

# 初赛知识点分块模拟题库

## 系统基础专题模拟题

1.世界上第一台计算机的名字是：（）

A ENIAC 阿尼亚克    B Hulk 浩克

C Deep Blue 深蓝    D AlphaGo 阿尔法狗

2.以下关于图灵测试的说法正确的是（）

A 测试要在软件的接口处进行，测试时不考虑程序内部的逻辑结构和内部特性。

B 对软件的过程性细节做细致的检查。

C 制作两个或多个版本，同一时间维度，分别让组成成分相同的访客随机的访问这些版本，最后分析、评估出最好版本，正式采用。

D 指测试者与被测试者（一个人和一台机器）隔开的情况下，随意提问，检测被测试者是否拥有智能。

3.以下 不属于 输入设备的是（）

A 扫码枪    B 触摸板    C 立体环绕音响    D 指纹识别按钮

4.以下 属于 输出设备的是（）

A 机械键盘    B 蓝牙耳机    C 固态硬盘    D 扫描仪

5.断电后数据会消失的存储设备（）

A 移动硬盘 B ROM C RAM D 光盘

6.以下 不属于 微软公司产品的是 ( )

A Window10 B Linux C Visual Studio D Powerpoint

7.以下关于编程语言说法 错误 的是 ( )

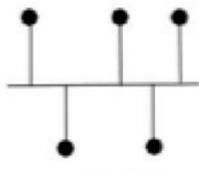
A 机器语言是计算机能够直接识别的语言。

B 高级语言不能直接运行，需要在编译或解释后才能运行。

C 在不同的设备中，汇编语言对应着不同的机器语言指令集，需要通过相应的汇编程序转换后，才能运行。

D 面向过程的编程语言不是高级语言。

8. 下图所示的网络结构属于哪种拓扑结构 ( )



A 环形网络 B 星型网络 C 网状网络 D 总线型网络

9.IP 地址 192.168.1.1 属于哪一类 IP 地址 ( )

A A 类地址 B B 类地址 C C 类地址 D D 类地址

10. 百度公司的域名是 www.baidu.com，用来提供搜索服务的其中一个服务器的 IP 地址是 220.181.38.149，建立 IP 地址与域名映射关系的网络协议是：( )

A HTTP B DNS C TCP D SMTP

11. 以下数字哪个和其他不一样 ( )

A 1234    B 1.234E+03    C 1.23e3+4    D 1.2e2+34

12 黑白照片如果每个像素由 1 位构成,就只能表示纯黑色和纯白色两种颜色。8 位的灰度照片除了黑白还包含了很多不同程度的灰色,请问这种灰度照片一共可以表示( )种颜色呢?

A 8 种    B 16 种    C 128 种    D 256 种

13 设计了一套包含了 1 万个字的字体存储在点阵 32\*32 的字模上大约需要多少空间? ( )

A 1.2MB    B 24KB    C 0.2GB    D 720B

14 《量子破碎》是一款微软工作室发行的游戏,容量高达 180GB,是电影和游戏的首次结合,仅仅在人物剧情上就达到了 80 多个 GB,把他称为“硬盘终结者”再合适不过了,现在 5G 技术的峰值传输速度为 20Gbps。如果在最理想的情况下载这款游戏,需要多少长时间 ( )。

A 4s    B 9s    C 32s    D 72s

15. NOIP2022 年开始只支持哪种语言。( )

A C 语言    B Java 语言    C Python 语言    D C++语言

## 数学专题

1.下面是四个不同进制的数,最大的一个是 ( )

A.(1AB)<sub>16</sub>    B.(456)<sub>10</sub>    C.(654)<sub>8</sub>    D.(101011011)<sub>2</sub>

2.二进制数 101.11 转换为十进制数是 ( )

A.5.3 B. 5.5 C. 5.75 D. 5.125

3.八位机器码 11100010 来表示顶点整数 (最高位为符号位), 当它是原码是表示的十进制值为-98; 当它是补码是表示的十进制值是 ( )

