

# 北京航空航天大学 2018 年 硕士研究生招生考试初试试题

科目代码: 991

## 数据结构与 C 语言程序设计 (共 9 页)

考生注意: 所有答题务必书写在考场提供的答题纸上, 写在本试题单上的答题一律无效 (本题单不参与阅卷)。

### 一、单项选择题(本题共 20 分, 每小题各 2 分)

1. 以下关于数据结构的叙述中, 正确的是\_\_\_\_\_。
  - A. 数据的逻辑结构独立于该数据的存储结构;
  - B. 数据的存储结构独立于该数据的逻辑结构;
  - C. 数据的逻辑结构唯一决定了该数据的存储结构;
  - D. 数据结构包括数据的逻辑结构和存储结构两个部分。
2. 以下关于线性表的叙述中, 错误的是\_\_\_\_\_。
  - A. 线性表采用顺序存储结构, 必须占用一片地址连续的存储单元;
  - B. 线性表采用顺序存储结构, 便于在表中进行插入和删除操作;
  - C. 线性表采用链式存储结构, 不必占用一片地址连续的存储单元;
  - D. 线性表采用链式存储结构, 便于在表中进行插入和删除操作。
3. 对于长度为  $n$  的非空顺序表, 以下 4 种操作中, 对应算法的时间复杂度为  $O(1)$  的是\_\_\_\_\_。
  - A. 在表的第  $i$  个位置插入一个新的数据元素( $1 \leq i \leq n+1$ );
  - B. 删除表中的第  $i$  个数据元素( $1 \leq i \leq n$ );
  - C. 查找表中第  $i$  个数据元素( $1 \leq i \leq n$ );
  - D. 对  $n$  个数据元素按值的大小进行排序。
4. 以下 4 种形式的链表中, 最适合作为队列的链表结构的是\_\_\_\_\_。
  - A. 单向链表;
  - B. 单向循环链表;
  - C. 双向链表;
  - D. 双向循环链表。

5. 若一棵二叉树的前序序列和后序序列分别为 a, b, c, d 和 d, c, b, a, 则该二叉树的中序序列不可能是\_\_\_\_\_。

- A. a, b, c, d;      B. b, c, d, a;      C. d, c, b, a;      D. c, b, d, a。

6. 一个连通图的生成树是包含该连通图全部顶点的\_\_\_\_\_。

- A. 极小连通子图;      B. 极大连通子图;  
C. 极小子图;      D. 极大子图。

7. 采用邻接表存储一个图所占用的存储空间大小\_\_\_\_\_。

- A. 仅与图中顶点的数目有关;  
B. 仅与图中边(或弧)的数目有关;  
C. 与图中顶点的数目和边(或弧)的数目都有关;  
D. 与图中边(或弧)的数目的平方有关。

8. 假设有 n 个关键字互为同义词, 若采用线性探测再散列法处理冲突, 则将这 n 个关键字散列到一个初始为空的散列地址空间中, 需要进行的探测次数为\_\_\_\_\_。

- A.  $n \times (n+1)$ ;      B.  $n \times (n+1)/2$ ;      C.  $n+1$ ;      D. n。

9. 在以下 4 种情况中, 最不利于快速排序法发挥其长处的是\_\_\_\_\_。

- A. 待排序的数据量太大;  
B. 待排序的数据元素的个数为奇数;  
C. 待排序的数据中含有多个值相同的元素;  
D. 待排序的数据已经基本按值有序。

10. 对于以下 4 种排序方法, 若待排序序列的初始状态为(1, 2, 3, 4, 5, 10, 6, 7, 8, 9), 则排序过程中元素之间的比较次数最少的是\_\_\_\_\_。

- A. 插入排序法;      B. 选择排序法;      C. 泡排序法;      D. 谢尔排序法。

## 二、简答题(本题共 20 分, 每小题各 5 分)

1. 线性表可以采用顺序存储结构, 也可以采用链式存储结构。在一个实际应用中, 若有多个线性表同时存在, 并且在处理过程中各表的长度会动态地发生变化, 请问: 在这种情况下, 线性表应该采用哪种存储结构? 为什么?

2. 若某完全二叉树的第 7 层有 10 个叶结点, 则该完全二叉树最少有多少个结点? 最多有多少个结点? 请简要说明结论的导出过程。

3. 设具有  $n$  个顶点、且边不带权的无向图采用邻接矩阵存储。请回答下列问题:

- (1) 如何计算该无向图中边的数目?
- (2) 如何计算该无向图中任意一个顶点的度?
- (3) 如何判断顶点  $i$  与顶点  $j$  之间是否存在边?

4. 为什么采用折半插入排序法所需要进行的元素之间的比较次数与待排序序列的初始状态无关? 在什么情况下, 采用折半插入排序法比采用直接插入排序法需要进行更多元素之间的比较?

