**Question 1:** **Write a full and complete sentence at each >>> line that describes the single code line above it:**

#define TIME\_LIMIT 3

#define TIME\_MAX 6

#define TIME\_MIN 1

#define FAIL 0

#define PASS 1

int main()

{

int value[5] = { …. };

**>>>**

int retValue = FAIL;

value[1]++;

**>>>**

if (value[1] == TIME\_LIMIT)

{

retValue = PASS;

}

if ((value[4] >= TIME\_MAX) || (value[0] < TIME\_MIN))

**>>>**

{

printf("ERROR: Value is out of range\n");

**>>>**

}

return retValue;

**>>>**

}

**Question 2: When compiling an application you’re supposed to test, you see the below error message. Write a bug report at the >>> that you would provide to the developer to help them fix it. What other information would you want to include that you don’t have access to below?**

F:\hold\1.CoreSC\CoreSC\build>scons --platform=e9xxx\_vk10sc\_windows\_x64 -j8 --gpu=polaris12 --apps=SampleApp

scons: Building targets ...

Compiling Object: F:\hold\1.CoreSC\corescapps\build\e9xxx\_vk10sc\_windows\_x64\vkframework\obj\debug\FwVkRenderText.obj

F:\hold\1.CoreSC\corescapps\vk10sc\framework\_apps\sampleApp\sampleApp.c(156) : error C2065: 'x' : undeclared identifier

F:\hold\1.CoreSC\corescapps\vk10sc\framework\_apps\sampleApp\sampleApp.c(156) : error C2146: syntax error : missing ';' before identifier 'appInfo'

scons: \*\*\* [F:\hold\1.CoreSC\corescapps\build\e9xxx\_vk10sc\_windows\_x64\apps\obj\debug\sampleApp\sampleApp.obj] Error 2

scons: building terminated because of errors.

**>>>**

**Question 3, Part A: What input values would you provide to the ‘light’ parameter in a test to completely test the below function?**

typedef enum lightOptions

{

BRIGHT = 128,

DARK,

DIFFUSE,

MIDDLE,

} lightOptions;

int importantFunction(unsigned int light)

{

switch (light)

{

case BRIGHT:

return PASS;

break;

case DARK:

return PASS;

break;

case MIDDLE:

return PASS;

break;

default:

return FAIL;

break;

}

}

**>>>**

**Question 3 Part B: How would you setup the test that uses the above inputs? Use this as your starting point and add to it as needed, print the result of calling the function, do not test for validity:**

void test1()

{

importantFunction();

}

**Question 4: Why is this code not robust (2 reasons) and how would you fix it?**

int getResult(unsigned int \*pData)

{

const unsigned int start = 5;

return start / \*pData;

}

**>>>**

**Question 5: Given the below function, what parameter values would you provide a test to ensure that RenderTexImage() is called?**

void testFunction(void \* pContext, GLenum target, GLint level, GLsizei width, GLenum format)

{

uint8 isFormatValid = (GL\_ALPHA == format) || (GL\_LUMINANCE == format);

unsigned chipReturnVal = 0;

if ((GL\_TEXTURE\_2D != target) || !isFormatValid)

{

coreaviGlSetError(pContext, GL\_INVALID\_ENUM);

}

else if ((level < 0) || (level > RENDER\_MAX\_TEXTURE\_LEVELS) || (width <= 0))

{

coreaviGlSetError(pContext, GL\_INVALID\_VALUE);

}

else

{

chipReturnVal = RenderTexImage(123);

}

}

**>>**

**target:**

**level:**

**width:**

**format:**

**Question 6:** At the **>>>** create a definition of ‘wholeBuffer’ such that both “Inside” and “Outside” are printed to the screen. You can add other variables to help out if you want. If you get stuck you can ask the interviewer for help.

struct part

{

unsigned int id;

};

struct whole

{

struct part \*partFirst;

};

int main()

{

int count = 0;

const struct whole \*wholeLocal = NULL;

**>>>**