

2024 北京市中小学信息学能力测评活动第一轮

基础知识测评（初中组）

一、单选题（共 15 题，每题 2 分，共计 30 分）

- 1、计算机的输入设备是（ ）。
A. 显示器 B. 键盘 C. 磁盘 D. 打印机
- 2、将十进制数 12.3125 化为二进制数为（ ）。
A. 1100.1 B. 1011.011 C. 1100.0101 D. 1100.101
- 3、某台计算机的内存容量为 640K，这里的 640K 容量是指（ ）字节
A. 640 个 B. 640*1000 个 C. 640 * 1024 个 D. 640*1024*1024 个
- 4、以下断电之后仍能保存数据的是（ ）。
A、硬盘 B、寄存器 C、显存 D、内存 E、高速缓存
- 5、在下面各世界顶级的奖项中，为计算机科学与技术领域做出杰出贡献的科学家设立的奖项是（ ）。
A、沃尔夫奖 B、诺贝尔奖 C、菲尔兹奖 D、图灵奖
- 6、在某段程序中，有 3 个整型变量 a、b、c，执行完以下代码段后，变量 c 的值是（ ）
a=15;
b=a*3;
a++;
c=a+b/2;
A、15 B、38 C、38.5 D、16
- 7、下面循环结构中，循环体执行的次数为（ ）次
int n=315;
while(n>0)
n=n/3;
A、无数 B、4 C、6 次 D、5 次
- 8、在 Linux 系统中，比较两个文件的有无不同之处的命令是（ ）
A、cmd B、diff C、fc D、cmp
- 9、对于入栈顺序是“a, b, c, d, e”的序列，下面（ ）不可能是合法的出栈序列
A、a, b, c, d, e B、d, b, c, a, e
C、b, a, d, c, e D、c, b, a, d, e
- 10、线性表若采用链表存储结构，要求内存中可用存储单元地址（ ）
A、必须连续 B、部分地址必须连续 C、一定不连续 D、连续不连续均可
- 11、对于一棵二叉树，它的中序遍历结果为“LKMACBFEHGID”，层次遍历结果为“AKBLMCEFGHI”，那么这棵二叉树对应的前序遍历结果为（ ）
A、AKLMBCDEFGHI B、ALMKBDECFHIG
C、LABMDCHGIKEF D、ALKMCBDEFGHI
- 12、一棵完全二叉树有 72 个结点，那么它的叶子结点有（ ）个
A、30 B、27 C、36 D、39

- 13、对于表达式 $a+b*(c+d)*e$ 的后缀形式是 ()
A、abcd*+*e+ B、a+b*c+d*e C、abcd+*e*+ D、bcda+e*+*
- 14、以下排序算法中，不需要进行关键字比较操作的算法是 ()
A、快速排序 B、冒泡排序 C、归并排序 D、基数排序
- 15、5 个没有区别的结点构成不同形态的二叉树的种类有 () 个
A、30 B、35 C、42 D、40

二、问题求解（共 2 题，每题 6 分，共计 12 分，每题有且仅有一个正确选项）

16、深夜四个人 ABCD 过桥，一次最多能过两个人，他们只有一个手电，每次通过的人中必须要有人带着手电，每个人所需的时间分别为 2 分钟、3 分钟、5 分钟、10 分钟；求最快可以多长时间全部过桥？

答：() 分钟

- A、18 B、22 C、21 D、24

17、田忌准备和齐王赛马，各自拿出的 11 匹马来比赛，胜负由每匹马的速度决定，速度大的赢，相等则平局。田忌可以安排自己的哪一匹马跟齐王的哪一匹马比赛，每匹马只能赛一次，并且必须赛一次。田忌赢一次赏金加 50 钱，输一次赏金减 50 钱，平局则不加不减。例如田忌有一匹马速度为 3，齐王有一匹马速度为 2，则这两匹马比赛，田忌能赚 50 钱。现在田忌的 11 匹马速度为 (4 7 5 6 10 9 9 1 5 3 6)，齐王的 11 匹马速度为 (7 2 9 1 1 3 1 8 1 3 11)，问求田忌最多能赚多少赏金钱？

