第十三届全国青少年信息学奥林匹克联赛初赛试题

（ 提高组 C 语言 二小时完成 ）

●● 全部试题答案均要求写在答卷纸上，写在试卷纸上一律无效 ●●

一、 单项选择题 （共 10 题，每题 1.5 分，共计 15 分。每题有且仅有一个正确答案）。

1．在以下各项中，（ ）不是 CPU 的组成部分。

A．控制器 B．运算器 C．寄存器 D．主板 E．算术逻辑单元(ALU)

2．在关系数据库中，存放在数据库中的数据的逻辑结构以（ ）为主。

A．二叉树 B．多叉树 C.哈希表 D．B+树 E.二维表

3．在下列各项中，只有（ ）不是计算机存储容量的常用单位。

A．Byte B．KB C.MB D.UB E.TB

4．ASCII 码的含义是（ ）。

A．二—十进制转换码 B．美国信息交换标准代码 C．数字的二进制编码

D．计算机可处理字符的唯一编码 E．常用字符的二进制编码

5．在 C 语言中，表达式 23|2^5 的值是（ ）

A．23 B．1 C.18 D.32 E.24

6．在 C 语言中，判断 a 等于 0 或 b 等于 0 或 c 等于 0 的正确的条件表达式是（ ）

A．!((a!=0)||(b!=0)||(c!=0))

B．!((a!=0)&&(b!=0)&&(c!=0))

C．!(a==0&&b==0)||(c!=0)

D．(a=0)&&(b=0)&&(c=0)

E．!((a=0)||(b=0)||(c=0))

7．地面上有标号为 A、B、C 的 3 根细柱，在 A 柱上放有 10 个直径相同中间有孔的圆盘，从上到下依次编号为 1，2，3，……，将 A 柱上的部分盘子经过 B 柱移入 C 柱，也可以在 B 柱上暂存。如果 B 柱上的操作记录为：“进，进，出，进，进，出，出，进，进，出，进，出，出”。那么，在 C 柱上，从下到上的盘子的编号为（ ）。

A．2 4 3 6 5 7 B．2 4 1 2 5 7 C．2 4 3 1 7 6

D．2 4 3 6 7 5 E．2 1 4 3 7 5

8．与十进制数 17.5625 对应的 8 进制数是（　）。

A．21.5625　　B．21.44　　C．21.73

D．21.731　　E．前 4 个答案都不对

9．欧拉图 G 是指可以构成一个闭回路的图，且图 G 的每一条边恰好在这个闭回路上出现一次（即一笔画成）。在以下各个描述中，不一定是欧拉图的是（　　）。

A．图 G 中没有度为奇数的顶点

B．包含欧拉环游的图（欧拉环游是指通过图中每边恰好一次的闭路径）

C．包含欧拉闭迹的图（欧拉迹是指通过图中每边恰好一次的路径）

D．存在一条回路，通过每个顶点恰好一次

E．本身为闭迹的图

10．一个无法靠自身的控制终止的循环称为“死循环”，例如，在 C 语言程序中，语句“while(1) printf(“*”);”就是一个死循环，运行时它将无休止地打印*号。下面关于死循环的说法中，只有（　）是正确的。

A．不存在一种算法，对任何一个程序及相应的输入数据，都可以判断是否会出现死循环，因而，任何编译系统都不做死循环检验

B．有些编译系统可以检测出死循环

C．死循环属于语法错误，既然编译系统能检查各种语法错误，当然也应该能检查出死循环

D．死循环与多进程中出现的“死锁”差不多，而死锁是可以检测的，因而，死循环也是可以检测的

E．对于死循环，只能等到发生时做现场处理，没有什么更积极的手段

二、 不定项选择题 （共 10 题，每题 1.5 分，共计 15 分。每题正确答案的个数大于或等于 1。多选或少选均不得分）。

11．设A=B=true，C=D=false，以下逻辑运算表达式值为真的有（　）。

A．(¬A∧B)∨(C∧D∨A)　　B．¬(((A∧B)∨C)∧D)

