

# //////////////////////////////////// 西邮 Linux 兴趣小组 2023 纳新面试题 //////////////////////////////////////

学长寄语：长期以来，西邮 Linux 兴趣小组的面试题以难度之高名扬西邮校内。我们作为出题人也清楚的知道这份试题略有难度。请你动手敲一下代码。别担心，若有同学能完成一半的题目，就已经十分优秀。其次，相比于题目的答案，我们对你的思路和过程更感兴趣，或许你的答案略有瑕疵，但你正确的思路和对知识的理解足以为你赢得绝大多数的分数。最后，做题的过程也是学习和成长的过程，相信本试题对你更加熟悉的掌握 C 语言的一定有所帮助。祝你好运。我们东区逸夫楼 FZ103 见！

- ① 本题目只作为西邮 Linux 兴趣小组 2023 纳新面试的有限参考。
- ② 为节省版面，本试题的程序源码省去了 `#include` 指令。
- ③ 本试题中的程序源码仅用于考察 C 语言基础，不应当作为 C 语言「代码风格」的范例。
- ④ 所有题目编译并运行于 x86\_64 GNU/Linux 环境。

## 0. 鼠鼠我啊，要被祸害了

有 1000 瓶水，其中有一瓶有毒，小白鼠只要尝一点带毒的水，24 小时后就会准时死亡。至少要多少只小白鼠才能在 24 小时内鉴别出哪瓶水有毒？

## 1. 先预测一下~

按照函数要求输入自己的姓名试试~

```
char *welcome() {
    // 请你返回自己的姓名
}

int main(void) {
    char *a = welcome();
    printf("Hi, 我相信 %s 可以面试成功!\n", a);
    return 0;
}
```

## 2. 欢迎来到 Linux 兴趣小组

有趣的输出，为什么会这样子呢~

```
int main(void) {
    char *ptr0 = "Welcome to Xiyou Linux!";
    char ptr1[] = "Welcome to Xiyou Linux!";
    if (*ptr0 == *ptr1) {
        printf("%d\n", printf("Hello, Linux Group - 2%d", printf("")));
    }
    int diff = ptr0 - ptr1;
    printf("Pointer Difference: %d\n", diff);
}
```

## 3. 一切都翻倍了吗

- ① 请尝试解释一下程序的输出。
- ② 请谈谈对 `sizeof()` 和 `strlen()` 的理解吧。
- ③ 什么是 `sprintf()`，它的参数以及返回值又是什么呢？

```
int main(void) {
    char arr[] = {'L', 'i', 'n', 'u', 'x', '\\0', '!'}, str[20];
    short num = 520;
    int num2 = 1314;
```

```
printf("%zu\\t%zu\\t%zu\\n", sizeof(*&arr), sizeof(arr + 0),
        sizeof(num = num2 + 4));
printf("%d\\n", sprintf(str, "0x%x", num) == num);
printf("%zu\\t%zu\\n", strlen(&str[0] + 1), strlen(arr + 0));
}
```

## 4. 奇怪的输出

程序的输出结果是什么？解释一下为什么出现该结果吧~

```
int main(void) {
    char a = 64 & 127;
    char b = 64 ^ 127;
    char c = -64 >> 6;
    char ch = a + b - c;
    printf("a = %d b = %d c = %d\\n", a, b, c);
    printf("ch = %d\\n", ch);
}
```

## 5. 乍一看就不想看的函数

“人们常说互联网凛冬已至，要提高自己的竞争力，可我怎么卷都卷不过别人，只好用一些奇技淫巧让我的代码变得高深莫测。”

这个 `func()` 函数的功能是什么？是如何实现的？

```
int func(int a, int b) {
    if (!a) return b;
    return func((a & b) << 1, a ^ b);
}

int main(void) {
    int a = 4, b = 9, c = -7;
    printf("%d\\n", func(a, func(b, c)));
}
```

