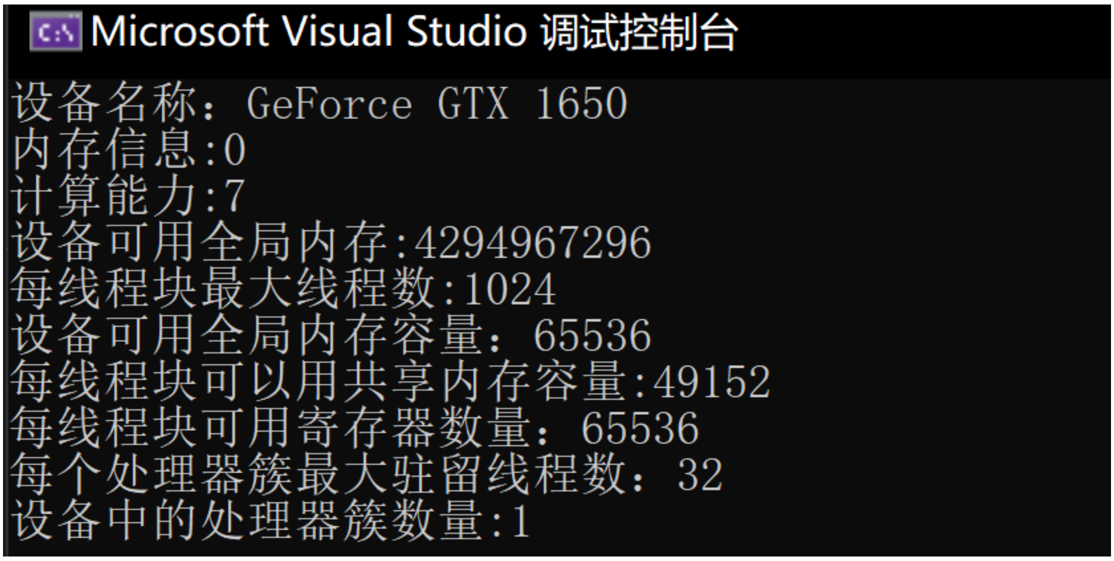
第一问.查询设备：
截图：



```
设备名称： GeForce GTX 1650
内存信息:0
计算能力:7
设备可用全局内存:4294967296
每线程块最大线程数:1024
设备可用全局内存容量： 65536
每线程块可以用共享内存容量:49152
每线程块可用寄存器数量： 65536
每个处理器簇最大驻留线程数： 32
设备中的处理器簇数量:1
```