答：田忌最多能赚 () 钱。

- A、450 B、300 C、200 D、500

三、看程序写结果（共 4 题，每题 7 分，共计 28 分，每题有且仅有一个正确选项）

18、

```
#include<iostream>
using namespace std;
int main() {
    int a,b,c,d;
    cin>>a;
    b=a/3600;
    c=a%3600/60;
    d=a%60;
    cout<<b<<" "<<c<<" "<<d;
    return 0;
}
```

输入：123456

输出：()

- A、34 18 26 B、36 17 34 C、34 17 36 D、30 18 33

19、

```
#include <bits/stdc++.h>
using namespace std;
int main() {
```

```

int a,i;
bool f=true;
cin>>a;
cout<<a<<"=";
for(i=2;i<=a;i++)
{
    while(a%i==0)
    {
        if(f==true)
        {
            cout<<i;
            f=false;
        }
        else
            cout<<"*"<<i;
        a=a/i;
    }
}
return 0;
}

```

输入：300

输出：（ ）

A、 $300=2*2*3*5*5$ B、 $2*2*5*5$ C、 $300=2*10*15$ D、 $300=2*2*5*5*3$

20、

```

#include <bits/stdc++.h>
using namespace std;
int main()
{
    int a[105]={0},b[105]={0};
    int n,k,t,i,j,num;
    cin>>n;
    for(i=0;i<n;i++)
        cin>>b[i];
    for(num=0;num<n;num++)
    {
        for(i=n-1;b[i]!=0;i--);
        a[i]=num;
        b[i]=-1;
        for(j=i+1;j<n;j++)
            b[j]--;
    }
}

```

```

    for(i=0;i<n;i++)
        cout<<a[i]<<" ";
    return 0;
}

```

输入: 8

0 1 1 1 2 3 5 7

输出: ()

A、1 4 7 2 3 5 0

B、0 6 4 1 2 3 5 7

C、0 6 3 0 6 4

D、0 0 5 1

2 3 9 2

21、

```

#include <bits/stdc++.h>
using namespace std;
int a[6][6], s[6];
bool v[6], f;
void dfs(int x)
{
    if (x == 6)
    {
        f = 1;
        for (int i = 1; i <= 5; i++)
            cout << char(s[i] + 'A' - 1) << ' ';
        cout << endl;
    }
    else
    {
        for (int i = 1; i <= 5; i++)
        {
            if (a[x][i] == 1 && v[i] == 0)
            {
                s[x] = i;
                v[i] = 1;
                dfs(x + 1);
                v[i] = 0;
            }
        }
    }
    return ;
}

```

```

int main() {
    for (int i = 1; i <= 5; i++)
        for (int j = 1; j <= 5; j++)

```

```

        cin >> a[i][j];
    dfs(1);
    if (f == 0)
        cout << "no";
    return 0;
}

```

输入: 1 0 1 1 0
 0 0 0 0 1
 0 0 1 0 1
 1 1 0 1 1
 0 0 0 1 0

输出: ()

A、C B E A D

B、A E C D B

C、C A E D Q

D、A E C B D

四、完善程序（共 2 大题 10 小题，每小题 3 分，共计 30 分，每题有且仅有一个正确选项）

（一）、（输出排列）给定一个长为 n 的数组 a （不保证元素互不相同），找出它的全排列里字典序不小于它的所有不同排列，并把它们按照字典序从小到大输出。下面的程序会对于每个排列，先输出它是这些排列中字典序第几大的，然后再输出它的每个元素。

请补全程序。

```

#include <bits/stdc++.h>
using namespace std;
int a[105];
int main()
{
    int n, i, j, c = 1;
    scanf("%d", &n);
    for(i = 1; i <= n; i++)
        scanf("%d", a + i);
    while(true)
    {
        printf("%d: ", ①);
        for(i = 1; i <= n; i++)
            printf("%d ", a[i]);
        printf("\n");
        for(②)
            if(a[i] < a[i + 1])
                break;
        if(i == ③)
            break;
        for(j = n; j > i; j--)
            if(④)

