

王飞龙的笔试

● 笔试信息

候选人姓名： 王飞龙

邮箱： 1438262313@qq.com

完成时间： 2022-03-03 21:54

试卷名称： C语言笔试题二

笔试评卷： 程红

评卷时间： 2022-03-04 09:42

面试官评价： 无

笔试得分：

0 分 / 100 分

笔试用时：

86 分钟 / 90 分钟

是否作弊： 未标记作弊

切屏记录： 1次 / 07秒

复制粘贴： -

视频监控： 候选人未开启

人脸识别： -

多人识别： -

换人识别： -

异常识别： -

● 得分详情

题型	用时	正确题数/总题数	得分/总分	得分率
问答题	85 分钟	0/12	0/100	0%

● 答题详情

一. 问答题（共12题） 得分： 0/100 得分率： 0% 用时： 85 分钟

1. 写一个宏MAX_ABC(A, B, C), 实现三个数的最大值。这个宏有三个参数：A、B、C分别代表输入的三个数

得分： 0/4 用时： 10 分钟 答题时间： 2022-03-03 20:27

答题详情

```
#define MAX_ABC(A,B,C) (A>B? ((A>C)? A:C):(B>C? B:C))
```

2. 在使用第1题中实现的宏时, 有什么需要注意的地方吗?

得分： 0/2 用时： 1 分钟 答题时间： 2022-03-03 20:38

答题详情

注意宏是整体替换, 括号必备, 不然会在替换时候出现错误。

3. 关键字static的作用是什么?

得分: 0/6 用时: 1 分钟 答题时间: 2022-03-03 20:39

答题详情

用在全局变量及函数, 约束其作用域。
用在局部变量, 延长其生命周期。

4. 在C语言中可以使用#define和typedef来实现声明一个已经存在的数据类型的同义字, 例如“#define UINT32 unsigned int”, “typedef unsigned int UINT32”, 这两者的区别是什么? 用哪一种实现比较好, 为什么?

得分: 0/4 用时: 5 分钟 答题时间: 2022-03-03 20:40

答题详情

define是一种字符替换的命名方式, typedef是正规的重命名数据类型的方法。严格意义上说使用typedef比较好。

举个例子吧:

UINT32 *P, Q, P是无符号整形指针, Q是无符号整形容器。而使用typedef就不一样, 两个都是无符号整形指针。

5. 试评价下面一段代码

```
int a[64][256][1024];
int i, j, k;
for (k = 0; k < 1024; k++)
    for (j = 0; j < 256; j++)
        for (i = 0; i < 64; i++)
            a[i][j][k] = i * 100 + j + 10 + k;
```

得分: 0/6 用时: 6 分钟 答题时间: 2022-03-03 20:46

答题详情

该代码初始化一个3维数组的值。但是该代码3层循环, 把循环次数最多的一次放在外面, 势必会造成循环cpu跨切循环的次数。

6. 计算下面结构体的size

```
struct
{
    int i_val_1;
    short s_val_1;
    char c_val_1;
    char c_val_2;
    short s_val_2;
    int i_val_2;
    int i_val_3;
    short s_val_3;
    char c_val_3;
    struct s_t
    {
        char c_val;
        int i_val;
        short s_val;
    } s_val_1, s_val_2;
};
```

得分: 0/5 用时: 2 分钟 答题时间: 2022-03-03 20:53

答题详情

假设内存以4字节对齐：

32

7. 找出下面的代码中的问题

```
void func1()
{
    char str_src[10], str_dst[10];
    int index;
    for (index = 0; index < 10; index++)
    {
        str_src[index] = 'a' + index;
    }
    strcpy(str_dst, str_src);
    printf("%s", str_dst);
}

void func2(char *str_src)
{
    char str_dst[10];
    if (str_src && strlen(str_src) <= 10)
    {
        strcpy(str_dst, str_src);
        printf("%s", str_dst);
    }
}
```

