

## 一、单项选择题（每题 2 分，共 16 分）

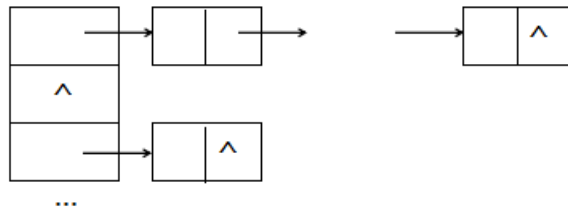
- 关于 main 函数的说明（ C ）是错误的。  
A.可以位于 C 程序的开头                      B.可以带参数  
C.能被自己所编写的程序中其余函数调用      D.一个程序中最多只能有一个
- 以下（ D ）是合法的关键字。  
A.then                      B.integer                      C.count                      D.break
- 以下（ B ）是合法的字符。  
A." a"                      B.' \017'                      C.' \0xhh'                      D.' abc'
- 下面正确的定义语句是（ C ）。  
A.char a='AB';                      B.float a=b=3.14;  
C.int a=10,\*b=&a;                      D.float \*a,b=&a;
- 下面正确的定义语句是（ D ）。  
A. int a[1][4]={1,2,3,4,5};                      B. double y[][3]= 0;  
C. long b[2][3]={{1},{1,2},{1,2,3}};                      D.float x[][3]={{1},{2},{3}};
- 当 c 的值不为 0 时，在下列选项中能正确将 c 的值赋给变量 a、b 的是（ C ）。  
A.c=b=a;                      B.(a=c) || (b=c);                      C.(a=c)&&(b=c);                      D.a=c=b;
- 有以下定义语句  
double a,b; int w; long c;  
若各变量已正确赋值，则下列选项中正确的表达式是（ C，D ）。  
A.a=a+b=b++      B.w%((int)a+b)      C.(c+w)%(int)a      D.w=a=b;
- 下面对于结构体类型的描述正确的是（ A ）。  
A.结构体类型的大小至少为各成员变量所占内存的总和  
B.结构体类型变量与简单数据类型变量的引用规则一致。  
C.结构体类型的大小为其成员中所占内存空间最大的变量的大小  
D.结构体类型的大小为其最后一个成员中所占内存空间的大小

## 二、填空题（每空 1 分，共 12 分）

- 如果用两个字节表示，-100 的八进制是（ -144 ），十六进制是（ -64 ）。
- 如果用两个字节表示，-100 的补码为（ 111111110011100 ）。
- 设有 float x=2.5,y=4.7;int a=7;请计算表达式 x+a%3\*(int)(x+y)%2/4 的值为（ 2.5 ），数据类型为（ double 或者 float ）。
- 条件“2<x<=5 或 x<-9”的 c 语言表达式为（ x>2&&x<=5||x<-9 ）。
- int (\*a)(int, int);表示（ 指向函数的指针，函数返回值为整型，参数为两个整数 ）。
- float \*b(void)表示（ 返回指针值的函数，无参，返回值类型为 float ）。
- long (\*c)[12];表示（ c 是一个指向长度 12，类型为 long 的数字指针 ）。
- char \*d="china" 表示（ d 是指向字符常量区空间里字符串 china 的指针 ）。
- 现规定一个基本整型变量占两个字节。如果有定义 struct {double a,char c}i; union {double a,char c}j;则变量 i 至少占（ 9 ）字节，变量 j 占（ 5 ）字节

## 三、简答题（每题 7 分，共 28 分）

1.桶表问题。下图为一桶表，桶表实际上是一个指针数组，共有10个数组元素，每个数组元素均为一个结构体指针（结构体名node，有两个成员一个为整型，一个为指针类型）“^”表示为空。（1）请写出node以及数组barrel的定义（3分）；（2）如果已知该数组的起始地址，采用按数组下标升序的顺序逐一访问每个桶，而每个桶按照链表本身的顺序进行访问，请写出相应的遍历算法（4分）。



2.流程图问题。请用结构化流程图（N-S图）画出第1小题第（2）问中遍历算法的流程（请注意：如果不会写则请写出一个对有100个整型元素的数组进行选择排列的程序然后画出其N-S图）。

3.折半查找问题。对于一个升序排列的整型数组可以采用折半查找的方式进行查找，即通过不断二分缩小查找范围直至找到需要寻找的数为止。该函数既可用递归的思想来写，也可用循环来写。下面请用递归思想写出该函数 find。

4.屏蔽字问题。利用屏蔽字和位运算输出整型变量 a（字长为两个字节）第 6-7 位所表示的数字。即如输入数字为 240，其二进制补码为 0000000011110000，则输出对应位表示的数字 3。要求（1）通过对 0 进行位运算得到一个屏蔽字 m（3分）；（2）通过 a 和 m 进行位运算输出相应数字（3分）。

#### 四、程序结果分析题（每题 3 分，共 12 分）

1.下面程序执行后输出为（ -12820 ）。

```
int fun(int x)
{ if (x/2)>0 fun(x/2-2);
  printf("%d",x);
}
void main()
{ fun(20);
  printf("\n");
}
```