```

```

        swap(a[j], a[n + i + 1 - j]);
    for(j = i + 1; j <= n; j++)
        if(a[j] > a[i])
        {
            ⑤;
            break;
        }
    }
    return 0;
}

```

22、①处应填（ ）

- A. `c`
- B. `c+1`
- C. `++c`
- D. `c++`

23、②处应填（ ）

- A. `i = 1; i < n; i++`
- B. `i = n - 1; i >= 1; i--`
- C. `i = 1; a[i] != a[n]; i++`
- D. `a[i] = a[n - 1]; i >= 1; i--`

24、③处应填（ ）

- A. `0`
- B. `1`

C. `n`

D. `n-1`

25、④处应填 ()

A. `j != n + i + 1 - j`

B. `j == n + i + 1 - j`

C. `j > n + i + 1 - j`

D. `a[j] > a[n + i + 1 - j]`

26、⑤处应填 ()

A. `a[j] = a[i]`

B. `swap(a[i], a[j])`

C. `swap(i, j)`

D. `a[i] = a[j]`

(二)、(二进制位加权和)给定一个 64 位二进制数 x , 假设 x 的二进制表示是 $a_{64}a_{63}\cdots a_1$, 即 $x=2^{63}a_{64}+2^{62}a_{63}+\cdots+a_1$, 求出 $\sum a_i$ 加起来的和。

下面程序将使用倍增法解决该问题, 我们以“求 x 二进制表示下 1 的个数”这一问题为例介绍倍增法的思想。

在一开始, 我们把给定的 x 看作是分成了 64 块, 每块长度为 1, 每个块内的二进制数表示原数这个块内的 1 的个数。

我们将进行若干轮操作, 每轮合并两个相邻的块, 让块长翻倍, 合并成的新块上存的二进制数就是两个原块上存的二进制数加起来。

这个示例代码可以对 $0 \sim 2^{64}-1$ 的 x 求出 1 的个数:

```
#include <bits/stdc++.h>
using namespace std;
```

```

typedef unsigned long long u64;
int main()
{
    u64 x;
    scanf("%llu", &x);
    x -= (x & 0xaaaaaaaaaaaaaaaaull) >> 1;
    x = (x & 0x3333333333333333ull) + ((x & 0xffffffffffffffffull) >> 2);
    x = (x & 0x0f0f0f0f0f0f0f0fu1l) + ((x & 0xf0f0f0f0f0f0f0f0ull) >> 4);
    x = (x & 0x00ff00ff00ff00ffu1l) + ((x & 0xff00ff00ff00ff00ull) >> 8);
    printf("%llu\n", x);
    return 0;
}

```

请补全程序：

```

#include <bits/stdc++.h>
using namespace std;
typedef unsigned long long u64;
int main()
{
    u64 i, x, y, P, Q, K, R;
    scanf("%llu", &x);
    y = ①;
    P = ②;
    Q = 1;
    for(i = 0; i < 6; i++)
    {
        K = ③;
        P /= K;
        R = P * Q;
        y = (y & R) + ④;
        y += ⑤;
        x = (x & R) + ((x >> (1 << i)) & R);
        Q *= K;
    }
    printf("%llu\n", y);
    return 0;
}

```

27、①处应填 ()

A. 0

B. x

C. 1

D. -1

28、②处应填 ()

A. 0

B. y

C. -1

D. 1

29、③处应填 ()

A. (1 << (1 << i))

B. (1 << i) + 1

C. i

D. (1ull << (1 << i)) + 1

30、④处应填 ()

A. ((y >> (1 << i)) & R)

B. ((y >> i) & Q)

C. (y & R)

D. $((y \wedge Q) \& R)$

31、⑤处应填 ()

A. $((x \gg (1 \ll i)) \& R) \ll (1 \ll i)$

B. $((x \gg (1 \ll i)) \& R) \ll i$

C. $((x \gg i) \& R) \ll i$

D. x

-----题目结束-----