得分: 0/6 用时: 5 分钟 答题时间: 2022-03-03 20:55

答题详情

func1可能在strcpy和printf时出现死机,原因是在对strsrc时没有0作为字符串结束符。
func2在if判断时没有考虑strdst的最大长度是10,出现的情况和原因同上。

8. 下面的函数的目的是申请一个整型数组,数组大小与输入数组的大小相同,每个数组元素与数组元素相同。指出下面代码中的问题并修改?

```
int* alloc_data(int in[100])
{
    int cnt = sizeof (in);
    int *p_new = malloc(cnt * sizeof(int));
    int index;
    for (index = 0; index <= cnt; index++)
    {
        *(p_new + index * sizeof (int)) = in[index];
    }
    return p_new;
}
```

得分: 0/15 用时: 4 分钟 答题时间: 2022-03-03 21:01

答题详情

当指针作为函数参数时,当地址使用,如在32位机上,cnt是4,不符合愿意。
malloc分配空间后最好清理一下分配到的堆空间。
for循环内部企图通过指针偏移的方式对这片堆进行初始化,但是sizeof(int)会导致越界。

9. 请写出下面几个代码的输出

```

void func1()
{
    unsigned int a = 1, b = 20;
    int c = -10;
    printf("%s\n", (a + c > a) ? "a + c > a" : "a + c < a");
    printf("%s\n", (b + c > b) ? "b + c > b" : "b + c < b");
}

void fun2()
{
    char *s[] = {"one", "two", "three", "four"};
    char *p = s[1];
    p++;
    printf("%c, %c\n", *(p + 1), s[3][3]);
    int size_s = sizeof(s);
    int size_p = sizeof(p);
    printf("%d, %d", size_s, size_p);
}

```

得分: 0/12 用时: 4 分钟 答题时间: 2022-03-03 21:06

答题详情

a + c > a
 b + c > b
 0, r
 5, 4

10. 写一个函数判断系统大小端

得分: 0/10 用时: 5 分钟 答题时间: 2022-03-03 21:11

答题详情

```

union data
{
    char a;
    int b;
}p;

int main()
{
    p.b = 0X0102;
    if(p.a == 0X01)
    {
        printf("大端\n");
    }
    else
    {
        printf("小端\n");
    }
}

```

11. 给定一个整型变量a,判断a是不是前缀数据(前缀数据:高bit位是连续的1,其他bit位都是0。例如0x80000000、0xFFFFC000、0xFFFFFFFF是前缀数据,0x0FFF0000、0xFFFFFA000不是前缀数据)

得分: 0/10 用时: 28 分钟 答题时间: 2022-03-03 21:16

答题详情

```

int judge(int a)
{
    int len = -1, one = 0;
    int all = sizeof(int) * 8;
    char *p = (char *)&a;
    char *q = NULL;

```

```

while(len++ && len < all)
{
    if(p[len] == 1 && len < (all-1))
    {
        q = &p[len] + 1;
        while(q[one++] == 0 && one <= (all-len-2));
        if(one == (all-len-1))
        {
            printf("is\n");
        }
        else
        {
            printf("not is\n");
        }
    }
    else
    {
        printf("not is\n");
    }
}
return 0;
}

```

12. 已知节点NODE_T *pNode在一个双向链表NODE_T *pList中, 写一个函数实现: 节点pNode从链表pList中退出

```

typedef struct node_t
{
    struct node_t *pre;
    struct node_t *next;

    void *data;
} NODE_T;

```

得分: 0/20 用时: 9 分钟 答题时间: 2022-03-03 21:44

答题详情

```

pNode->pre->next = pNode->next;
pNode->next->pre = pNode->pre;
free(pNode);
pNode = NULL;

```

• 查看回放

点击下方链接查看候选人笔试回放

<https://www.showmebug.com/iwpads/DYYSQAJUBRFRPOZN>

⚠ 任何人都可以通过此链接加入“面试官31131的团队”，请不要转发给外人。链接会于此报告生成的30天后失效，失效时间 2022-04-02 21:54，失效后仅团队内成员登录查看回放