**王建江 20180620 程序专题**

**A 单选题（10题/教师）**

每一类题目的每道题标注考察的知识点和难易程度（难、中、易）

1、有一个有序数组 int a[]={1，4，6，10，18，35，42，53，67，71，78，84，92，99}。当用二分查找法查找键值为84的结点时，经（ B ）次比较后查找成功。

A) 4 B)3 C)2 D)12

【注】 知识点 ：查找；程度：易

2、下面程序段的时间复杂度是( A )。

s =0;

for( i =0; i<n; i++)

for(j=0;j<n;j++)

s +=B[i][j];

sum = s ;

A） O(n2) B） O(n)

C） O(m\*n) D）O(1)

【注】 知识点 ：算法分析；程度：中

3、设单向链表中指针p指向结点m，若要删除m之后的结点（若存在），则需修改指针的操作为（ A ）。

A）p->next=p->next->next; B） p=p->next;

C）p=p->next->next; D） p->next=p;

【注】 知识点 ：链表；程度：中

4、对待排序的元素序列进行划分，将其分为左、右两个子序列，再对两个子序列施加同样的排序操作，直到子序列为空或只剩一个元素为止。这样的排序方法是（ A ）。

A）直接选择排序 B）直接插入排序

C）快速排序 D）起泡排序

【注】 知识点 ：排序；程度：易

5、在一个单链表中，已知q结点是p结点的前趋结点，若在q和p之间插入s结点，则须执行（ A ） 。

A）q->next=s; s->next=p; B）s->next=p->next; p->next=s;

C）p->next=s->next; s->next=p D）p->next=s; s->next=q;

【注】 知识点 ：链表；程度：中

6、若有以下宏定义：

#define N 2

#defint Y(n) ((N+1)\*n)

则执行语句z=2\*（N+Y(5)）;后的结果是（**B**）

1. 语句有错误 B）z=34 C）z=70 D）z无定值

7、阅读下列程序，程序的运行结果为（c）。

void main()

{ struct st

{ int n;

struct st \*next ;

} a[3]={5,&a[1],7,&a[2],9,&a[0]},\*p=a;

int i;

for(i=0; i<5; i++) p=p->next;

printf("%d\n",p->n);

}

A）5 B）7 C）9 D）Null

【注】 知识点 ：结构；程度：中

8、 阅读程序，写出运行结果（**A）**。

main()

{ struct std

{ int id;

char \*name;

float sf;

} a, \*p=&a;

int i=a.id=1998;

char \*s=a.name="Windos 98";

float f=a.sf=1800;

printf( "%d is %s sal %f\n",i,s,f);

printf( "%d is %s sal %f\n",p->id,p->name,p->sf);

}

A）1998 is Windows sal 1800.00 B）1998 is Windows 98 sal 1800.00 C） 1998 is Windows 98 sal 1800 D）1998 is Windows sal 1800

【注】 知识点 ：结构、指针；程度：中

**9）阅读下面程序，写出运算结果 （ A ）。**

#include <stdio.h>

main()

{ int a[]={ 3,8,4,2 },i;

int x=0;

for( i=0; i<4; i++)

{ sub(a,x); printf(“%3d”,x); }

}

sub(int \*s, int y)

{ static int t=0;

y=s[t];

t++;

}

1. 0 0 0 0 **B) 3 8 4 2 c) 3 3 3 3 D) 3 0 0 0**

【注】 知识点 ：函数、指针、变量类型；程度：中

**10）** 使用分治法求解不需要满足的条件是（A ）。

（A）子问题必须是一样的 （B）子问题不能够重复

（C）子问题的解可以合并 （D）原问题和子问题使用相同的方法解

【注】 知识点 ：算法；程度：中

**B改错题（6题/教师）**

1 采用链表存储一个整数数列时，其各个节点的内存地址必须是连续的。【连续不连续都可以】

2 for(i=1，t=1，s=0；i<=n；i++) {t=t\*i；s=s+t；}的时间复杂度为O(n2)。【O(n)】

3 确保在动态计算一个地址时，被计算的地址是合理的并指向某个有意义的地方。特 别要保证指向一个结构或数组的内部，那么当指针增加或者改变后，可以指向不同一个结构、数组或NULL。【仍然指向同一个结构或数组】

4 由于数组与指针的等价性，可以用malloc（）来分配的内存的指针来模拟数组：

int da=(int \* **)malloc(10**\*sizeof(int)); 之后, i取值 在0～9 之间，则表达式da[i]是合法的;

【int da=(int \* **)malloc(10**\*sizeof(int));】

5 归并排序与冒泡排序的平均时间都是O（n2）. [最坏时间 ]

6 程序如下：

char \*c[] = {"ENTER", "NEW", "POINT", "FIRST"};

char \*\*cp[]= {c+3, c+2, c+1, c};

char \*\*\*cpp = cp;

main()

{

\*\*++cpp； \*--\*++cpp; \*cpp[-2]+3);

printf("%s\n", cpp[-1][-1]+1);

}

结果输出：OINT

【**EW】**