代码：

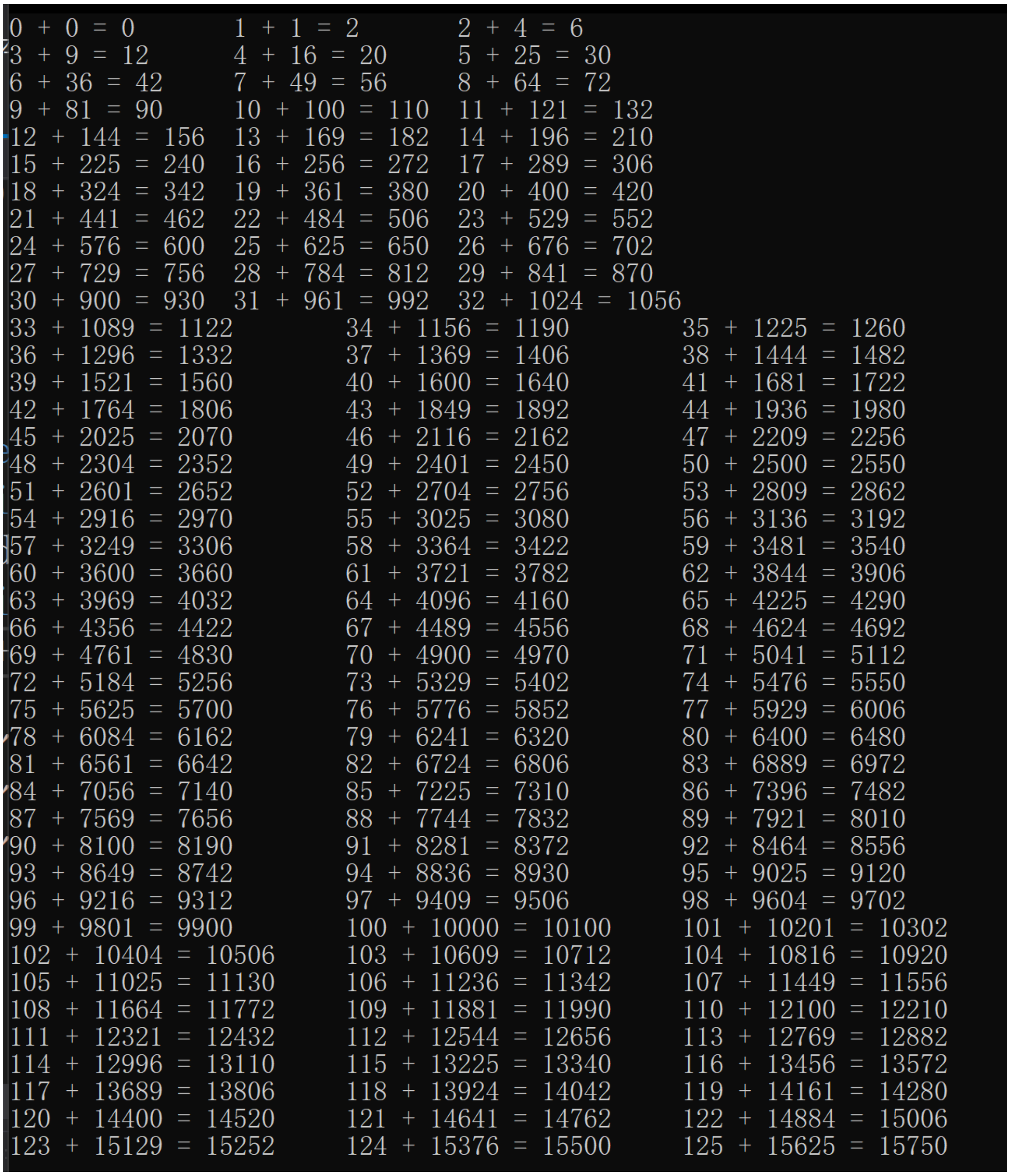
```
#include "stdio.h"
#include "cuda_runtime.h"
#include "device_launch_parameters.h"
#include <iostream>
using namespace std;

int main()
{
    cudaDeviceProp prop;

    int count;
    cudaGetDeviceCount(&count);

    for (int i = 0; i < count; i++)
    {
        cudaGetDeviceProperties(&prop, i);
        cout << "设备名称： " << prop.name << endl;
        cout << "内存信息:" << i << endl;
        cout << "计算能力:" << prop.major << endl;
        cout << "设备可用全局内存:" << prop.totalGlobalMem << endl;
        cout << "每线程块最大线程数:" << prop.maxThreadsPerBlock << endl;
        cout << "设备可用全局内存容量： " << prop.totalConstMem << endl;
        cout << "每线程块可以用共享内存容量:" << prop.sharedMemPerBlock<< endl;
        cout << "每线程块可用寄存器数量： " << prop.regsPerBlock << endl;
        cout << "每个处理器簇最大驻留线程数： " << prop.warpSize << endl;
    }
    cout << "设备中的处理器簇数量:" << count << endl;
    return 0;
}
```