C．A∧(B∨C∨D)∨D　　D．(A∧(D∨C)) ∧B

12．命题“P→Q”可读做P蕴涵Q，其中P、Q 是两个独立的命题。只有当命题P成立而命题Q不成立时，命题“P→Q”的值为false，其他情况均为true。与命题“P→Q”等价的逻辑关系式是（　）。

A．¬P∨Q　　B．P∧Q　　C．¬(P∨Q)　　D．¬(¬Q∧P)

13．$(2070)_{16}$ + $(34)_8$的结果是（　）。

A．$(8332)_{10}$　　B．$(208C)_{16}$

C．$(100000000110)_2$　　D．$(20214)_8$

14．已知 7 个结点的二叉树的先根遍历是 1 2 4 5 6 3 7（数字为结点的编号，以下同），后根遍历是 4 6 5 2 7 3 1，则该二叉树的可能的中根遍历是（　）

A．4 2 6 5 1 7 3　　B．4 2 5 6 1 3 7

C．4 2 3 1 5 4 7　　D．4 2 5 6 1 7 3

15．冗余数据是指可以由其他数据导出的数据，例如，数据库中已存放了学生的数学、语文和英语的三科成绩，如果还存放三科成绩的总分，则总分就可以看作冗余数据。冗余数据往往会造成数据的不一致，例如，上面 4 个数据如果都是输入的，由于操作错误使总分不等于三科成绩之和，就会产生矛盾。下面关于冗余数据的说法中，正确的是（　）。

A．应该在数据库中消除一切冗余数据

B．与用高级语言编写的数据处理系统相比，用关系数据库编写的系统更容易消除冗余数据

C．为了提高查询效率，在数据库中可以适当保留一些冗余数据，但更新时要做相容性检验

D．做相容性检验会降低效率，可以不理睬数据库中的冗余数据

16．在下列各软件中，属于 NOIP 竞赛（复赛）推荐使用的语言环境有（　）。

A．gcc　　B．g++

C．Turbo C　　D．free pascal

17．以下断电之后仍能保存数据的有（　）。

A．硬盘　　B．ROM　　C．显存　　D．RAM

18．在下列关于计算机语言的说法中，正确的有（　）。

A．高级语言比汇编语言更高级，是因为它的程序的运行效率更高

B．随着Pascal、C等高级语言的出现，机器语言和汇编语言已经退出了历史舞台

C．高级语言程序比汇编语言程序更容易从一种计算机移植到另一种计算机上

D．C是一种面向过程的高级计算机语言

19．在下列关于算法复杂性的说法中，正确的有（　）。

A．算法的时间复杂度，是指它在某台计算机上具体实现时的运行时间

B．算法的时间复杂度，是指对于该算法的一种或几种主要的运算，运算的次数与问题的规模之间的函数关系

C．一个问题如果是NPC类的，就意味着在解决该问题时，不存在一个具有多项式时间复杂度的算法。但这一点还没有得到理论上的证实，也没有被否定

D．一个问题如果是NP类的，与C有相同的结论

20．近20年来，许多计算机专家都大力推崇递归算法，认为它是解决较复杂问题的强有力的工具。在下列关于递归算法的说法中，正确的是（　）。

A．在1977年前后形成标准的计算机高级语言“FORTRAN77”禁止在程序使用递归，原因之一是该方法可能会占用更多的内存空间

B．和非递归算法相比，解决同一个问题，递归算法一般运行得更快一些

C．对于较复杂的问题，用递归方式编程往往比非递归方式更容易一些

D．对于已经定义好的标准数学函数 sin(x)，应用程序中的语句“y=sin(sin(x));”就是一种递归调用

三．问题求解（共 2 题，每题 5 分，共计 10 分）

1．给定 n 个有标号的球，标号依次为 1，2，…，n。将这 n 个球放入 r 个相同的盒子里，不允许有空盒，其不同放置方法的总数记为 S(n,r)。例如，S(4,2)=7，这 7 种不同的放置方法依次为 {(1),(234)}, {(2),(134)}, {(3),(124)}, {(4),(123)}, {(12),(34)}, {(13),(24)}, {(14),(23)}。当 n=7,r=4 时，S(7,4)= _____。