A.-98 B.-97 C.-31 D.-30

4.二进制数-1001010 的补码是 ( )

A.0010101 B.10110110 C.10010101 D.01101010

5.用十六进制、八进制和十进制写了一个等式:  $52-19=33$ , 式中三数的进制数各不相同, 则 52、19、33 有可能分别为 ( )

A.十进制, 十进制, 十进制

B.八进制, 十六进制, 十进制

C.十进制, 十六进制, 八进制

D.十进制, 八进制, 十六进制

6.十进制数 100 的反码和补码表示分别是 ( )

A.9BH 和 64H B.64H 和 9BH

C.64H 和 64H D.9BH 和 9BH

7.一个有符号单字节的最小二进制是 ( )

A.10000000 B.11111111 C.01111111 D.00000000

8.已知  $x=11001010B$ ,  $y=00001111B$ ,  $z=01011100B$ , 那么  $x|y\&z=$  ( )

A.11001110B B.01110110B

C.11101110B D.01001100B

9.当  $A \geq B \& \& B \geq C$  的取值为真时，那么表达式  $A > C || B == C$  的值为 ( )

A.为真 B. 有可能为假

C. 无法判定结果真假 D. 只有 A、B、C 都为正数时才为真

10.某社区有 30 个人，可以报名参加 A, B, C 三种活动。已知有 6 人报名 A 活动，8 人报名 B 活动，15 人报名 C 活动，且其中有 3 人报名全部活动，则至少有 ( ) 人没有报名任何活动。

A.5 B.7 C.9 D.10

11.若今天是星期四，从今天起第 10 天后是星期日， $10^{2021}$  天后那一天是星期 ( )

A.一 B.二 C.六 D.日

12.在所有三位数中，各位数字从高位到低位顺次减小的数共有 ( ) 个

A.120 B.91 C.88 D.137

13.一个商场有  $m$  种小球，每种小球有无数个，现要选出  $n$  个小球，如当  $m=2$ ,  $n=3$  时，有 4 种选择方案:  $\{0, 3\}$ ,  $\{1, 2\}$ ,  $\{2, 1\}$ ,  $\{3, 0\}$ ，问，当  $m=4$ ,  $n=4$  时，有 ( ) 种选择方案？

A.33 B.34 C.35 D.36

14.从 1 到 10 中有两个数 1, 7 不能被 2, 3, 5 整除，那么 1~1000 中有 ( ) 个不能被 2,

3, 5 整除?

A.266 B.267 C.268 D.269

15.从 9~100 之间的质数个数有 ( ) 个

A.19 B.20 C.21 D.22

16.有一个数列: 1, 2, 5, 12, 29.....第 8 项为 ( )

A.171 B.256 C.387 D.408

## C++编程语言部分

### 栈队列链表板块

1.如果一个栈初始时空, 入栈顺序为 a,b,c,d.则出栈顺序有\_\_\_种;

2.已知某个栈的出栈顺序是 ABECD,则下列入栈顺序可能是( ) 不定项选择题,

A.ABECD

B.BEACD

C.DCEBA

D.BAECD

3.如果一个队列初始时空, 入队顺序为 a,b,c,d.则出栈顺序有\_\_\_种;

4.有一链表如图:我们只知道节点 x 的信息:

即 `lin[x].data, lin[x].next ,lin[x].pre;`

至少写出两种从链表中删除节点 x 的正确顺序是\_\_\_\_\_ (排序,不保证都选)

1.link[x].next=link[x].pre=0;

2.link[link[x].pre].next=link[x].next;

3.link[link[x].next].pre=link[x].pre;

4.link[link[x].pre].pre=link[x].next;

5.设某算法的计算时间表示为递推关系式  $T(N)=2T(N/2)+1$  ( $n$  为正整数) 及  $T(1)=0$ , 则该算法的时间复杂度为\_\_\_\_\_。

6.下列代码的时间复杂度是()。

```
for(int i=1;i<=N;++i)

    for(int j=i;j<=N;j+=i)

        cout<<"a";
```

