```c
#include <stdio.h>

// OpenCL Headers
#include <CL/opencl.h>

int main(void)
{
    // function declarations
    void printOpenCLDeviceProperties(void);

    // code
    printOpenCLDeviceProperties();
}

void printOpenCLDeviceProperties(void)
{
    // code
    printf("OpenCL INFORMATION :\n");
    printf("===========================================================================\n");

    cl_int result;
    cl_platform_id ocl_platform_id;
    cl_uint dev_count;
    cl_device_id *ocl_device_ids;
    char oclPlatformInfo[512];

    // get first platform ID
    result=clGetPlatformIDs(1,&ocl_platform_id,NULL);
    if(result != CL_SUCCESS)
    {
        printf("clGetPlatformIDs() Failed\n");
        exit(EXIT_FAILURE);
    }

    // get GPU device count
    result=clGetDeviceIDs(ocl_platform_id,CL_DEVICE_TYPE_GPU,0,NULL,&dev_count);
    if(result != CL_SUCCESS)
    {
        printf("clGetDeviceIDs() Failed\n");
        exit(EXIT_FAILURE);
    }
    else if(dev_count==0)
    {
        printf("There Is No OpenCL Supprted Device On This System.\n");
        exit(EXIT_FAILURE);
    }
    else
    {
        // get platform name
        clGetPlatformInfo(ocl_platform_id,CL_PLATFORM_NAME,500,&oclPlatformInfo,NULL);
        printf("OpenCL Supporting GPU Platform Name : %s\n",oclPlatformInfo);
        // get platform version
        clGetPlatformInfo(ocl_platform_id,CL_PLATFORM_VERSION,500,&oclPlatformInfo,NULL);
        printf("OpenCL Supporting GPU Platform Version : %s\n",oclPlatformInfo);
        // print supporting device number
        printf("Total Number Of OpenCL Supporting GPU Device/Devices On This System : %d\n",dev_count);
        // allocate memory to hold those device ids
        ocl_device_ids=(cl_device_id *)malloc(sizeof(cl_device_id)*dev_count);
        // get ids into allocated buffer
        clGetDeviceIDs(ocl_platform_id,CL_DEVICE_TYPE_GPU,dev_count,ocl_device_ids,NULL);
        char ocl_dev_prop[1024];
        int i;
        for(i=0;i<(int)dev_count;i++)
        {
            printf("\n");
            printf("********** GPU DEVICE GENERAL INFORMATION ***********\n");
            printf("=====================================================\n");
            printf("GPU Device Number                               : %d\n",i);
            clGetDeviceInfo(ocl_device_ids[i],CL_DEVICE_NAME,sizeof(ocl_dev_prop),&ocl_dev_prop,NULL);
            printf("GPU Device Name                                 : %s\n",ocl_dev_prop);
            clGetDeviceInfo(ocl_device_ids[i],CL_DEVICE_VENDOR,sizeof(ocl_dev_prop),&ocl_dev_prop,NULL);
            printf("GPU Device Vendor                               : %s\n",ocl_dev_prop);
            clGetDeviceInfo(ocl_device_ids[i],CL_DRIVER_VERSION,sizeof(ocl_dev_prop),&ocl_dev_prop,NULL);
            printf("GPU Device Driver Version                       : %s\n",ocl_dev_prop);
            clGetDeviceInfo(ocl_device_ids[i],CL_DEVICE_VERSION,sizeof(ocl_dev_prop),&ocl_dev_prop,NULL);
            printf("GPU Device OpenCL Version                       : %s\n",ocl_dev_prop);
            cl_uint clock_frequency;
            clGetDeviceInfo(ocl_device_ids[i],CL_DEVICE_MAX_CLOCK_FREQUENCY,sizeof(clock_frequency),&clock_frequency,NULL);
            printf("GPU Device Clock Rate                           : %u\n",clock_frequency);

            printf("\n");
            printf("*********** GPU DEVICE MEMORY INFORMATION **********\n");
            printf("====================================================\n");
            cl_ulong mem_size;
            clGetDeviceInfo(ocl_device_ids[i],CL_DEVICE_GLOBAL_MEM_SIZE,sizeof(mem_size),&mem_size,NULL);
            printf("GPU Device Global Memory                        : %llu Bytes\n",(unsigned long long)mem_size);
            cl_device_local_mem_type local_mem_type;
            clGetDeviceInfo(ocl_device_ids[i],CL_DEVICE_LOCAL_MEM_SIZE,sizeof(mem_size),&mem_size,NULL);
            printf("GPU Device Local Memory                         : %llu Bytes\n",(unsigned long long)mem_size);
```

```c
            clGetDeviceInfo(ocl_device_ids[i], CL_DEVICE_MAX_CONSTANT_BUFFER_SIZE,sizeof(mem_size),&mem_size,NULL);
            printf("GPU Device Constant Buffer Size                : %llu Bytes\n",(unsigned long long)mem_size);

            printf("\n");
            printf("************** GPU DEVICE COMPUTE INFORMATION **************\n");
            printf("============================================================\n");
            cl_uint compute_units;
            clGetDeviceInfo(ocl_device_ids[i],CL_DEVICE_MAX_COMPUTE_UNITS,sizeof(compute_units),&compute_units,NULL);
            printf("GPU Device Number Of Parallel Processors Cores : %u\n",compute_units);
            size_t workgroup_size;
            clGetDeviceInfo(ocl_device_ids[i],CL_DEVICE_MAX_WORK_GROUP_SIZE,sizeof(workgroup_size),&workgroup_size,NULL);
            printf("GPU Device Work Group Size                     : %u\n",(unsigned int)workgroup_size);
            size_t workitem_dims;
            clGetDeviceInfo(ocl_device_ids[i],CL_DEVICE_MAX_WORK_ITEM_DIMENSIONS,sizeof(workitem_dims),&workitem_dims,NULL);
            printf("GPU Device Work Item Dimensions                : %u\n",(unsigned int)workitem_dims);
            size_t workitem_size[3];
            clGetDeviceInfo(ocl_device_ids[i],CL_DEVICE_MAX_WORK_ITEM_SIZES,sizeof(workitem_size),&workitem_size,NULL);
            printf("GPU Device Work Item Sizes                     : %u/%u/%u\n",(unsigned int)workitem_size[0],(unsigned int)workitem_size[1],(unsigned int)workitem_size[2]);
        }
        free(ocl_device_ids);
    }
}

```