三、综合题(本题共 20 分, 每小题各 5 分)

1. 已知某双向链表的链结点类型定义为:

```
typedef struct node{
    ElemType data;          /* 数据域 */
    struct node *llink, *rlink; /* 指向结点的直接前驱和直接后继的指针域 */
} *DLinkedList;
```

$p$  为指向链表中某结点的指针(假设所指结点不是链表的尾结点)。请用一句话写出下列算法的功能。

```
void FUNC(DLinkedList p)
{
    DLinkedList q;
    q=p->rlink;          /* q 指向 p 所指结点的直接后继结点 */
    if(q->rlink!=NULL)    /* 若 q 所指结点不是链表的最后结点 */
        q->rlink->llink=p; /* 将结点 p 插入到结点 q 的直接后继结点的前面 */
    p->rlink=q->rlink;     /* 将结点 q 的直接后继结点插入到结点 p 的后面 */
    p->llink->rlink=q;     /* 将结点 q 插入到结点 p 的前面 */
    q->llink=p->llink;
    q->rlink=p;
    p->llink=q;
}
```



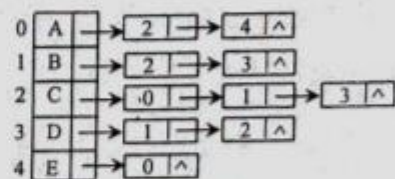
2. 设某堆栈采用顺序存储结构, 元素  $a_1, a_2, a_3, a_4, a_5, a_6$  依次进栈, 若 6 个元素的出栈顺序为  $a_2, a_3, a_4, a_6, a_5, a_1$ , 则堆栈的容量至少应该是多少个元素的空间? 请利用图示说明你的结论。

3. 已知某二叉树采用顺序存储结构(如题三 3 图所示), 其中, 空白处表示结点不存在。请分别给出该二叉树的前序遍历序列和后序遍历序列。



题三 3 图

4. 已知某无向图采用邻接表存储, 邻接表如题三 4 图所示。请分别写出从顶点 A 出发进行深度优先遍历与广度优先遍历后得到的遍历序列。



题三 4 图

#### 四、算法设计题(本题 15 分)

已知长度为  $n$  的顺序表  $A$  中元素按值从小到大排列, 请写出在该顺序表中插入一个新的数据元素  $item$  的非递归算法。

要求: 插入新元素之前采用折半查找法确定新元素的插入位置。

#### 五、单项选择题(本题共 20 分, 每小题各 2 分)

1. 以下关于 C 语言变量声明的语句中, 正确的是\_\_\_\_\_。

- A. `int my_num = 10000;;`                      B. `int Smynum = 10000;;`  
C. `int mynum = 10,000;;`                      D. `int my num = 10000;;`

2. 若变量  $a, b, c$  的取值分别是 1, 2, 3, 则表达式 " $!((b+c)>(a+4))$ " 的值是\_\_\_\_\_。

- A. 0;                      B. 1;                      C. 2;                      D. 3.

3. 以下关于循环语句的叙述中, 正确的是\_\_\_\_\_。

- A. `for` 循环语句的三个部分必须都要有表达式;  
B. `while` 循环语句的循环体内至少要有一条语句;  
C. `do...while` 循环语句的循环体至少会被执行一次;  
D. `continue` 语句可以退出包含它的整个循环体。

4. 对于下列代码段:

```
switch(option) {  
    case 'H' : printf("Hello ");  
    case 'W' : printf("Welcome ");  
    case 'B' : printf("Bye");  
    break;  
}
```

若 option 的取值为'W', 则该代码段的输出结果是\_\_\_\_\_。

- A. Welcome;
- B. Welcome Bye;
- C. Hello Welcome Bye;
- D. 以上结果都不对。

5. 若有以下变量的声明语句:

```
int i = 1, a[ ] = {0, 2, 4}, *b;  
b = &i;
```

则下列选项中, 其结果与表达式 “\*(a+1)” 相等的是\_\_\_\_\_。

- A. a[0];
- B. \*a + i;
- C. \*(a+b);
- D. \*(a\*b)。

6. 以下关于结构体和共同体的叙述中, 错误的是\_\_\_\_\_。

- A. 结构体和共同体都可以包含多个成员;
- B. 共同体的所有成员所占的空间大小必须相同;
- C. 结构体的成员可以是一个共同体;
- D. 共同体的成员可以是一个结构体。

7. 若已知有如下宏的定义

```
#define CANBERRA(x, y) ((x-y) / (x+y))
```

则以下表达式中, 返回结果值最大的是\_\_\_\_\_。

- A. CANBERRA(3.0, 2.0);;
- B. CANBERRA(4.0, 1.0);;
- C. CANBERRA(1.0+2.0, 0.0+2.0);;
- D. CANBERRA(1.0+2.0, 1.0+1.0);。

8. 对于以下 C 程序, 其正确的输出是\_\_\_\_\_。

```
#include <stdio.h>
```

```
int main(){  
    char str1[ ] = "Hello";  
    char str2[ ] = "Hello";  
    if (str1 == str2) printf("Equal\n");  
    else printf("Unequal\n");  
    return 0;  
}
```

