

1、下列程序运行的结果是什么？答：（ ）

```
/* main.c */
int i=0;
int main()
{
    foo();
    return 0;
}

/* foo.c */
int i=1;
void foo()
{
    printf("%d", i);
}
```

- A. 编译错误 B. 链接错误
C. 段错误 D. 有时打印输出 1，有时打印输出 0；
- 2、在链接时，对于什么样的符号一定不需要进行重定位？答：（ ）
A. 不同 C 语言源文件中定义的函数
B. 同一 C 语言源文件中定义的全局变量
C. 同一函数中定义时不带 static 的变量
D. 同一函数中定义时带有 static 的变量
- 3、考虑如下两个程序（fact1.c和fact2.c）：

```

/* fact1.c */
#define MAXNUM 12
int table[MAXNUM];
int fact(int n);
int main(int argc, char **argv) {
    int n;
    table[0] = 0;
    table[1] = 1;
    if (argc == 1) {
        printf("Error: missing argument\n");
        exit (0);
    }
    argv++;
    if (sscanf(*argv, "%d", &n) != 1 || n < 0 || n >= MAXNUM)

```

```

    {
        printf ("Error: %s not an int or out of range\n",
*argv);
        exit (0);
    }
    printf("fact(%d) = %d\n", n, fact(n));
}

/* fact2.c */
int* table;
int fact(int n) {
    static int num = 2;
    if (n >= num) {
        int i = num;
        while (i <= n) {
            table[i] = table[i-1] * i;
            i++;
        }
        num = i;
    }
    return table[n];
}

```

(1) 对于每个程序中的相应符号，给出它的属性（局部变量、强全局变量或弱全局变量），以及它在链接后位于 ELF 文件中的什么位置？（提示：如果某表项中的内容无法确定，请画 X）（6 分）

fact1.c

变量	类型	ELF Section
table		
fact		
num		

fact2.c

变量	类型	ELF Section
table		
fact		
num		

(2) 对上述两个文件进行链接之后，会对每个符号进行解析。请给出链接后下列符号被定义的模块 (fact1 or fact2)。(2 分)

	定义模块
table	
fact	
num	

(3) 使用 gcc (命令: gcc -o fact fact1.c fact2.c) 来编译之后得到的可执行文件是否能够正确执行? 为什么? (2 分)

4、请阅读以下程序，然后回答问题 (假设程序中的函数调用都可以正确执行):

```
int main() {  
    printf("A\n");  
    if (fork() == 0) {  
        printf("B\n");  
    }  
    else {  
        printf("C\n");  
        A  
    }  
    printf("D\n");  
    exit(0);  
}
```

(1) 如果程序中的 A 位置的代码为空，列出所有可能的输出结果: (1 分)

(2) 如果程序中的 A 位置的代码为:

```
waitpid(-1, NULL, 0);
```

列出所有可能的输出结果: (2 分)

(3) 如果程序中的 A 位置的代码为:

```
printf("E\n");
```

列出所有可能的输出结果: (2 分)