1. 某机器提供 128KB 的地址空间，4 个段，代码段位于最低地址空间，其次是数据段、堆和栈，且栈位于最高地址空间，向下增长，如图所示。

   1) 虚地址为多少二进制位？每个段最大为多少 KB？每个段的起始虚地址

   2) 假设一个进程被加载到物理内存，其段表如图所示，请问以下虚地址对应的物理地址分别是多少？

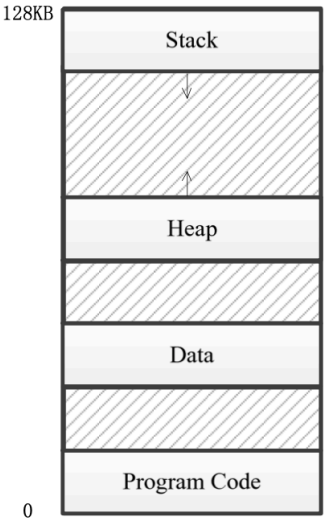
      a) 0x12ec4

      b) 0x1f362

      c) 0x0b600

      d) 0x1a850

| Seg | Base | Size | Grow |
|---|---|---|---|
| 0 | 0x100000 | 2KB | 1 |
| 1 | 0x180000 | 2KB | 1 |
| 2 | 0x240000 | 3KB | 1 |
| 3 | 0x200000 | 4KB | 0 |




2.

Suppose that a machine has 48-bit virtual addresses and 32-bit physical addresses.

(a) If pages are 4 KB, how many entries are in the page table if it has only a single level? Explain.

(b) Suppose this same system has a TLB (Translation Lookaside Buffer) with 32 entries. Furthermore, suppose that a program contains instructions that fit into one page and it sequentially reads long integer elements from an array that spans thousands of pages. How effective will the TLB be for this case?

3.

Consider the following C program:

```
int    X[N];
int step = M;      /* M is some predefined constant */
for (int i = 0; i < N; i += step) X[i] = X[i] + 1;
```

(a) If this program is run on a machine with a 4-KB page size and 64-entry TLB, what values of *M* and *N* will cause a TLB miss for every execution of the inner loop?

(b) Would your answer in part (a) be different if the loop were repeated many times? Explain.

4.

A computer has 32-bit virtual addresses and 4-KB pages. The program and data together fit in the lowest page (0–4095) The stack fits in the highest page. How many entries are needed in the page table if traditional (one-level) paging is used? How many page table entries are needed for two-level paging, with 10 bits in each part?

5.

A computer whose processes have 1024 pages in their address spaces keeps its page tables in memory. The overhead required for reading a word from the page table is 5 nsec. To reduce this overhead, the computer has a TLB, which holds 32 (virtual page, physical page frame) pairs, and can do a lookup in 1 nsec. What hit rate is needed to reduce the mean overhead to 2 nsec?