2.执行完下列语句段后，i 值为（ 5 ）。

```
int i;
int f(int x)
{ static int k;
  x += k++;
  return x;
}
```

```

}
void main()
{ i=f(f(4));
}

```

3. 下面程序段中循环体的执行次数是（ 3 ）次。

```

a=10;b=0;
do { b=b+2;a-=2+b;} while(a>=0);

```

4. 下面程序执行后输出为（ cond ）。

```

void main()
{ char *str1[]={ "first", "second", "third", "fourth" };
  char *p1, **p2=str1;
  p1=*(p2+1)+2;
  printf("%s\n", p1);
}

```

## 五、程序设计题（共 32 分）

1. **Fibonacci 数列问题。**输出 Fibonacci 数列的前 15 项，要求每行输出 5 项。

Fibonacci 数列举例：1,1,2,3,5,8,13.....（每空 2 分，共 8 分）

```

#include <stdio.h>
void main()
{
  int fib[15], i;
  fib[0]=1;

  fib[1]=1;

  for (i=2; i<15; i++)
    fib[i]= fib[i-2]+fib[i-1];
  for(i=0; i<15; i++)
  {
    printf("%d\t", fib[i]);
    if ( i%5==4 ) printf("\n");
  }
}

```

2. 用辗转相除法求两个整数的最大公约数、最小公倍数问题。提示：GCM 为最大公约数，LCM 为最小公倍数。（每空 2 分，共 8 分）

```

#include<stdio.h>
void main()
{ int n,m,nm,r,t;
  printf("Enter m,n=?");
  scanf("%d%d",&m,&n);
  nm=n*m;
  if (m<n)
  { t=m; m=n; n=t; }
  r=m%n;
  while ( r!=0 )
  { m=n;
    n=r;
  }
}

```

```

    _____r=m%n_____;
}
printf("The max GCM=%d\n",n);
printf("The min LCM=%d\n", _____nm/n_____);
}

```

3. 游戏设计问题。现在请大家为贪食蛇游戏设计部分数据结构和算法：

(1) 贪食蛇实际上可以用一个链表来实现其数据结构，每个节点记录其 x、y 坐标和下一节点的地址，请写出贪食蛇每个节点的结构体 snake 的定义方法 (2 分)；

```

Struct snack{
    int x;
    int y;
    Struct snack *next;}

```

(2) 如果用一个整型二维数组 map 作为全局变量记录其地图，地图为一个正方形，行列数均为 N (#define N 100)，每个数组元素的值则可能为 1-果子，2-毒草，3-地雷，请写出该数组的定义 (2 分)；

```

#define N 100
Int map[N][N];

```

(3) 编写贪食蛇下一步是否碰到什么的函数

```
int check(struct snake *head,int direction)
```

其中 head 代表贪食蛇链表的表头，direction 代表蛇的运动方向 (1-4 分别代表上、下、左、右四种方向)，当下一步碰到果子则函数返回值为 1，碰到毒草则函数返回值为 2，碰到地雷则函数返回值为 3，什么都没有碰到则返回值为 0 (提示：算法需要计算下一步蛇头的位置，然后在二维数组中查找该处有什么) (4 分)；

```

int xpos,ypos,res;
xpos=head->x;
ypos=head->y;
switch(direction)
{
case 1:ypos--;break;
case 2:ypos++;break;
case 3:xpos--;break;
case 4:xpos++;
}
res=map[xpos][ypos];
return res;

```

(4) 当贪食蛇碰到地雷时实际是链表的结点去掉一半, 如果蛇本身只有一个结点则蛇死亡, 游戏失败。假设有一个全局变量 `int len`; 记录蛇的长度, 有一个全局变量 `int GameStatus`; 记录游戏状态 (0 为游戏进行, 1 为游戏失败), 则请编写一个函数实现该功能

```
struct snake *bomb(struct snake *head)
```

其中 `head` 代表贪食蛇链表的表头, 函数返回值为贪食蛇链表的表头 (4 分)。

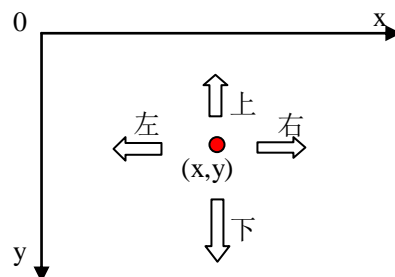
```
Int current=len;  
if(len<=1)  
{  
head=0;  
len=0;  
game status=1;  
}
```

(5) 当贪食蛇碰到食物时实际是在链表的表头增加一个结点, 则请编写一个函数实现该功能

```
struct snake *eat(struct snake *head,int x,int y)
```

其中 `head` 代表贪食蛇链表的表头, `x` 和 `y` 代表果子的坐标, 函数返回值为贪食蛇链表的表头 (4 分)。

说明: (1) 一个像素点在屏幕上的坐标(x,y)通常遵循下图所示的屏幕坐标系。



(2) 暂时不考虑贪食蛇撞到墙上及其运动时的动画处理等问题。

```
{struct snack *p;  
p=struct snack*(malloc());  
p->x=x;  
p->y=y;  
p->next=head;  
head=p;  
return head;}
```