2．N 个人在操场里围成一圈，将这 N 个人按顺时针方向从 1 到 N 编号，然后，从第一个人起，每隔一个人让下一个人离开操场，显然，第一轮过后，具有偶数编号的人都离开了操场。依次做下去，直到操场只剩下一个人，记这个人的编号为 J(N)，例如，J(5)=3，J(10)=5，等等。则 J(400)=_____。

（提示：对 $N=2^m+r$ 进行分析，其中 $0 \leq r<2^m$ ）。

四．阅读程序写结果（共 4 题，每题 8 分，共计 32 分）

1.
```
#include <stdio.h>
int main()
{int i,p[5],q[5],x,y=20;
  for(i=0;i<=4;i++)
    scanf("%d", &p[i]);
  q[0]=(p[0]+p[1])+(p[2]+p[3]+p[4])/7;
  q[1]=p[0]+p[1]/((p[2]+p[3])/p[4]);
  q[2]=p[0]*p[1]/p[2];
  q[3]=q[0]*q[1];
  q[4]=q[1]+q[2]+q[3];
  x=(q[0]+q[4]+2)-p[(q[3]+3)%4];
  if(x>10)
    y+= (q[1]*100-q[3])/(p[p[4]%3]*5);
  else
    y+=20+(q[2]*100-q[3])/(p[p[4]%3]*5);
  printf("%d,%d\n", x,y);
  return 0;
```

```
 }
```

/*注：本例中，给定的输入数据可以避免分母为 0 或数组元素下标越界。 */

输入：6 6 5 5 3

输出：_____

```
2. #include <stdio.h>
void fun(int *a,int *b)
{int *k;
 k=a; a=b; b=k;
}
main( )
{int a=3,b=6,*x=&a,*y=&b;
 fun(x,y);
 printf("No.1: %d,%d ",a,b);
 fun(&a,&b);
 printf("No.2: %d,%d\n",a,b);
}
```

输出:_____

```
3. #include "math.h"
#include "stdio.h"
main()
{int a1[51]={0};
 int i,j,t,t2,n=50;
 for (i=2;i<=sqrt(n);i++)
   if(a1[i]==0)
    {t2=n/i;
     for(j=2;j<=t2;j++) a1[i*j]=1;
    }
 t=0;
 for (i=2;i<=n;i++)
  if(a1[i]==0)
    {printf("%4d",i);  t++;
     if(t%10==0) printf("\n");
    }
 printf("\n");
}
```

输出：_____

_____

4.
```
#include "stdio.h"
char ch[]={'q','A','S','O','R','T','E','X','A','M','P','L','E'};
int n=12;
void shift(int k, int n)
 {char v;
  int j;
  v=ch[k]; j=k+k;
  while (j<=n)
    {if((j<n) && (ch[j]<ch[j+1])) j++;
     if (v<ch[j])
       { ch[j/2]=ch[j]; j*=2; }
     else
       return;
     ch[j/2]=v;
    }
 }
void hpsrt(void)
 {int k;
 char tmp;
 for (k=n/2; k>0; k--)  shift(k,n); /* 建堆 */
 printf("No.1: ");
 for(k=1; k<=n; k++)  putchar(ch[k]);
 putchar('\n');
 for (k=n; k>0; k--)
   { tmp=ch[1]; ch[1]=ch[k]; ch[k]=tmp;
     shift(1,k-1);
    }
 }
main()
{int k;
 hpsrt();
 printf("No.2: ");
 for(k=1; k<=n; k++)  putchar(ch[k]);
 putchar('\n');
}
```

输出：_____

_____

五．完善程序（前 5 空，每空 2 分，后 6 空，每空 3 分，共 28 分）

1.（格雷码，Gray Code）

格雷码是对十进制数的一种二进制编码。编码顺序与相应的十进制数的大小不一致。其特点是：对于两个相邻的十进制数，对应的两个格雷码只有一个二进制位不同。另外，最大数与最小数之间也仅有一个二进制位不同，以 4 位二进制数为例，编码如下：

