

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

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 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-assemble-run-debug the programming assignment. 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 Friday evening.

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

- 1) running any program from the PC other than **Keil uVision**, or a calculator,
- 2) communicating with **anyone else** except for the instructors **by any means** about this exam until Friday
- 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 **Exam2.c**

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: _____ April 7, 2016

```

UART #1
Exam2_ManhattanDistance
Test of Pack
Yes, Your= 0x0, Score = 4
Yes, Your= 0x30, Score = 8
Yes, Your= 0x4F, Score = 12
Yes, Your= 0x59, Score = 16
Yes, Your= 0xBC, Score = 20
Test of ManhattanDistance
Yes, Your= 3, Score = 24
Yes, Your= 8, Score = 28
Yes, Your= 15, Score = 32
Yes, Your= 0, Score = 36
Yes, Your= 30, Score = 40
Test of Highest
Yes, Your= 4, Score = 46
Yes, Your= 5, Score = 52
Yes, Your= 6, Score = 58
Yes, Your= 1, Score = 64
Yes, Your= 0, Score = 70
Test of MaxManhattanDistance
Yes, Your= 4, Score = 75
Yes, Your= 9, Score = 80
Yes, Your= 21, Score = 85
Yes, Your= 8, Score = 90
Yes, Your= 22, Score = 95
Yes, Your= 15, Score = 100
End of Exam2_ManhattanDistance
  
```

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/xxx>
 User: yyy
 Password: zzz

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 file **Exam2.c**. 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 wish create backup versions of your program. If you wish to roll back to a previous version, simply open one of the backup versions.

My main program will call your functions multiple times, and will give your solution a performance score of 0 to 100. *You should not modify my main program or my example data.* Each time you add a block of code, you should run my 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.

The exam has four parts a) through d), details of which are given in the starter code (**Exam2.c**). Each of the questions is independent, however, calling your subroutines in a) and b) in subroutine from d) can simplify your solution.

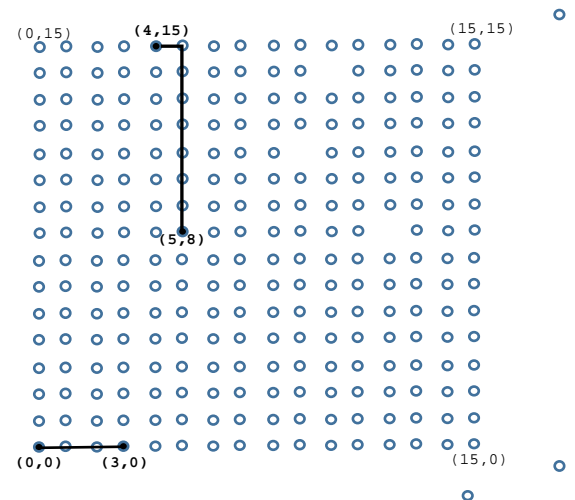
Part a) Write a C function, called **Pack**, which takes a pair of 4-bit inputs (x, y) and packs them into an 8-bit result and returns. Note that, the two 4-bit inputs are passed to your function as the lower four bits of the 8-bit variables x and y. The input x is the higher nibble (4-bits) of the result and the input y is the lower nibble.

Ex1: If (x=0x05, y=0x09) then **Pack** returns 0x59

Ex2: If (x=0x04, y=0x0F) then **Pack** returns 0x4F

Part b) Write a C function, called **ManhattanDistance**, which calculates the distance between two points in a city grid. The two points, n_1 and n_2 are passed as packed 8-bit coordinate pairs (x_1, y_1) and (x_2, y_2) . If each intersection on a city grid is assigned a coordinate pair (x,y), the Manhattan distance between two such intersections (x_1, y_1) and (x_2, y_2) is given as the distance (in city blocks) between the two intersections, given by

$$|x_2 - x_1| + |y_2 - y_1|$$



That is, the sum of the absolute difference between their x and y coordinates.

Ex1: The Manhattan distance from n_1 (0x00) to n_2 (0x30) is $(|0-3| + |0-0|) = 3+0 = 3$.

Ex2: The Manhattan distance from n_1 (0x4F) to n_2 (0x58) is $(|5-4| + |8-15|) = 1+7 = 8$.

Part c) Write a C function called **Highest** that returns the largest number n such that $a^n \leq m$. The given inputs to your function are a and m .

Ex1: $a=2$, $m=16$; the highest power of 2 that makes $2^n \leq 16$ is 4 ($2^0=1$, $2^1=2$, $2^2=4$, $2^3=8$, **$2^4=16$** , $2^5=32$)

Ex1: $a=3$, $m=5$; the highest power of 3 that makes $3^n \leq 250$ is 5 ($3^0=1$, $3^1=3$, $3^2=9$, $3^3=27$, $3^4=81$, **$3^5=243$** , $3^6=729$)

Part d) Write a C function called **MaxManhattanDistance** that finds the maximum Manhattan distance among all pairs of points given in an array of specified size. The two inputs to your function are the number of points (which can be between 2 to 10) and the array of points. Each point is a struct of two members, x , y which are the coordinates of the point.

```
struct point{
    uint8_t x;
    uint8_t y;
}
typedef struct point point_t
```

Note that with n points there are $n*(n-1)/2$ pairs of points to consider.

Ex1: Array has 3 points, $\{(0,2), \{1,1\}, \{2,0\}\}$. There are 3 Manhattan distances to consider:

$(0,2)$ to $(1,1) \rightarrow 1+1 = 2$; $(0,2)$ to $(2,0) \rightarrow 2+2 = 4$;

$(1,1)$ to $(2,0) \rightarrow 1+1 = 2$

The maximum among these is **4**.

Ex1: Array has 4 points, $\{(1,1), \{1,4\}, \{0,2\}, \{6,0\}\}$. There are 6 Manhattan distances to consider:

$(1,1)$ to $(1,4) \rightarrow 0+3 = 3$; $(1,1)$ to $(0,2) \rightarrow 1+1 = 2$; $(1,1)$ to $(6,0) \rightarrow 5+1 = 6$;

$(1,4)$ to $(0,2) \rightarrow 1+2 = 3$; $(1,4)$ to $(6,0) \rightarrow 5+4 = 9$;

$(0,2)$ to $(6,0) \rightarrow 6+2 = 8$;

The maximum among these is **9**.

Important Notes:

- Your functions should work for all cases given to it by the grader.
- Handle the simple cases first and the special cases last.
- ***This exam is different from other exams. You should NOT call your function in part c) from within your function in part d). You may or may not want to call a) and b) from d)***

Submission Guidelines:

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