- A. Unequal;                      B. Equal;  
C. 该程序无法通过编译;        D. 该程序运行时出错。

9. 已知有以下 sample.c 程序的定义:

```
/* sample.c */  
#include<stdio.h>  
int main(int argc, char *argv[ ] ) {  
    printf("%c", *++argv[2] );  
    return 0;  
}
```

将该程序编译成可执行文件 sample 后, 若在命令行下输入如下命令:

sample January February March

则该命令正确的输出是\_\_\_\_\_。

- A. J;                      B. a;                      C. F;                      D. e.

10. 对于以下递归函数的定义:

```
void recur (int num) {  
    if (num == 0) return;  
    else printf("%d,", num);  
    recur (num--);  
}
```

若执行“recur(3);”这样的函数调用,则该函数调用正确的输出是\_\_\_\_\_。

- A. 1,2,3,;  
B. 3,2,1,;  
C. 3,2,1,0,;  
D. 该函数调用会陷入死循环。

六、综合题(本题共 20 分, 每小题各 5 分)

1. 对于下列 **for** 循环语句, 请将其改写为功能完全相同的 **while** 循环语句。

```
int i, j, count = 0;
for (i = 0; i < 100; i++){
    for (j = 100; j >= i; j -= 2){
        count += j - i;
    }
}
```

2. 若分别采用下列两种不同的方式声明字符串变量 msg1 和 msg2,

```
char *msg1;      char msg2[20];
```

则请问：这两种声明方式有什么不同？使用它们时分别需要注意什么？

3. 在 C 语言中, 读文件的函数 `fread` 和 `fscanf` 有什么不同? 它们分别在什么情况下使用?

4. 已知数字 0 的 ASCII 码为 48, 请写出下列程序的正确输出。

```
#include <stdio.h>

int main(){
    char c=48;
    int i, mask=01;
    for (i=1; i<=5; i++){
        printf("%c", c | mask);
        mask = mask << 1;
    }
    return 0;
}
```



### 七、程序设计题（本题 15 分）

请编写一 C 语言函数 `freqs`，该函数用于计算某个长字符串(称为源字符串)中指定子串(称为目标子串)出现的频率。该函数有两个参数，第一个参数为源字符串(`source`)，第二个参数为目标子串(`destination`)；函数返回目标子串在源字符串中出现的次数。若目标子串没有出现在源字符串中，则函数返回 0。例如：若源字符串为“`abcdefgdefkdxef`”，目标子串为“`def`”，则函数返回 2。

约定：函数中需要考虑字符重复出现的情形。例如，若源字符串为“`aaaa`”，目标子串为“`aa`”，则目标子串在源字符串中出现的频率为 3。

### 八、程序设计题（本题 20 分）

FASTQ 文件是一种存储生物序列(通常称为核酸序列)以及相应质量评价的文本格式文件。整个文件包含多条生物序列，每条生物序列由四行信息组成：第 1 行为序列标识，以“`@`”开头；第 2 行是具体的生物序列(即由 A、T、G、C 组成的核酸序列)；第 3 行以“`+`”开头，后面是序列的描述信息；第 4 行是质量信息，与第 2 行的序列一一对应，每个序列均有一个质量评分，采用字符表示，该字符的 ASCII 码值即为质量值，取值为 33 到 126 的 ASCII 码(即从字符“`!`”到字符“`~`”)。下面给出的是单条生物序列的一个示例(整个文件就是由多个这样的生物序列依次存放组成的)。

```
@SEQ_ID
GATTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
+Description
!''*((((***+))%%%++)(%%%%).1***-+*''))**55CCF>>>>>CCCCCCC65
```

请根据上面的描述，编写一 C 语言程序，检查某个文件是否是合法的 FASTQ 文件。该程序接收一个文件名参数，输出为检查结论。

提示：

(1) 检查规则：以每四行为一个处理单元，在这四行中，第 1 行以“`@`”开头；第 2 行的序列只包含 A、T、G、C 四种核酸(表示核酸的四个字符大小写均可，即小写 a、t、g、c 也是合法字符，并假定序列长度不超过 2000)；第 3 行以“`+`”开头；第 4 行的长度与第 2 行相同，并且为合法的质量取值。建议分别将每个检查规则编写成独立的函数。

(2) 采用命令行参数的形式接收文件名参数。



(3) 为了简化处理逻辑, 当检查到文件中有 1 处格式不合法后, 程序即可终止, 输出文件格式不合法的提示。只有当文件中所有的内容都合法, 程序才输出文件格式合法的提示。

