

面试题

答题时间：40 分钟

一. 选择题。

(1) 能将高级语言程序转换成目标语言程序的是_____。
A. 调试程序 B. 解释程序 C. 编译程序 D. 编辑程序

(2) 若有一下类型说明语句：

```
char w; int x; float y, z;
```

则表达式 $w*x+z-y$ 的结果为_____类型。

A. float B. char C. int D. Double

(3) 下列程序的输出结果是_____。

```
main()
{
    int a=0, b=0, c=0;
    if(++a > 0 || ++b > 0)
        ++c;
    printf("\n a = %d, b = %d, c = %d ", a, b, c);
}
```

A. a=0, b=0, c=0 B. a=1, b=1, c=1
C. a=1, b=0, c=1 D. a=0, b=1, c=1

(4) 以下程序段的输出结果是_____。

```
main()
{
    char s1[10], s2[10];
    scanf("%s", s1);
    gets(s2);
    puts(s1);
    Puts(s2);
}
```

输入数据如下：（此处代表回车符）

```
aaa
bbb
```

A. aaa B. bbb C. aaa\0bbb\0 D. aaabbb

(5) 若有定义：`int a[4][10]`；则以下选项中对数组元素 `a[i][j]` 引用错误的是_____。

($0 \leq i < 4, 0 \leq j < 10$)

A. `*(&a[0][0]+10*i+j)` B. `*(a+i)+j` C. `*(*(a+i)+j)` D. `*(a[i]+j)`

(6) 若一个外部变量的定义形式为 `static int x;`，那么，其中 `static` 的作用应该是_____。

A. 将变量存储在静态存储区 B. 使变量 `x` 可以由系统自动初始化
C. 使 `x` 只能在本文件内引用 D. 使 `x` 的值可以永久保留

(7) 在数据结构中，从逻辑上可以把数据结构分为_____。

A. 动态结构和静态结构 B. 紧凑结构和非紧凑结构
C. 线性结构和非线性结构 D. 内部结构和外部结构

- (8) 下面是对宏定义的描述, 不正确的是_____。
- A. 宏不存在类型问题, 宏名无类型, 它的参数也无类型
 - B. 宏替换不占用运行时间
 - C. 宏替换时先求出实参表达式的值, 然后带入形参运算求值
 - D. 宏替换只不过是字符替代而已
- (9) 下列程序的执行结果是_____。

```
#include

union un
{
    int    i;
    char   c[2];
};

void main()
{
    union un    x;
    x.c[0] = 10;
    x.c[1] = 1;
    printf("\n %d", x.i);
}
```

- A. 266 B. 11 C. 265 D. 138
- (10) 若有如下说明, 则_____的叙述是正确的。

```
struct st
{
    int    a;
    int    b[2];
}a;
```

- A. 结构体变量 **a** 与结构体成员 **a** 同名, 定义是非法的
- B. 程序只在执行到该定义时才为结构体 **st** 分配存储单元
- C. 程序运行时为结构体 **st** 分配 6 个字节存储单元
- D. 类型名 **struct st** 可以通过 **extern** 关键字提前引用 (引用在前, 说明在后)

二. 填空题

- (1) 对于长度为 **n** 的顺序存储的线性表, 当随机插入和删除一个元素时, 需平均移动元素的个数为_____。
- (2) 设有 **char a, b;**, 若要通过 **a&b** 运算屏蔽掉 **a** 中的其它位, 只保留第 2 位和第 8 位 (右起为第一位), 则 **b** 的二进制是_____。
- (3) 数据库技术的主要特点为数据的集成性、数据的高_____和低冗余性、数据的独立性和_____。
- (4) 函数 **delete(s, i, n)** 作用是从字符串 **s** 中删除从第 **i** 个字符开始的 **n** 个字符, 请填空。

```
void    delete(char s[], int i, int n)
{
    int j, k, length = 0;
    While(s[length])
    {
```

```

        }
        --i;
        j = i;
        if( )
        {
            k = i+n;
            if(i+n <= length)
            while(k < length)
            s[j++] = s[k++];
            S[j] = '\0';
        }
    }
}

```

三. 简答题

(1) 分别给出 bool, int, float, 指针变量与“零值”比较的 if 语句(假设变量名为 var)

(2) 为什么标准头文件都有类似以下的结构?

```

#ifndef __WELCOME
#define __WELCOME

extern "C"{
#endif
/*.....*/
#ifdef __plus
}

#endif
#endif /*__WELCOME*/

```

(3) struct 和 class 的区别以及 struct 和 union 的区别。

(4) 编写一个函数，把一个 char 组成的字符串循环右移 n 个。比如原来是“abcdefghi”如果 n = 2, 移位后应该是“hiabcdefgh”

```
//pstr 是指向以 '\0' 结尾的字符串的指针  
//steps 是要求移动的 n  
void loopmove(char *pstr, int steps)
```

