12 and 37 are sitting at table number 0,
- students 34 and 123 are sitting at table number 1,
- students 1 and 4 are sitting at table number 2,
- students 13 and 54 are sitting at table number 3, and
- students 5 and 8 are sitting at table number 4.

There are multiple correct solutions. Assuming the input is valid, the lab seating chart should include each student exactly once, such that the two students at each table have the same group number.

Note: The elements of class array (**EE319K**) and the lab seating array (**lab**) are of different data types, **Student_t** and **LabTable_t** respectively.



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.