一、 选择题：

1. 以下程序的输出结果是 _____

```
#include <stdio.h>
int main(void)
{
        unsigned int x = 3, y = -1, z = 2;
        if (x > y)
                if (y < 0)
                        z = 0;
                else
                        z += 1;
        printf("%d", z);
        return 0;
}
```

A. 0　　B. 2　　C. 3　　D. 无正确答案

答案：B。如果无“unsigned”答案是 A

2. 下面程序的运行结果是 _____

```
char c[5]={ ‘a’ ,’ b’ ,’ \0’ ,’ c’ ,’ \0’ };
printf( “%s” ,c);
```

A. ‘a’ ‘b’　　B. abc　　C. ab c　　D. ab

答案：D。ab　printf 遇到 ‘\0’ 作为字符串结束标记

3. 下面程序段中，for 循环的执行次数是 _____

```
char *s = “\ta\018bc” ;
for( ; *s != ‘\0’ ; s++)
        printf( “*” );
```

A. 9　　B. 没有正确答案　　C. 6　　D. 2　　E. 5

答案：C。6　\01 被显示成八进制的 ASCII 1 输出。六个字符依次 0x9|0x61|0x1|0x38|0x62|0x63|，

若将 8 改为 <7 的任意数则为 5 次

4. 下面程序段的运行结果是 _____

```
int n=0;
while ( n++ <= 2) ;
printf(“%d”,n);
```

A. 2　　B. 3　　C. 4　　D. 有语法错误

答案：C。n 为 2 满足条件自增为 3，再次判断不满足条件，循环跳出，后缀自增增为 4.

5. 输出结果是 _____ 假定程序运行在 32 位平台上

```
typedef  struct{
    int a;
    int b;
} node;
        int  foo(int a[10], char b[10 ], node c, node d[10])
        {
                printf("a = %d\n", sizeof A ;
                printf("b = %d\n", sizeof B. ;
                printf("c = %d\n", sizeof C. ;
                printf("d = %d\n", sizeof D. ;
                return 0;
        }
```

A. 40 10 4 80　　B. 4 4 8 4　　C. 40 10 8 80

D. 4 4 8 40　　E. 4 10 8 10　　F. 8 8 8 8

答案为 B. 参数 1/2/4 都是指针，32 位系统下都为 4，参 3 是结构体，大小为 8; 64 位平台为 F

6. 设有程序段：

```
int k = 10;
while (k = 0) {
    k = k -1;
}
```

则下面描述中正确的是 _____

A. while 循环执行 10 次　　B. 循环是无限循环

C. 循环语句一次也不执行　　D. 循环体语句执行一次

答案为 C.　　k = 0 表达式作为判别表达式

7. 已知 x = 43， ch = ‘A’ ， y = 0；则表达式(x >= y && ch < ‘B’ && !y)的值是 _____

A. 0 B. 语法错 C. 1 D. “假”

答案为 C. 1 只涉及 char 型和 int 型隐式转换，未发生短路运算，表达式均为非 0 值

8. 若有定义 char *language[]={"FORTRAN", "BASIC", "PASCAL", "JAVA", "C"}; 则 language[2] 的值是 _____

A. 一个字符 B. 一个地址 C. 一个字符串 D. 不定值

答案为 B.

9. 以下函数中，和其他函数不属于一类的是____。 (阿里巴巴笔试题)

A. fwrite B. putc C. pwrite

D. putchar E). getline F) scanf

答案：C。 其他函数都是标库函数，该函数为系统函数(系统调用)

二、 填空题：

1. 本题均已 32 位操作系统为例：

若有定义：char *p = “abcd”，sizeof(p) = _____4，strlen(p) =_____4

若有定义：char p[10] = “hello”，sizeof(p) = _____10，strlen(p) =_____5

若有定义：void *p = malloc(100), sizeof(p) = _____4

2. 求 sizeof(p)，以 64 位平台为例。

```
char *p  = “hello!”;          _____
char p[] = “hello!”;          _____
void *p  = malloc(100);       _____
```

8、6、8

3. 下列程序执行后的输出结果是 _____ 12。 都是对地址操作，传址

```
void func (int *a,  int b[ ])
{
        b [0] = *a + 6;
}
int main (int argc, char* argv[ ])
```

```
{
        int a, b [5] ={0};
        a = 6;
        b [0] = 6;
        func(&a,  b);
        printf（"%d\n", b[0]）;
        return 0;
}
```

