

## 西南交通大学 2007 年年硕士研究生入学考试试题解析

### 试题名称：程序设计与数据结构

#### 一、填空题

1. 函数

解析：函数是一个 c 语言的基本组成单位

2. 8, 110

解析：c 语言中 1-9 开头表示十进制，以 0 开头表示 8 进制，以 0x 开头表示 16 进制

3.  $(a*a-\text{sqrt}(b)*c)/d+e$

解析：开根号 c 语言表达式为 sqrt()  $\frac{a^2 - \sqrt{b} * c}{d} + e$

4. 3.5

解析： $a \% 5 = 2, (\text{int})(x+y) = 11$ , 所以  $2 * 11 \% 3 = 1, 1/4 = 0$

5. 13

解析： $(x=0)?13:8$  表达式的含义是：若  $x=0$  为真，则表达式的值为 13，否则为 8

6. 10

解析： $a*=b++$ ; x 先执行  $a*=b$ , 然后执行  $b++$

7. 13

解析：\\,\\,\\n 分别表示三个字符，Hello world 共有十个字符，总长度为 13，计算字符串的长度不计算 \\0

8. a[3]的地址 a[0]+3 的值

解析： $*p=a$  表示将数组 a 的地址给了 p， $p+3$  表示指针顺着移动了 3，即 a[3]的地址，则  $*p$  表示值， $*p+3$  表示 a[0]+3 的值

9. 顺序存储，链式存储

解析：数据的存储结构有顺序存储，链式存储，索引，散列，常用的是前两种

10. 先进先出 先进后出

解析：栈、队列的定义

11. 以列序为主序的存储方式和以行序为主序的存储方式

解析：见课本 P92

12. 中序

解析：二叉树的前序和中序，中序和后序能唯一确定一棵二叉树

13. 7

解析： $n=n_0+n_1+n_2=2n_0+n_1-1$ , 因为共有 14 个结点，取  $n_1=1$ , 所以  $n_0=7$

14. 17

解析：在完全二叉树中，若某一结点的右孩子存在，则编号为  $2i+1$  即  $8*2+1=17$

15.  $\frac{n(n-1)}{2}$

解析：第一次比较  $n-1$  次，第二次比较  $n-2$  次……第  $n$  次比较 1 次，共  $1+2+\dots+(n-1)$   
 $=\frac{n(n-1)}{2}$

## 二、单项选择题

1. 选 B

解析：见课本 P106

2. 选 B

解析：D 错误的原因就是不能用赋值语句将一个字符常量直接给一个字符数组赋值

3. 选 B

解析：不可以嵌套定义，每一个函数都有一个内存地址，它不可以在其他函数中定义，但函数可以嵌套调用

4. 选 D

解析： $\text{int}(*\text{ptr})[10]$ ;  $\text{ptr}$  是数组指针，指向一个  $\text{int}[10]$  数组

5. 选 D

解析： $\text{int}$  占 2B，长度为十的数组占 20B，例如  $\text{int } x[] = \{2, 4, 6\}$ ，则占得字节数为  $3 * \text{sizeof}(\text{int})$

6. 选 D

解析： $\text{while}()$  括号里表达式为真，则  $\text{a}!=0$

7. 选 A

解析：全局变量其默认的存储类型是 `extern`，而局部变量是 `auto`，函数中声明的变量是局部变量

8. 选 C

解析：见课本 P336

9. 选 B

解析：B 的循环体， $P=1$  每次循环完  $P$  都被重新赋值，循环到  $i=5$ ，执行  $p*=i$ ，结果为  $p=5$

10. 选 D

解析：见 P330

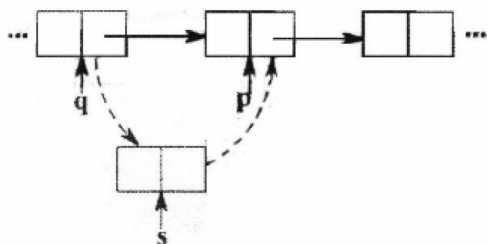
11. 选 B

解析：从第  $i+1$  个元素到第  $n$  个元素都向前移动一个位置

12. 选 B

13. 选 A

解析： $q \rightarrow \text{next} = s$ ;  $s \rightarrow \text{next} = p$ ;



14. 选 D

解析：栈空的条件是  $s.\text{top} == s.\text{base}$

15. 选 D

解析：深度为  $k$  的二叉树最多有  $2^k - 1$  个结点，即  $2^4 - 1 = 15$  个

16. 选 D

解析：二叉排序树的定义

17. 选 B

解析：画出折半查找树， $ASL_{sllac} = 74 / 20 = \frac{37}{10}$

18. 选 C

解析：P258

19. 选 B

解析：若采用快速排序，起不到它原有的效果，蜕化为起泡排序，时间复杂度为  $O(n^2)$

20. 选 A

解析：堆排序时间主要花费在建堆上，每次排序即可选出一个最小元素

### 三、阅读程序

\* \* \* \* \*

\* \* \* \*

1.

\* \* \*

\*

2.        9    7    8    13    4

3.        a b c d e f g h ?

d e f g h

4. 解析：P48

#### 四、程序填空

1.        return s

2.        a[i][i]        printf(“%d”,s)

3.        N        &a[j][i]

4.        pa&&pb        pc→next=pa        pa=pa→next        pc=pb

#### 五、简要回答问题

解析：1.