答案及解析

选择题

第1题.选择C。

第2题.选择C。

- 根据乘法运算符，如果w是一个char类型，那么w * x的结果将是int类型，因为char类型会被提升为int类型进行计算。
- 根据加法运算符，如果w * x的结果是int类型，那么w * x + z的结果也将是int类型，因为int与float相加时，float类型会被自动转换为int类型。

- 最后, 根据减法运算符, 如果 $w * x + z$ 的结果是int类型, 那么 $w * x + z - y$ 的结果也将是int类型, 因为int类型与float类型相减时, float类型会被自动提升为int类型。

第3题.选择C。

1. 初始化变量 a、b、c 的值都为 0。
2. 进入循环, 条件表达式 $++a > 0 \ || \ ++b > 0$ 中的第一个操作数 a 先进行自增运算, a 变为 1。然后由于 1 大于 0, 整个条件表达式的结果已经为真, 不再进行后续的操作数的计算。所以 $++b > 0$ 的自增运算不会被执行。
3. 进入循环体, 执行 $++c$, 将 c 的值自增 1, c 变为 1。
4. 打印输出结果, 使用 printf 函数输出 a、b、c 的值。a 的值为 1, b 的值由初始化时的 0 保持不变, c 的值为 1。
5. 输出结果为 "a=1, b=0, c=1"。

因此, 正确的输出结果应该是 C. a=1, b=0, c=1。

第4题。选择D。

第5题。选择B。

第6题。选择C。

A 选项中提到的 "将变量存储在静态存储区" 并不是 `static` 关键字的作用, 而是所有具有静态存储期的变量 (包括全局变量和静态局部变量) 的存储方式, 与 `static` 关键字无关。

B.无论一个变量是否使用了 `static` 关键字, 编译器都会根据其作用域和存储类型对其进行初始化。

D.尽管静态局部变量的生存期延长到程序结束, 但其值仍然可以被修改, 不一定保持不变。

第7题。选择C。

1. 线性结构: 线性结构是指数据元素之间存在一对一的线性关系, 即每个元素只有零个或一个直接前驱元素, 以及零个或一个直接后继元素。线性结构的典型代表是线性表、栈、队列等。
2. 非线性结构: 非线性结构是指数据元素之间存在多对多的关系, 即每个元素可能有多个直接前驱元素和多个直接后继元素, 数据元素之间没有严格的顺序关系。非线性结构的典型代表是树、图等。

第8题。选择B。

宏替换是在编译时期进行的,发生在实际的编译之前, 它会在预处理阶段将宏调用替换为对应的代码。因此, 宏替换是发生在程序运行之前的, 会占用一定的编译时间和资源。

对于C项的理解, 举个例子:

```
#include <stdio.h>

#define SQUARE(x) ((x) * (x))

int main() {
    int a = 5;
    int b = SQUARE(a + 2);

    printf("b = %d\n", b);

    return 0;
}
```

在上面的例子中，宏 `SQUARE` 接受一个参数 `x`，并返回 `x` 的平方。在 `main` 函数中，我们定义了一个变量 `a`，并将其赋值为 5。然后，我们使用 `SQUARE` 宏来计算 `(a + 2)` 的平方，并将结果赋值给变量 `b`。

在宏替换过程中，`SQUARE(a + 2)` 首先会将参数 `a + 2` 的值求出来，即 7，然后将这个值带入到宏定义中进行替代，即 `((a + 2) * (a + 2))`。最终，变量 `b` 的结果就是 49。

第9题。选择A.

小端模式存储，使用联合体共享内存，`0000 0001 0000 1010 = 266`

第10题。选择B.

在 C 语言中，定义结构体类型仅仅是定义了一个新的类型，它并不会分配任何存储空间。只有当我们定义该结构体类型的变量时，计算机才会为该变量分配相应的存储空间。因此，在定义结构体类型 `struct st` 时，计算机并没有为其分配内存空间。只有在定义该结构体类型的变量时，才会分配内存空间。

填空题

1. $(n+1)/2$ 。

推导：假设我们有一个长度为 n 的顺序存储线性表，索引从 0 到 $n-1$ 。插入或删除元素时，随机选择的位置可以是 0 到 n 之间的任意整数（包括 0 和 n ）。

对于插入操作，如果我们随机选择将元素插入到位置 i ($0 \leq i \leq n$)，那么需要将索引为 $i, i+1, i+2, \dots, n-1$ 的所有元素向后移动一位，为新元素腾出位置。因此，需要移动的元素个数为 $n-i$ 。

对于删除操作，如果我们随机选择删除位置 i ($0 \leq i \leq n$)，那么需要将索引为 $i+1, i+2, \dots, n-1$ 的所有元素向前移动一位，填补删除的空缺。同样地，需要移动的元素个数为 $n-i$ 。

通过对所有可能的位置进行求和，并考虑每个位置的概率相等，我们可以计算平均移动元素的个数。

对于插入操作来说，平均移动元素的个数为： $\Sigma(n-i) / n$ ，其中 i 取值范围为 0 到 n

$= (n-0 + n-1 + n-2 + \dots + 2 + 1 + 0) / n = ((n+1)*n)/2 / n = (n+1)/2$

删除同理

2. 01000001.