## 6. 自定义过滤

请实现 `filter()` 函数：过滤满足条件的数组元素。

提示：使用函数指针作为函数参数并且你需要为新数组分配空间。

```
typedef int (*Predicate)(int);
int *filter(int *array, int length, Predicate predicate,
            int *resultLength); /*补全函数*/

int isPositive(int num) { return num > 0; }

int main(void) {
    int array[] = {-3, -2, -1, 0, 1, 2, 3, 4, 5, 6};
    int length = sizeof(array) / sizeof(array[0]);
    int resultLength;
    int *filteredNumbers = filter(array, length, isPositive,
                                   &resultLength);

    for (int i = 0; i < resultLength; i++) {
        printf("%d ", filteredNumbers[i]);
    }
    printf("\\n");
    free(filteredNumbers);
    return 0;
}
```



## 7.静...态...

- ① 如何理解关键字 `static`?
- ② `static` 与变量结合后有什么作用?
- ③ `static` 与函数结合后有什么作用?
- ④ `static` 与指针结合后有什么作用?
- ⑤ `static` 如何影响内存分配?

## 8.救命! 指针!

数组指针是什么? 指针数组是什么? 函数指针呢? 用自己的话说出来更好哦, 下面数据类型的含义都是什么呢?

```
int (*p)[10];
const int* p[10];
int (*f1(int))(int*, int);
```

## 9.咋不循环了

程序直接运行, 输出的内容是什么意思?

```
int main(int argc, char* argv[]) {
    printf("[%d]\n", argc);
    while (argc) {
        ++argc;
    }
    int i = -1, j = argc, k = 1;
    i++ && j++ || k++;
    printf("i = %d, j = %d, k = %d\n", i, j, k);
    return EXIT_SUCCESS;
}
```

## 10.到底是不是 TWO

```
#define CAL(a) a * a * a
#define MAGIC_CAL(a, b) CAL(a) + CAL(b)
int main(void) {
    int nums = 1;
    if(16 / CAL(2) == 2) {
        printf("I'm TWO(>ω<)/\n");
    } else {
        int nums = MAGIC_CAL(++nums, 2);
    }
    printf("%d\n", nums);
}
```

## 11.克隆困境

试着运行一下程序, 为什么会出现这样的结果?

直接将 s2 赋值给 s1 会出现哪些问题, 应该如何解决? 请写出相应代码。

```
struct Student {
    char *name;
    int age;
};
```

```
void initializeStudent(struct Student *student, const char *name,
                      int age) {
    student->name = (char *)malloc(strlen(name) + 1);
    strcpy(student->name, name);
    student->age = age;
}

int main(void) {
    struct Student s1, s2;
    initializeStudent(&s1, "Tom", 18);
    initializeStudent(&s2, "Jerry", 28);
    s1 = s2;
    printf("s1 的姓名: %s 年龄: %d\n", s1.name, s1.age);
    printf("s2 的姓名: %s 年龄: %d\n", s2.name, s2.age);
    free(s1.name);
    free(s2.name);
    return 0;
}
```

## 12.你好, 我是内存

作为一名合格的 C-Coder, 一定对内存很敏感吧~来尝试理解这个程序吧!

```
struct structure {
    int foo;
    union {
        int integer;
        char string[11];
        void *pointer;
    } node;
    short bar;
    long long baz;
    int array[7];
};

int main(void) {
    int arr[] = {0x590ff23c, 0x2fbc5a4d, 0x636c6557, 0x20656d6f,
                0x58206f74, 0x20545055, 0x6577202c, 0x6d6f636c,
                0x6f742065, 0x79695820, 0x4c20756f, 0x78756e69,
                0x6f724720, 0x5b207075, 0x33323032, 0x7825005d,
                0x636c6557, 0x64fd6d1d};
    printf("%s\n", ((struct structure *)arr)->node.string);
}
```

## 13.GNU/Linux (选做)

注: 嘿! 你或许对 Linux 命令不是很熟悉, 甚至你没听说过 Linux。但别担心, 这是选做题, 了解 Linux 是加分项, 但不了解也不扣分哦!

你知道 `cd` 命令的用法与 `/` `.` `~` 这些符号的含义吗?

请问你还懂得哪些与 GNU/Linux 相关的知识呢~

恭喜你攻克所有难关! 迎难而上的决心是我们更为看重的。

来到这里的人已是少数, 莫踌躇在成功的门槛前。

自信一点, 带上你的笔记本电脑, 来东区逸夫楼 FZ103 面试吧!

排版: 纸鹿, 有问题扣他鸡腿。