A. $O(N \ln N)$       B. $O(N^2)$       C. $O(N \cdot \sqrt{N})$       D. $O(N)$

## 树论

1、具有 12 个结点的完全二叉树有 ( ) 个度为 2 的结点

A.5    B.4    C.6    D.3

2、满二叉树中所有结点个数是 ( ), 其中  $k$  是树的深度

A.    B.    C.    D.

3、由 3 个结点所构成的二叉树有 ( ) 种形态

A.2 B.3 C.4 D.5

4、设一棵完全二叉树有 700 个结点，则共有 ( ) 个叶子结点

A.349 B.350 C.351 D.352

5、设一棵完全二叉树具有 1000 个结点，则此完全二叉树有 ( ) 个叶子结点，有 ( ) 个度为 2 的结点，有 ( ) 个结点只有非空左子树，有 ( ) 个结点只有非空右子树。

A.499 500 0 0 B.499 500 1 0 C.500 499 1 0 D.500 499 0 0

6、如果根结点的深度记为 1，则一棵恰有 249 个叶结点的二叉树的深度最少是 ( )

A.6 B.7 C.8 D.9

7、一棵具有 224 个叶子结点的完全二叉树，最多有 ( ) 个结点

A.448 B.449 C.450 D.451

8、若一棵完全二叉树中某结点无左孩子，则该节点一定是( )。

A.度为 1 的结点 B.度为 2 的结点 C.分支节点 D.叶子结点

9、一棵结点数为 2021 的完全二叉树，则最多有多少个叶子结点？

A.1009 B.1010 C.1011 D.1012

10、完全二叉树采用顺序存储方式，顺序方式是指从根节点开始从上往下，从左往右，一次存入到顺序结构的数组中。假定根节点存放的下标是 0 号位置，那么第 2020 号的父节点存在的情况下，应当存放在 ( ) 号。

A.1009 B.1010 C.1011 D.1012

11、完全二叉树采用顺序存储方式，顺序方式是指从根节点开始从上往下，从左往右，一次存入到顺序结构的数组中。假定根节点存放的下标是 1 号位置，那么第 2021 号的父节点存在的情况下，应当存放在（ ）号。

A.1009 B.1010 C.1011 D.1012

12、二叉树的先序遍历为 F B A C D E G H，中序遍历 A B D C E F G H，则后序遍历为

A.A D E C B H G F

B.A B D E C G H F

C.G H A D E C B F

D.H G A D E C B F

13、一棵二叉树的中序遍历为 ABC，则先序遍历不可能的是（ ）。

A.BAC B. ABC C. CBA D. ACB

14、后序遍历序列与中序遍历序列相同的二叉树为（ ）。

B.根结点无左子树的二叉树

C.根结点无右子树的二叉树

D.只有根结点的二叉树或非叶子结点只有左子树的二叉树

E.只有根结点的二叉树或非叶子结点只有右子树的二叉树

15、二叉树的后序遍历为 D E C B G F A，中序遍历 B D C E A F G，则先序遍历为

A.A B C D E F G

B.A B C D E F G

C.B C A E F D G

D.C A B E F G D

16、二叉树中 ( ) 第二个访问根节点

A.先序遍历 B.中序遍历 C.后序遍历 D.以上都是

17、关于哈夫曼树的说法，以下错误的是 ( )

A.权值越大的叶结点越靠近根节点

B.权值越小的叶结点越远离根节点

C.有度为 1 的点

D.有  $n_0$  个叶子结点的哈夫曼树，结点个数有  $2*n_0-1$  个

18、以下那一组是哈夫曼编码 ( )

A. 100 001 10 1

B. 110 001 10 1

C. 111 101 11 0

D. 100 101 11 0

19、给定以下权值  $w=\{5,29,7,8,14,23,3,11\}$ ，构建一个哈夫曼树，规定左分支为 0，右分

支为 1。那么权值为 5 的哈夫曼编码是多少？

A. 0000 B. 0001 C. 0011 D. 0100

20、以下那一组不是哈夫曼编码 ( )