4. 内存布局问题（stack,heap,data,bss), 写出下列所有定义的变量在内存中位置 _____

```
#include <stdio.h>
int   a = 0;
int   b;
extern int x;
static   char   c, y = 'C';
int   main(int   argc, char* argv[ ])
{
        char      d=4;
        static   short   e, f = 0x23;
        char   *p = (char *)malloc(20);
        a++;
        b=100;
        c=(char)++a;
        e=(++D. ++;  //程序风格不好，勿仿！
        printf("a=%d,   b=%d,   c=%d,   d=   %d,   e=%d",a,b,c,d,e);
        return   0;
}
```

a:初始化为 0 的全局变量→bss　　b：未初始化的全局变量 bss

c：未初始化的静态全局 bss　　d：局部变量 stack

e：未初始化的静态局部变量 bss　　f：初始化为非 0 的静态局部变量 data

p：普通局部变量 stack，但 p 指向的空间(即 p 值)位于 heap

y：静态全局初始化为非 0 变量 data　　x：声明变量，只在 bss 记录，无地址可言

5. 有关内存的思考题

```
void GetMemory(char *p)
{
        p = (char *)malloc(100);
}
```

```
void Test(void)
{
        char *str = NULL;
        GetMemory(str);
        strcpy(str, "hello world");
        printf(str);
}
```
请问运行 Test 函数会有什么样的结果？简述原因

```
char *GetMemory(void)
{
        char p[] = "hello world";
        return p;
}
void Test(void)
{
        char *str = NULL;
        str = GetMemory();
        printf(str);
}
```
请问运行 Test 函数会有什么样的结果？简述原因

```
void GetMemory(char **p, int num)
{
        *p = (char *)malloc(num);
}
void Test(void)
{
        char *str = NULL;
        GetMemory(&str, 100);
        strcpy(str, "hello");
        printf(str);
}
```
请问运行 Test 函数会有什么样的结果？简述原因

```
void Test(void)
{
        char *str = (char *) malloc(100);
        strcpy(str, “hello”);
        free(str);
        if(str!= NULL)
        {
                strcpy(str, “world”);
                printf(str);
        }
}
```
请问运行 Test 函数会有什么样的结果？简述原因，应如何避免。

问 1：段错误！str 指针在执行 strcpy 的时候仍为 NULL

问 2：乱码，非正常值。str 接收了函数 GetMemory 的返回值，即局部变量 p 指针指向的 stack 上的地址。但该地址随着函数调用结束无实际意义。

问 3：正常打印 hello。GetMemory 函数的调用为 Test 函数的局部变量 str 做了有效初始化，执行 strcpy 的时候，str 指向 heap 空间一片有效地址(100 字节)。但应择机释放内存

问 4：能正常打印 world 到屏幕。调用 free 只是告诉动态内存分配器内存使用完毕，可以回收。但 str 变量中仍保存着该块内存区域的首地址。在该内存没有被覆盖前，仍能通过 str 变量正常访问。应在 free 后手动将 str=NULL;

6. 标准 I/O 提供了三种类型的缓冲，目的是尽可能减少使用 read 和 write 的次数。它们是：_____ 、_____ 、_____.

全缓冲、行缓冲、无缓冲

7. Linux 系统中，以 _____ 的方式访问物理设备，每个文件都使用 _____ 来标识。

文件、inode(或 i 节点)

8. 为一个文件更名时，该文件的实际内容并未移动，只需构造一个指向现有 i 节点的 _____ ，并解除与 _____ 的链接。

新目录项、旧目录项

9. i 节点包含了大多数与文件有关的信息：_____ 、_____ 、_____ 、指向该文件所占用的数据块指针等等。只有两项数据存放在目录项中：_____ 和 _____。

文件类型、文件访问权限位、文件长度； 文件名、i 节点编号

10. Linux 操作系统中，各种 ftp 服务器使用一般步骤有：_____、_____、_____、_____、_____、_____。

安装，服务器配置，服务器启动，客户端登录，上传、下载文件，退出登录。

11. vim 中，将 17 至 59 行中出现的所有 empathy 字符串替换为 example 的指令是 :_____

:17, 59s /empathy/example/g

12. 使用 _____ 命令，可在/usr/目录中搜索“EAGIN”字符串的宏定义。

sudo grep -r 'EAGAIN' /usr | grep 'define'