第二问. 向量相加



```
0 + 0 = 0          1 + 1 = 2          2 + 4 = 6
3 + 9 = 12         4 + 16 = 20        5 + 25 = 30
6 + 36 = 42        7 + 49 = 56        8 + 64 = 72
9 + 81 = 90        10 + 100 = 110     11 + 121 = 132
12 + 144 = 156     13 + 169 = 182     14 + 196 = 210
15 + 225 = 240     16 + 256 = 272     17 + 289 = 306
18 + 324 = 342     19 + 361 = 380     20 + 400 = 420
21 + 441 = 462     22 + 484 = 506     23 + 529 = 552
24 + 576 = 600     25 + 625 = 650     26 + 676 = 702
27 + 729 = 756     28 + 784 = 812     29 + 841 = 870
30 + 900 = 930     31 + 961 = 992     32 + 1024 = 1056
33 + 1089 = 1122          34 + 1156 = 1190          35 + 1225 = 1260
36 + 1296 = 1332          37 + 1369 = 1406          38 + 1444 = 1482
39 + 1521 = 1560          40 + 1600 = 1640          41 + 1681 = 1722
42 + 1764 = 1806          43 + 1849 = 1892          44 + 1936 = 1980
45 + 2025 = 2070          46 + 2116 = 2162          47 + 2209 = 2256
48 + 2304 = 2352          49 + 2401 = 2450          50 + 2500 = 2550
51 + 2601 = 2652          52 + 2704 = 2756          53 + 2809 = 2862
54 + 2916 = 2970          55 + 3025 = 3080          56 + 3136 = 3192
57 + 3249 = 3306          58 + 3364 = 3422          59 + 3481 = 3540
60 + 3600 = 3660          61 + 3721 = 3782          62 + 3844 = 3906
63 + 3969 = 4032          64 + 4096 = 4160          65 + 4225 = 4290
66 + 4356 = 4422          67 + 4489 = 4556          68 + 4624 = 4692
69 + 4761 = 4830          70 + 4900 = 4970          71 + 5041 = 5112
72 + 5184 = 5256          73 + 5329 = 5402          74 + 5476 = 5550
75 + 5625 = 5700          76 + 5776 = 5852          77 + 5929 = 6006
78 + 6084 = 6162          79 + 6241 = 6320          80 + 6400 = 6480
81 + 6561 = 6642          82 + 6724 = 6806          83 + 6889 = 6972
84 + 7056 = 7140          85 + 7225 = 7310          86 + 7396 = 7482
87 + 7569 = 7656          88 + 7744 = 7832          89 + 7921 = 8010
90 + 8100 = 8190          91 + 8281 = 8372          92 + 8464 = 8556
93 + 8649 = 8742          94 + 8836 = 8930          95 + 9025 = 9120
96 + 9216 = 9312          97 + 9409 = 9506          98 + 9604 = 9702
99 + 9801 = 9900          100 + 10000 = 10100       101 + 10201 = 10302
102 + 10404 = 10506       103 + 10609 = 10712       104 + 10816 = 10920
105 + 11025 = 11130       106 + 11236 = 11342       107 + 11449 = 11556
108 + 11664 = 11772       109 + 11881 = 11990       110 + 12100 = 12210
111 + 12321 = 12432       112 + 12544 = 12656       113 + 12769 = 12882
114 + 12996 = 13110       115 + 13225 = 13340       116 + 13456 = 13572
117 + 13689 = 13806       118 + 13924 = 14042       119 + 14161 = 14280
120 + 14400 = 14520       121 + 14641 = 14762       122 + 14884 = 15006
123 + 15129 = 15252       124 + 15376 = 15500       125 + 15625 = 15750
```

其中 a[i]=i,b[i]=i^2

代码：

```
#include "stdio.h"
#include "cuda_runtime.h"
#include "device_launch_parameters.h"
#include <iostream>
using namespace std;
#define N 128
__global__ void add(int* a, int* b, int* c)
```

```
{
    int tid = threadIdx.x; //获取数据索引位置，每个线程对应一个位置
    if (tid < N)
        c[tid] = a[tid] + b[tid];
}

int main()
{
    int a[N], b[N], c[N];
    int* dev_a, * dev_b, * dev_c;
    cudaMalloc((void**)&dev_a, N * sizeof(int));
    cudaMalloc((void**)&dev_b, N * sizeof(int));
    cudaMalloc((void**)&dev_c, N * sizeof(int));
    for (int i = 0; i < N; i++)
    {
        a[i] = i;
        b[i] = i * i;
    }
    cudaMemcpy(dev_a,a,N * sizeof(int),cudaMemcpyHostToDevice);
    cudaMemcpy(dev_b, b, N * sizeof(int), cudaMemcpyHostToDevice);
    add << <1, N >> > (dev_a, dev_b, dev_c);
    cudaMemcpy(c, dev_c, N * sizeof(int), cudaMemcpyDeviceToHost);
    for (int i = 0; i < N-2; i+=3)
    {
        printf("%d + %d = %d\t", a[i], b[i], c[i]);
        printf("%d + %d = %d\t", a[i+1], b[i+1], c[i+1]);
        printf("%d + %d = %d\n", a[i + 2], b[i + 2], c[i + 2]);
    }
    cudaFree(dev_c);
    return 0;
}
```