A. 0001 0000 100 11

B. 11 100 1010 1011

C. 001 0000 01 11

D. 110 101 11 111

21、给定以下权值  $w=\{5,29,7,8,14,23,3,11\}$ ，构建一个哈夫曼树，那么带权路径长度是多少？

A. 270 B. 271 C. 272 D. 273

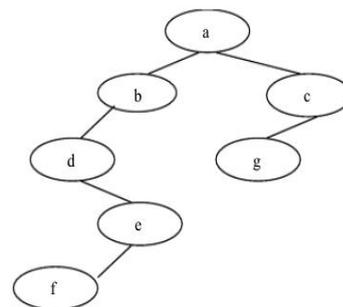
22、图示二叉树的中序遍历是 ( )。

A. a b c d g e f

B. d f e b a g c

C. d b a e f c g

D. d e f b a g c



23、将一棵有 100 个节点的完全二叉树从根这一层开始，

每一层从上往下，从右至左编号，根节点的编号 1，则编号最大的非叶节点的编号为 ( )。

A. 48 B. 49 C. 50 D. 51

## 图论

- 1、设无向图  $G$  中顶点个数为  $n$ ，则图  $G$  中至多有多少条边  
A.  $n$  B.  $n-1$  C.  $n(n-1)/2$  D.  $n(n-1)$
- 2、设有向图  $G$  中顶点个数为  $n$ ，则图  $G$  中至少有多少条边  
A. 0 B.  $n$  C.  $n(n-1)$  D.  $n(n-1)/2$
- 3、以下说法正确的是( )  
A. 任何连通图的连通分量只有一个，即是它本身  
B. 对于一个图而言，拓扑排序的结果唯一。  
C. 图的顶点集合是有穷可以为空的。  
D. 图的顶点集合可以为空，边集也可以为空。
- 4、无向完全图是图中每对顶点之间都恰有一条边的简单图。已知无向完全图  $G$  有 5 个顶点，则它共有 ( ) 条边。  
A. 9 B. 10 C. 20 D. 21
- 5、在一个无向图中，所有顶点的度数之和等于所有边数的( )倍  
A.  $1/2$  B. 1 C. 2 D. 4
- 6、 $n$  个顶点的强连通图至少有( )条边。  
A.  $n$  B.  $n+1$  C.  $n-1$  D.  $n*(n-1)$
- 7、 $n$  个顶点的强连通图,至少需要的边数连接的图形形状 ( )。  
A. 无回路 B. 有回路 C. 环状 D. 树状
- 8、对于一个具有  $n$  个顶点的无向图，如果采用邻接矩阵的存储形式，则该邻接矩阵的大小是( )。  
A.  $n$  B.  $(n-1)*2$  C.  $n-1$  D.  $2*n$

9、图的生成树 ( )。

- A. 唯一 B.不唯一 C.唯一性不能确定

10、G 是一个非连通无向图，共有 29 条边，则该图至少有 ( ) 个顶点

- A.6 B.7 C.8 D.9

11、最小生成树是指 ( )。

- A.由连通网所得到的边数最少的生成树。  
B.由连通网所得到的顶点数相对较少的生成树。  
C.连通网中所有生成树中权值之和最小的生成树  
D.连通网的极小连通子图。

