



Epic Games - C++ Programming Exercise

Thank you for taking the time to complete this short programming exercise!

Instructions:

- Please give an approximate time that it took you to solve the questions - it's OK to spend as much time as you want, but usually 2 hours is sufficient.
- State your assumptions and explain your choices when you solve each problem. If you ask for a clarification, we will usually direct you back to this instruction.
- We request that you keep the answers to these questions private. If you post them to a web link, GitHub account etc. please do your best to make them private.
- If possible, please deliver your response in a plain text file or document format, zip file is fine. We do not accept PDFs.

All code is run through a test suite to test its production quality. These tests affect whether or not our interview process continues, so please take them seriously. Note that there may be multiple ways to solve these programming exercises - document your assumptions and choices. Your implementation is expected to be robust, and production ready! Consider whether you'd be comfortable checking this into Epic's source depot.

Question 1

Write the code for the following function, without using any built-in functions except malloc or operator new. Document your assumptions and explain your choices.

```
char* itoa(int Value, int Base);
```

where the returned value is allocated on behalf of the caller, value is the integer to convert, and base is octal, decimal, or hex.

Question 2

Write a function with the following signature that, given a row-major matrix of integers, builds a string with the entries of that matrix appended in clockwise order. Unlike the previous question, you may use built-in functions. You may also use your solution from the previous question, but are not required to. OutBuffer is guaranteed to be valid and large enough to hold all of the data. Document your assumptions and explain your choices.

```
void BuildStringFromMatrix(int* Matrix, int NumRows, int NumColumns, char* OutBuffer);
```

For instance, the 3x4 matrix below:

```
[2, 3, 4, 8,  
5, 7, 9, 12,  
1, 0, 6, 10]
```

would make the string "2, 3, 4, 8, 12, 10, 6, 0, 1, 5, 7, 9".