十进制数	格雷码	十进制数	格雷码
0	0000	8	1100
1	0001	9	1101
2	0011	10	1111
3	0010	11	1110
4	0110	12	1010
5	0111	13	1011
6	0101	14	1001
7	0100	15	1000

如果把每个二进制的位看作一个开关，则将一个数变为相邻的另一个数，只须改动一个开关。因此，格雷码广泛用于信号处理、数-模转换等领域。

下面程序的任务是：由键盘输入二进制数的位数 n (n<16)，再输入一个十进制数 m(0≤m<2^n)，然后输出对应于 m 的格雷码（共 n 位，用数组 gr[]存放）。

为了将程序补充完整，你必须认真分析上表的规律，特别是对格雷码固定的某一位，从哪个十进制数起，由 0 变为 1，或由 1 变为 0。

```
#include <stdio.h>
main()
{int bound=1,m,n,i,j,b,p,gr[15];
 printf("input n,m\n");
 scanf("%d%d",&n,&m);
 for(i=1;i<=n;i++) bound= ① ;
 if(m<0||m>=bound)
   {printf("Data error!\n");
     ②  ;
   }
 b=1;
 for(i=1;i<=n;i++)
  {p=0; b=b*2;
```

```
    for( ③ ;j<=m;j++)
      if( ④ )
       p=1-p;
     gr[i]=p;
    }
   for(i=n; ⑤ )
    printf("%1d",gr[i]);  /* 在 "%1d" 中出现的是数字 1，不是字母 l */
   printf("\n");
  }
```

2.（连续邮资问题）某国发行了 n 种不同面值的邮票，并规定每封信最多允许贴 m 张邮票，在这些约束下，为了能贴出{1，2，3,…，maxvalue}连续整数集合的所有邮资，并使 maxvalue 的值最大，应该如何设计各邮票的面值？例如，当 n=5、m=4 时，面值设计为{1，3，11，15，32}，可使 maxvalue 达到最大值 70（或者说，用这些面值的 1 至 4 张邮票可以表示不超过 70 的所有邮资，但无法表示邮资 71。而用其他面值的 1 至 4 张邮票如果可以表示不超过 k 的所有邮资，必有 k≤70）。

下面是用递归回溯求解连续邮资问题的程序。数组 x[1:n]表示 n 种不同的邮票面值，并约定各元素按下标是严格递增的。数组 bestx [1:n]存放使 maxvalue 达到最大值的邮票面值（最优解），数组 y[maxl]用于记录当前已选定的邮票面值 x[1:i]能贴出的各种邮资所需的最少邮票张数。请将程序补充完整。

```
#include <stdio.h>
#define NN 20
#define maxint 30000
#define maxl 500  /*邮资的最大值*/
int n,m,bestx[NN],x[NN],y[maxl],maxvalue=0;
void result()
{输出结果：最大值：maxvalue  及 最优解： bestx[1:n]（略）
}
void backtrace(int i,int r)
{ int j,k,z[maxl];
 for(j=0;j<= ① ;j++)
   if(y[j]<m)
     for(k=1;k<=m-y[j];k++)
      if(y[j]+k<=y[ ② ])
        y[ ③ ]=y[j]+k;
    while(y[r]<maxint) r++;
  if(i>n)
  {if(r-1>maxvalue)
   {maxvalue= ④ ;
```

```
    for(j=1;j<=n;j++)
     bestx[j]=x[j];
   }
   return;
  }
for(k=0;k<maxl;k++)
  z[k]=y[k];
 for(j= ⑤ ;j<=r;j++)
 {x[i]=j;
  ⑥ ;
  for(k=0;k<maxl;k++)
  y[k]=z[k];
 }
}
void main()
{int j;
 printf("input n,m:\n");
 scanf("%d%d",&n,&m);
 for(j=1;j<maxl;j++)
  y[j]=maxint;
 y[0]=0;  x[0]=0; x[1]=1;
 backtrace(2,1);
 result();
}
```