

## 2017 年北京航空航天大学招收硕士研究生入学考试试题 参考答案

（考生注意：全部答案必须写在答题纸上否则后果自负！）

考试科目代码：991

考试科目：数据机构与 C 语言程序设计

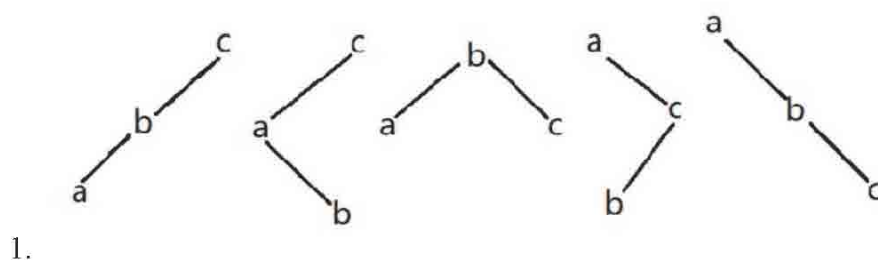
### 一、解：

题号	1	2	3	4	5	6	7	8	9	10
答案	D	B	A	B	A	C	C	D	D	D

### 二、解：

1. 顺序
2. 链式
3. 2
4.  $n(k-1) + 1$
5. 中序遍历序列
6.  $2(n-1)$
7. 队列
8. 哈希（散列）查找法
9. 14
10.  $O(n)$ ,  $O(n^2)$ ,  $O(n \log_2 n)$

### 三、解：



2. ABCDEF ;    ABCEDF

3. 算法各自有各自的适用场合。根据数据原始状态不同，排序过程性以及结果稳定性要求不同，采用不同的排序算法。

4. 第一趟： 49 13 27 50 76 38 65 97

第二趟： 27 13 49 38 65 50 76 97

第三趟： 13 27 38 49 50 65 76 97

#### 四、解：

```
void DeteleTheSame(LinkList list)
```

```
{
```

```
    LinkList p,q,t,s;
```

```
    p = list;
```

```
    p = p->link;           //头节点未存放数据
```

```
    while(p!=NULL)
```

```
    {
```

```
        t = p;           //t 指针保证始终在 q 指针前一位
```

```
        for(q = p->link; q != NULL;)
```

```
        {
```

```
            s = q;
```

```
            if(p->data == q->data)    //值相同
```

```
            {
```

```
                t->link = q->link;    //t 指针指向 q 前一个元素，这时 t->link 指
```

向 q->link,跳过 q

```
                q = q->link;           //q 指向下一个元素
```

```
                free(s);               //释放当前这个多余元素的地址
```

```
            }
```

```
        } else           //值不同
```

```
        {
```

```

t = t->link;           //t 和 q 要同时后移一位
q = q->link;           //上面值相同的情况只要求 q 后移，t 不后
移一位
    }
}
p = p->link;
}
}

```

### 五、解：

题号	1	2	3	4	5	6	7	8	9	10
答案	D	B	B	B	C	A	C	A	B	C

### 六、解：

```

1. switch(mo){
    case 1: max = 31; break;
    case 2: max = 28; break;
    case 3: max = 31; break;
    case 4: max = 30; break;
    case 5: max = 31; break;
    case 6: max = 30; break;
    default: max = 0; break;
}

```

2. 这三个指针变量一样大，并且都是 4。因为指针变量中存储的都是地址变量，然而地址变量的长度都是一样大小，所以无论该地址变量所指数据类型大小，该地址变量的长度都为 4。

3. 结构体（structure）是一种构造类型，它是由若干“成员”组成的。每一个成员可

以是一个基本数据类型或者又是一个构造类型，而且每个成员的数据类型可以相同也可以不相同。共同体（union）将几种不同的变量储存在同一内存单元中，也就是使用覆盖技术，几个变量互相覆盖，这种几个不同的变量共同占用一段内存的结构，可见二者最大的区别就是所占用的内存，结构体（structure）所占用的内存是分量内存之和，共同体（union）所占用的内存是等于最大的分量的内存。