13. Ubuntu 系统中可使用 _____ 在线安装 tree 工具；使用 _____ 将 tree 卸载。

sudo apt-get install tree；sudo apt-get remove tree

14.查找/usr 目录下大小介于 900K-2M 之间的文件。并将这些文件的详细信息存入文件 result.txt 中 _____

find /usr/ -size +900k -size -2M -exec ls -lh {} \; > result.txt 或

find /usr/ -size +900k -size -2M | xargs ls -lh > result.txt

15.为减少磁盘物理操作 ,Linux 系统采用 _____、_____ 机制来读写磁盘文件，提高 IO 效率。

预读入、缓输出

三、 判断题：

1. 数组元素可以使用下标表示，也可以使用指针表示。 ( ) √
2. 字符串中最后可以不用 '\0' 表示结束 ( ) ×
3. 若有说明 int a[3][4]; 则 a[1+1][4]是对 a 数组元素的正确引用。 ( ) ×
4. 指针数组的元素是类型相同指针的集合。 ( ) √
5. break 语句可以出现在各种不同循环语句的循环体中。 ( ) √
6. for 循环是只有可以确定循环次数时才可使用，否则不能用 for 循环。 ( ) ×
7. 指针运算实际是地址运算，指针加一就是实际的地址值加一。 ( ) ×
8. 对于内核而言，所有打开的文件都通过文件描述符引用。 ( ) √
9. 称 read、write 为不带缓冲的 I/O 函数，是因为它们不使用内核缓冲区 ( ) ×
10. unlink 可以删除文件链接计数，当该数达到 0，内核即删除该文件 ( ) ×

四、 问答题：

1. 请写出 Linux 系统下压缩和解压缩命令(.tar.gz 格式文件)。简单解释一下 Makefile(即对 Makefile 的理解) 并阐述 Makefile 文件的作用是什么？

   压缩：tar zcvf xxx.tar.gz xxx1 xxx2 xxx3

   解压：tar zxvf xxx.tar.gz

   Makefile 是由程序员自行编写的用于提高程序编译效率的脚本文件。它可以提高程序编译工作效率，降低出错概率，管理冗杂的项目文件。其作用是将程序编译期间使用到的命令组织成一个脚本文件，方便统一管理和维护，降低程序重复编译出错概率。

2. Linux 系统中 aliee 用户宿主目录下有文件 exfile.doc，目录 Music。请分别为其创建硬链接和符号链接放置系统目录/usr/lib/abcd/目录下，链接名自定义。

   准备环境：sudo mkdir /usr/lib/abcd/

   exfile.doc 文件：硬链接：ln /home/aliee/exfile.doc /usr/lib/abcd/exfile.doc.h

   软链接：ln -s /home/aliee/exfile.doc /usr/lib/abcd/exfile.doc.s

   Music 目录：　硬链接：无法创建目录的硬链接

   软链接：ln -s /home/aliee/Music /usr/lib/abcd/Music_dir

3. 已知 IP 192.168.24.66 主机 Monica 用户宿主目录下有 games 目录。请使用 scp 命令将其拷贝至 aliee 用户的宿主目录下。假定 Monica 已经安装 openssh-server。且用户 aliee 已获知其登录密码。

   sudo scp -r Monica@192.168.24.66:/home/Monica/games　/home/aliee/

4. 分别写出使用文字设定法和数字设定法修改目录 dollar 权限为“drwxr-x--x”的命令，并查看修改是否成功

   文字设定法：chmod u=rwx,g=rx,o=x dollar

   数字设定法：chmod 751 dollar
   查看：ls -ld dollar

5. 如何生成 Linux 下的动态库和静态库，静态库和动态库的优缺点各是什么？

   静态库：使用静态库制作工具：ar rs libname.a a.o b.o c.o

   优点：将函数库中的函数本地化。寻址方便，速度快。

   缺点：每个使用静态库的进程都要将库编译到可执行文件中加载到内存。内存消耗严重

   动态库：使用 gcc -shared 选项：gcc -shared -o libname.so a.o b.o c.o

   优点：多进程共享一份库文件，节省内存、易于更新

   缺点：相较于静态库而言库函数访问略慢。

6. 使用 gdb 调试工具调试程序的先决条件是什么？列举 gdb 调试工具启动调试、设置断点、查看变量、设置跟踪变量、单步调试、退出调试的常用命令。

   程序可运行，并使用-g 选项进行编译；

   start/run　break/b　print/p　display　step/s next/n　quit

7. 请写出 gcc 编译.c 源程序生成可执行文件，中间经历的 4 个过程及各阶段生成何种文件。 解释编译选项-L -l -I 各自作用。

预处理 -E 生成.i 预处理文件（C source）

编译 -S 生成.s 汇编文件（assembler source）

汇编 -c 生成.o 目标文件/可重定位文件（relocatable）

链接 无参数 生成 可执行文件（executable）

-L：指定动态库所在位置； -l：指定动态库名； -I：指定头文件所在位置

五、 编程题：

1. 编写一个程序，将小于 n 的所有质数找出来 (完美世界面试题)
2. 编程程序，要求程序执行效果等同于命令 cat file1 - file2 > out 执行效果。该命令可将 file1 文件内容、用户键盘输入内容、file2 文件内容合并至 out 文件中。注：Linux 系统下，Ctrl+d 可输出一个文件结束标记 EOF。
3. 编程统计指定目录下普通文件个数。包括其子目录下的普通文件，将文件总数打印至屏幕。