(1)  $10 \times 15 \times 20 = 3000$

(2)  $\text{loc } a[5][6][7] = 1000 + (5 \times 15 \times 20 + 6 \times 20 + 7) \times 4 = 7508$

(3)  $\text{loc } a[5][6][7] = 1000 + (7 \times 10 \times 15 + 6 \times 10 + 5) \times 4 = 5460$

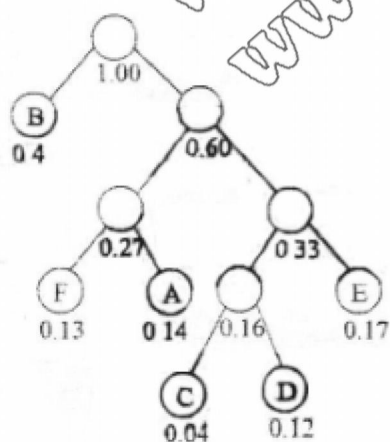
2. 解析：

①随着队头元素的不断出队，数组会产生一些空单元，入队只能在队尾进行，使得这些空单元无法使用，这时入队产生的溢出叫假溢出

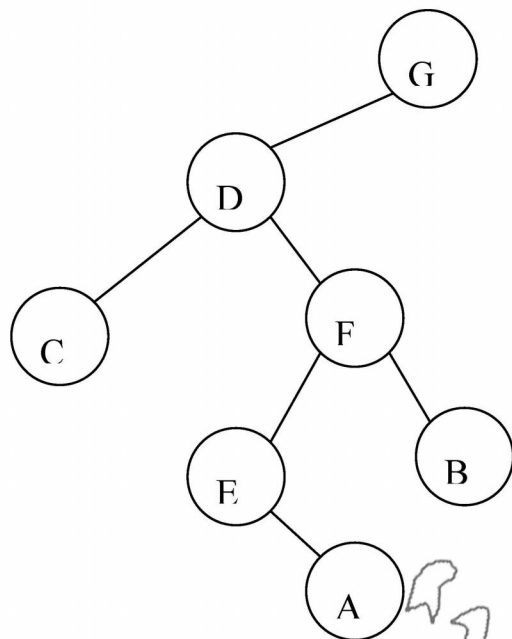
②头尾指针不断增加，已经使用过的单元无法再使用

③将队列臆造为一个环状空间，即循环队列

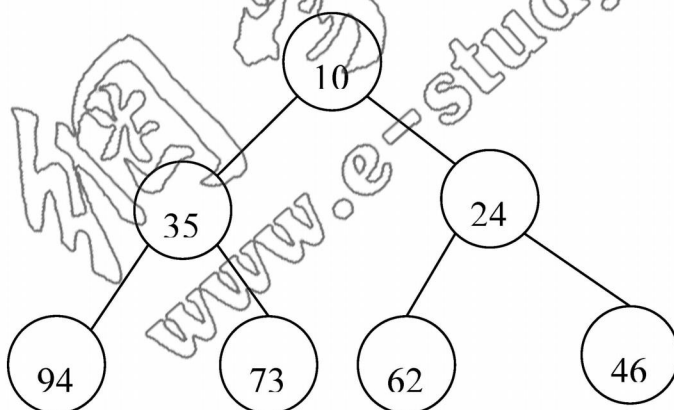
3. 解析：哈夫曼树如下：



4. 解析：二叉树如下：



5.解析：不是堆，见课表 P281，调整堆如下：（小根堆）



6.解析：构造的哈希表如下：

0	1	2	3	4	5	6	7	8	9	10	11
33	1	12	13	34	38	27	22				
1	1	2	2	4	1	2	8				

## 六、程序设计

1.解析：

```

#include "stdio.h"
#include "math.h"
    
```

```
double squard(double x,const int a)
```

```
{  
    double as=(x+a/x)/2  
    if(fabs(x-as)<0.000001)  
    {  
        return as;  
    }else  
    {  
        return squard(as,a)  
    }  
}  
int main(int argc,char*argv[])  
{  
    double x;  
    printf("请输入大于 0 的整数值");  
    scanf("%lf",&x);  
    x=squard(x,100);  
    printf("%f\n",x);  
    return 0;  
}
```

2.解析:

```
#include<stdio.h>  
#include<math.h>  
void main( )  
{  
    float s=0.0,w=0.0,f=0.0;  
    int k,r,n;  
    printf( "输入一个正整数 n: \n" );  
    scanf( "%d" ,&n);  
    for(i=1;i<n;i++)  
{
```

```
k=pow(-1, a+1);  
  
f=k*i;  
  
r=(i+1)*(i+1);  
  
W=f/r;  
  
S+=w;  
  
}  
  
printf( "%8.6f" , s);  
  
}
```

### 3. 解析:

```
#include "stdio.h"  
  
int main(int argc, char* argv[])  
{  
    int x;  
    printf("请输入一个大于等于 2 的自然数n");  
    scanf("%d", &x);  
    printf("%d=", x);  
    for(int i=2; i<=x; i++)  
    {  
        while(x%i!=0)  
        {  
            x=x/i;  
            if(x==1)  
            {  
                printf("%d", i);  
                break;  
            }  
            printf("%d*", i);  
        }  
    }  
}
```

4.解析：

```
void separate(linklistA,linklistB)
{
    LNode *r,*p,*q;
    B=(Linklist*)malloc(sizeof(LNode));
    B->next=NULL;
    r=B;p=A->next;
    while(p!=NULL&& p->next!=NULL)
    {
        q=p->next;
        if(q!=NULL)
        {
            p->next=q->next;
            r->next=q;r=q;p=p->next;
        }
        r->next=NULL;
    }
}
```

5.解析：

```
keyType predt= 32767;
int JudgeBST(BiTree bt)
{
    int b1,b2;
    if(bt==NULL)return 1;
    else
    {
        b1=JudgeBST(bt->lchild);
        if(b1==0||predt>=bt->data)return 0;
        predt=bt->data;
        b2=JudgeBST(bt->rchild);
        return b2;
    }
}
```