面试题

C 部分

1.static 关键字的作用?

1)第一个作用：隐藏。 当我们同时编译多个文件时，所有未加static前缀的全局变量和函数都具有全局可见性。

2)static的第二个作用是保持变量内容的持久。存储在静态数据区的变量会在程序刚开始运行时就完成初始化，也是唯一的一次初始化。共有两种变量存储在静态存储区：全局变量和static变量，只不过和全局变量比起来，static可以控制变量的可见范围，说到底static还是用来隐藏的。

3)static的第三个作用是默认初始化为0。其实全局变量也具备这一属性，因为全局变量也存储在静态数据区。在静态数据区，内存中所有的字节默认值都是0x00，某些时候这一特点可以减少程序员的工作量。

2.线程与进程的区别和联系?

一个进程可以有一个或者多个线程组成,

进程和程序并不是一一对应的，一个程序执行在不同的数据集上就成为不同的进程，可以用进程控制块来唯一地标识每个进程。而这一点正是程序无法做到的，由于程序没有和数据产生直接的联系，既使是执行不同的数据的程序，他们的指令的集合依然是一样的，所以无法唯一地标识出这些运行于不同数据集上的程序。一般来说，一个进程肯定有一个与之对应的程序，而且只有一个。而一个程序有可能没有与之对应的进程(因为它没有执行),也有可能有多个进程与之对应(运行在几个不同的数据集上)。

3.堆和栈的区别?

1)、栈区（stack）— 由编译器自动分配释放 ，存放函数的参数值，局部变量的值等。其操作方式类似于数据结构中的栈。

2)、堆区（heap） — 一般由程序员分配释放， 若程序员不释放，程序结束时可能由 OS 回收 。

4.C 语言如何判断两个单向无环链表是否相交?

只需判断两个链表的尾节点地址是否相同，相同则相交，不同则不相交

5.程序在内存中运行时，内存分几个区，各自用途?

1）栈 ——有编译器自动分配释放 2. 堆 —— 一般由程序员分配释放，若程序员不释放，程序结束时 可能由 OS 回收 3. 全局区（静态区） —— 全局变量和静态变量的存储是放在一块 的，初始化的全局变量和静态变量在一块区域，未初始化的全局变量和未初始化的静态变量在相邻的另一块区域。程序结束释放。 4. 另外还有一个专门放常量的地方。程序结束释放。

6.引用与指针有什么区别

指针指向一块内存，它的内容是所指内存的地址；引用是某块内存的别名

1. 指针是一个实体，而引用仅是个别名；
2. 引用使用时无需解引用(*)，指针需要解引用；
3. 引用只能在定义时被初始化一次，之后不可变；指针可变；
4. 引用没有 const，指针有 const；
5. 引用不能为空，指针可以为空；
6. “sizeof 引用”得到的是所指向的变量(对象)的大小，而“sizeof 指针”得到的是指针本身(所指向的变量或对象的地址)的大小；
7. 指针和引用的自增(++)运算意义不一样；
8. 从内存分配上看：程序为指针变量分配内存区域，而引用不需要分配内存区域。

7.

编程：

1.请问运行完Test函数后，会有什么样的结果。

（1）

```
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？

答：程序崩溃。

因为GetMemory并不能传递动态内存，

Test函数中的 str一直都是 NULL。

strcpy(str, "hello world");将使程序崩溃。

（2）

```
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？
(2)答：可能是乱码。
因为GetMemory返回的是指向“栈内存”的指针，该指针的地址不是 NULL，但其原现的内容已经被清除，新内容不可知。

（3）

```
void GetMemory2(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？
答：
（1）能够输出hello
（2）内存泄漏

（4）

```
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？
答：篡改动态内存区的内容，后果难以预料，非常危险。
因为free(str);之后，str成为野指针，
if(str != NULL)语句不起作用。

2.编写strcpy函数（10分）

已知strcpy函数的原型是

char *strcpy(char *strDest, const char *strSrc);

其中strDest是目的字符串，strSrc是源字符串。

（1）不调用C++/C的字符串库函数，请编写函数 strcpy

```
char *strcpy(char *strDest, const char *strSrc);
{
    assert((strDest!=NULL) && (strSrc !=NULL));
    char *address = strDest;
    while( (*strDest++ = * strSrc++) != '\0' )
       NULL ;
    return address ;
}
```

（2）strcpy能把strSrc的内容复制到strDest，为什么还要char * 类型的返回值？

答：为了实现链式表达式。

例如 int length = strlen(strcpy(strDest, “hello world”));

3.

```
#include <stdio.h>

int main()
{

    int a = 0x0101;
    int b = 0x0202;
    int c;
    c = a&(~b);
    c = c|b;
    printf("%x,%d\n",c,c);
    return 0;
}
```

答案： 303,771

4.完成下面函数以实现使用辗转相除法获取两个数（假设两个数都大于0）的最大公约数

example: gcd(20,5) = 5. gcd(3,10) = 1. gcd(1620,1280) = 20.

```
unsigned int gcd(unsigned int
a,unsigned int b)
```

```
{
int c =0;

    if(m%n==0) c =n;
        else
         c=   gcd(n,m%n);
    return c;
}
```

5.用嵌套方式写一个函数，函数返回N的阶层，要求尽量写完整

```
int fun(int n)
{
         if (n == 1)
                  return n;
         return n*(n-1);
}
```

6. 请写出下列数据类型的范围　　char,unsigned char,short,int
答：char （-127~128） unsigned char（0~255） short（-2^16-1 ~2^16）
int(-32768~32767)

7.请写出下面函数的返回值

```
 char fuc1()                                    {
unsigned int a = 6;
int b = -12;
return(a+b>6)?1:0
}
#define SQP(x)(x*x)
int fuc2()
{
int a = 3;
return SQR(a+2);
}
```

答：1　11

8.请计算下列结构所占字节数

```
Typedef structYouKnoow
{
int id;
short age;
char level;

}
```

答：8

9. 关键字 const 有什么含义？
const 修饰谁，谁在整个程序运行过程中不能变

10. 下方代码输出结果为
```
main()
{
 int a[5] = {1, 2, 3, 4, 5};
 int *ptr = (int *)(&a+1);
 printf(“%d%d”, *(a+1), *(ptr-1));
}
```
答：D:4, 5

11.用 c/c++实现冒泡排序
```
      void swap_sort(int *p, int n) {
            int I, j;
            int tmp;
            for(i=0; i<n-1; i++) {
                  for(j=0; j<n-1-I; j++) {
                        if(p[j]>p[j+1])    {
                              tmp=p[j];
                              p[j]=p[j+1];
                              p[j+1]=tmp;
              }
                  }
       }
   }
```

12.用嵌套的方式写一个函数，该函数返回 N 的阶乘（N!=1*2*..*N）
```
int func(int n){if (n==1) return 1; return n * x(n-1);}
```