结构体（structure）与共同体（union）主要有以下区别：

1) 结构体（structure）与共同体（union）都是由多个不同的数据类型成员组成，但在任何同一时刻，共同体（union）中只存放了一个被选中的成员，而结构体（structure）的所有成员都存在。在结构体（structure）中，各成员都占有自己的内存空间，它们是同时存在的。一个结构体（structure）变量的总长度等于所有成员长度之和。在共同体（union）中，所有成员不能同时占用它的内存空间，它们不能同时存在。共同体（union）变量的长度等于最长的成员的长度。

2) 对于共同体（union）的不同成员赋值，将会对其它成员重写，原来成员的值就不存在了，而对于结构体（structure）的不同成员赋值是互不影响的。

4. #include <stdio.h>

int main(void)

{

    //文件指针

    FILE \*fp;

    //文件路径

    char fileName[] = "c:\\tag.txt";

    //用来保存读取的最后一行字符

    char str[100] = { '\0' };

    //用来保存从文件读取的字符

    char ch;

    //文件内部指针 fseek()函数的偏移

    int i = -1;

    //如果文件打开失败

```
if ((fp = fopen(fileName, "r")) == NULL)
{
    printf("the file is empty.");
    return -1;
}
//移动指针离文件结尾 1 字节处
fseek(fp, i, SEEK_END);
//读取一个字符
ch = fgetc(fp);
if(ch==EOF)
{
    printf("the file is empty.");
    return -1;
}
//如果读取的字符不等于换行，如果等于换行说明是最后一行的上一行的换行
符
while (ch != '\n')
{
    //文件内部指针从文件结尾向前移动
    i--;
    fseek(fp, i, SEEK_END);
    ch = fgetc(fp);
}
i = 0;
ch = fgetc(fp);
//如果未到文件结尾
while (!feof(fp))
{

```

```
        //把读取的字符保存字符数组中

        str[i] = ch;

        i++;

        ch = fgetc(fp);

    }

    //输出从最后一行中读取的字符

    printf("%s\n", str);

    //关闭文件

    fclose(fp);

    return 0;

}
```

#### 七、解：

```
int fun(int count, int a)
{
    int i,tmp=1;
    for(i=1;i<=a;i++)
        tmp=count*tmp;
    return tmp;
}

int psum(int a, int n)
{
    int n,count=1;
    long a,sum=0,temp=0;
    if(a<=0 || n<=0)
        return 0;
    while(count<=n){
        temp = fun(count,a);
        sum += temp;
    }
}
```

```
count++;  
  
}  
  
return sum;  
  
}
```

## 八、解：

```
#include<stdio.h>  
  
int stoi(char *,int ,int);  
  
void output(int *,int);  
  
void main()  
{  
    char str[50];  
    int a[100]; //字符串中最多包含 100 个转化后的数字  
    int n=0; //转化后数字数组长度.  
    int i=0, j=-1; //i 是数组下标, j 作为某连续数字子字符串第一个字符位置.  
    int tmp=0; //转化后的数字  
    gets(str); //输入字符串.  
    while(str[i]) //当数组元素不是字符串结束符时.执行 while  
    {  
        if(str[i]>='0'&&str[i]<='9'&&j== -1)  
        {  
            j=i;  
            i++;  
        }  
        if(str[i]>='0'&&str[i]<='9'&&j!= -1)  
            i++;  
        if(str[i]<'0'&&str[i]>'9'&&j== -1)  
            i++;  
    }
```

```
if(str[i]<'0'&&str[i]>'9'&&j!=-1)
{
    tmp = stoi(str,j,i-1);
    a[n++] = tmp;
    j = -1;
    i++;
}
}
output(a,n);
}

int stoi(char *str,int j,int i) //将指定长度字符串数组转化为一个整数
{
    int m,res=0;
    for(m=j;m<=i;m++)
        res+=res*10+(str[m]-'0');
    return res;
}

void output(int a[],int n) //用选择排序算法对数组进行排序，然后输出
{
    for(int i=0; i < n-1; i++)
    {
        int j = i;
        int max = a[j];
        for(int k=i; k < n; k++) {
            if (a[k] > max) {
                j = k;
                max = a[k];
            }
        }
    }
}
```



```
    }  
    int t = a[i];  
    a[i] = a[j];  
    a[j] = t;  
}  
for(int i=0; i < n; i++)  
{  
    printf("%d ", a[i]);  
}  
}
```