Q1:

1)

|  |  |  |
| --- | --- | --- |
|  | CPU | GPU |
| Hyper-threading | 2 threads per core | 4 to 10 per core |
| Threads for task-parallelism or data-parallelism | task-parallelism | data-parallelism |
| design goal | increase processor-usage | hide memory latency |

2)

OpenGL: Open graphics library. Is a cross-language, cross-platform application programming interface (API) for rendering 2D and 3D vector graphics.

Eg1: A simple red cube drawn with minimal OpenGL calls.

Eg2: Arcball like rotation of a chunky dinosaur.

3)

JOGL: The JOGL project hosts the development version of the Java Binding for the OpenGL API, and is designed to provide hardware-supported 3D graphics to applications written in Java.

JOGL provides full access to the APIs in the OpenGL, ES and EGL specification as well as nearly all vendor extensions. Use the content under the jogl.jar package.

4)

3D Modeling: In 3D computer graphics, 3D modeling is the process of developing a mathematical representation of any [surface](https://en.wikipedia.org/wiki/Surface_(mathematics)) of an object (either inanimate or living) in three dimensions via specialized software.

3D rendering: is the 3D computer graphics process of converting 3D models into 2D images on a computer.

5)

Throughput: is the number of such actions executed or results produced per unit of time.

Memory latency: is the time between initiating a request for a byte or word in memory until it is retrieved by a processor.

Adjust hardware to reduce latency, such as choosing the right processor (CPU or FPGA), choosing SSD, high-performance computing (HPC) specialized interconnect solutions, Data Plane Development Kit (DPDK) and so on.

Q2.

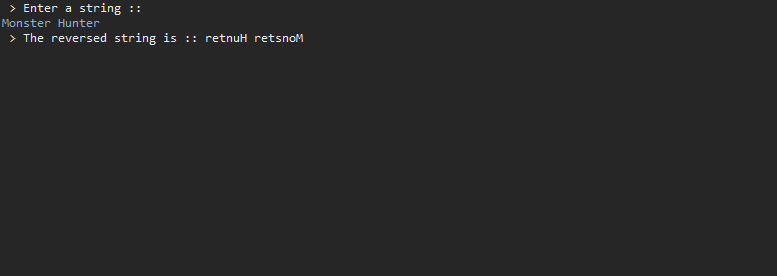
(a) The output is:



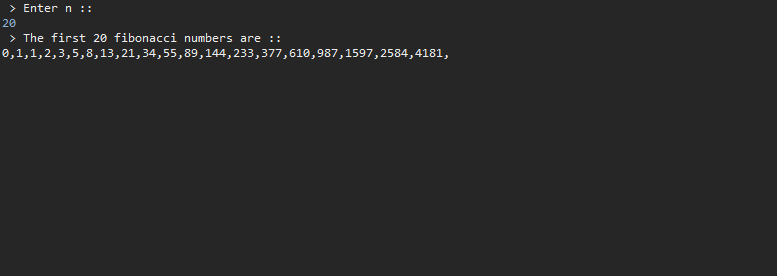
(b) The output is:



(c) The output is:



(d) The output is:



(e) The output is:



Q3

b) struct Student{

string name;

int id;

int midterm;

int project;

int finalterm;

int nextID;

}

JNI native - > JNIEXPORT jobjectArray JNICALL Java\_com\_test\_Test\_getStudentDetails( JNIEnv \*, jclass )

c)

static void \*Studentthread (void \*p)

    {

        jlong res = longlastfn((long)p);

        JNIEnv \*env;

        int n=javaVM->AttachCurrentThread(&env,NULL);

        // use env to access any function

        javaVM->DetachCurrentThread();

        return (void\*)res;

    }

  static void \*Graderthread (void \*p){}

    static jlong fibNR(JNIEnv \*env, jclass clazz, jlong n) {

        pthread\_t st1,st2,st3,st4,st5,gt;

        pthread\_create(&st1,NULL, Studentthread,(void\*)n);

pthread\_create(&st2,NULL, Studentthread,(void\*)n);

pthread\_create(&st3,NULL, Studentthread,(void\*)n);

pthread\_create(&st4,NULL, Studentthread,(void\*)n);

pthread\_create(&st5,NULL, Studentthread,(void\*)n);

pthread\_create(&gt,NULL, Graderthread,(void\*)n);

        return 0;

}

Q4