3. 高性能、易扩展性

数据的独立性是指数据库中的数据与应用程序相互独立，数据库的结构和存储方式可以在不影响应用程序的情况下进行修改。这使得数据库系统具有良好的适应性和灵活性，可以根据需求进行调整和优化。

数据的集成性是指数据库可以将多个不同来源、不同格式的数据集成到一个统一的数据模型中。这使得数据库系统可以处理和管理大量复杂的数据，提供跨系统、跨组织的数据共享和交互。

数据的高性能和低冗余性是数据库技术追求的目标之一。通过使用索引、查询优化技术等手段，数据库系统可以实现快速的数据访问和高效的数据处理。同时，数据库设计也追求消除数据的冗余，避免数据的重复存储和更新操作，提高数据的存储效率和一致性。

数据的易扩展性是指数据库系统可以方便地扩展和调整存储容量、计算资源等，以满足不断增长和变化的数据需求。数据库系统支持的分布式架构和水平扩展技术可以实现大规模数据存储和处理的能力。

4.

```
void del(char *s,int i,int n)
{
```

```

    int j ,k,length=0;
    while(s[length]) length++;
    --i;
    j=i;
    if(i<length)
    {
        k= i+n;
        if(i+n <=length)
            while(k < length)
                s[j++] = s[k++];
        s[j]='\0';
    }
}

```

简答题

(1)

bool 类型变量与零值比较：

```

bool var = false;
if (var == false)
{
    // 执行代码，当 var 的值为 false 时
}

```

int 类型变量与零值比较：

```

int var = 0;
if (var == 0)
{
    // 执行代码，当 var 的值为 0 时
}

```

float 类型变量与零值比较：

```

float var = 0.0;
if (var == 0.0)
{
    // 执行代码，当 var 的值为 0.0 时
}

```

指针变量与零值比较：

```

int* var = nullptr; // 或者 int* var = NULL;
if (var == nullptr) // 或者 if (var == NULL)
{
    // 执行代码，当 var 的值为 nullptr (或 NULL) 时，表示指针为空指针
}

```

需要注意的是，bool 类型变量的零值是 false，int 类型变量的零值是 0，float 类型变量的零值是 0.0，指针变量的零值通常是 nullptr 或 NULL，表示空指针。

(2)

1. 防止头文件的重复包含：当一个源文件中多次包含同一个头文件时，头文件保护机制可以防止头文件的内容被重复定义，避免出现重复定义错误。
2. 提高编译速度：在编译过程中，如果多个源文件包含同一个头文件，没有头文件保护机制的话，会导致头文件的内容被多次编译，增加了编译时间。而使用头文件保护可以避免这种情况，提高编译的效率。
3. 解决符号重定义错误：头文件保护还可以解决不同的头文件中可能存在相同的宏定义、结构体、函数等名称，避免出现符号重定义的错误。

(3)

1. struct 和 class 的区别：

- 默认访问权限：在 `struct` 中，默认的成员访问权限是公共的 `public`，而在 `class` 中，默认的成员访问权限是私有的 `private`。
- 数据封装：`struct` 被广泛用于表示一组相关的数据，通常将它们公开以便外部访问。而 `class` 则更多地用于实现面向对象的封装，将数据和操作封装在一起，对外提供接口进行访问和操作。
- 继承：在 `struct` 中，默认继承方式是公有继承，即派生类可以访问基类的公有成员。而在 `class` 中，默认继承方式是私有继承，即派生类无法访问基类的公有成员。

2. struct 和 union 的区别：

- 内存分配：`struct` 是一种用户自定义的数据类型，可以定义多个成员，每个成员占用不同的内存空间，并且结构体的大小为所有成员大小的和。而 `union` 是一种特殊的数据类型，它的所有成员共享同一块内存空间，不同成员可以在不同的时间保存不同的值。`union` 的大小取决于最大的成员大小。
- 成员访问：`struct` 中的各个成员可以同时被访问和使用，每个成员有独立的内存地址。而 `union` 中的成员共享同一块内存空间，只能同时访问一个成员，访问其他成员会导致原有成员的值被覆盖。
- 存储方式：`struct` 以顺序方式存储各个成员，每个成员在内存中占据连续的空间。而 `union` 所有成员共享同一块内存空间，它们在内存中重叠存储。

总结起来，`struct` 和 `class` 的主要区别在于默认的成员访问权限和默认的继承方式，而 `union` 则是一种特殊的数据类型，用于共享内存空间。

(4)

```
void loopmove(char *pstr, int steps) {
    if (pstr == NULL || steps <= 0)
        return;

    int len = strlen(pstr);
    steps %= len; // 处理 steps 大于字符串长度的情况

    if (steps == 0)
        return;

    char temp;
    for (int i = 0; i < steps; i++) {
        temp = pstr[len - 1];

        for (int j = len - 1; j > 0; j--) {
            pstr[j] = pstr[j - 1];
        }

        pstr[0] = temp;
    }
}
```



```
}  
}
```