12、设 G 是有 n 个结点、m 条边 ( $n \leq m$ ) 的连通图，需要多少条边才能使得 G 变成一棵树。

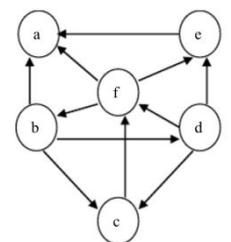
- A.n B.m C.n-1 D.n-m

13、邻接表是图的一种 ( )。

- A.顺序存储结构 B.链式存储结构 C.索引存储结构 D.散列存储结构

14、如图，顶点 f 的出度和入度分别为多少？

- A. 3,2  
B. 2,3  
C. 4,2  
D. 3,3



15、要连通具有 n 个顶点的有向图，至少需要 ( ) 条边。

- A.n-1 B.n C.n+1 D.2n

## 阅读代码理解专题

### 1. 第一题

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 int n,ans,cnt;
4 int a[1000][1000],rec[1000],dp[1000][1000];
5 int dfs(int i,int j){
6     if(dp[i][j]!=-1)
7         return dp[i][j];
8     if(i==n)
9         return dp[i][j]=a[i][j];
10    return dp[i][j] = max(dfs(i+1,j),dfs(i+1,j+1))+a[i][j];
11 }
12 int main(){
13     cin>>n;
14     for(int i=0;i<n;i++){
15         for(int j=0;j<=i;j++){
16             cin>>a[i][j];
17         }
18     }
19     memset(dp,-1,sizeof(dp));
20     cout << dfs(0,0);
21     return 0;
22 }
```

#### • 判断题

- 1) 输入 n 为 0，程序会出现运行错误。( )
- 2) 删除第 19 行的初始化代码，程序无论输入什么都会输出 0。( )
- 3) 如果除了 n 以外后续输入的数字都是 0，那么结果一定是 0。( )

#### • 单选

- 4) 输入 n 为 5，接下来需要输入多少个数字。( )  
A 10 个    B 15 个    C 20 个    D 25 个

5) 如果输出的结果是 10。以下输入可以得到这个结果的是 ( )

A 3 1 2 3 4 5 6

B 3 6 5 4 3 2 1

C 2 1 2 3

D 2 3 2 1

6) 输入 n 为 10，接下来输入的数字都是 1.输出的结果是 ( )

A 1      B 10      C 20      D 100

2.      第二题

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 int n,ans;
4 int a[1000],dp[1000];
5 int main(){
6     cin>>n;
7     for (int i=0; i<n; i++) {
8         cin>>a[i];
9     }
10    ans = dp[0] = a[0];
11    for (int i=1; i<n; i++) {
12        dp[i] = max(a[i],a[i]+dp[i-1]);
13    }
14    for (int i=0; i<n; i++) {
15        ans = max(ans,dp[i]);
16    }
17    cout << ans;
18    return 0;
19 }
```

• 判断题

- 1) 输入 n 为 0, 程序一定输出 0。( )
- 2) 删除第 10 行,程序会在运行时报错。( )
- 3) 除了 n 以外, 输入的所有数字都是 0, 结果也一定是 0。( )
- 4) 除了 n 以外, 输入的所有数字都是负数, 结果可能是 0。( )

• 单选

- 5) 输入 5 1 2 3 4 5 ,输出的结果是 ( )

A 5   B 10   C 15   D 25

6) 如果输出的结果是 10。以下输入无法得到这个结果的是 ( )

- A 10 1 2 3 4 5 6 7 8 9 10
- B 5 0 1 2 3 4
- C 10 1 -1 2 -2 3 -3 4 -4 5 5
- D 5 -1 1 4 5 -5

3.

```
1 #include <iostream>
2 using namespace std;
3 typedef long long LL;
4 LL array[21][21][21];
5 LL fun(LL x, LL y, LL z) {
6     if(x<=0 || y<=0 || z<=0) return 1;
7     else if(x>20 || y>20 || z>20) return fun(20,20,20);
8     else if(array[x][y][z] != 0) return array[x][y][z];
9     else if(x<y && y<z)
10        array[x][y][z]=fun(x,y,z-1)+fun(x,y-1,z-1)-fun(x,y-1,z);
11     else
12        array[x][y][z]=fun(x-1,y,z)+fun(x-1,y-1,z)+fun(x-1,y,z-1)-fun(x-1,y-1,z-1);
13     return array[x][y][z];
14 }
15 int main() {
16     LL x,y,z;
17     cin >> x >> y >> z;
18     cout << fun(x, y, z);
19     return 0;
20 }
```

判断题:

1) 将第 7 行的  $x>20 \parallel y>20 \parallel z>20$  改为  $x>20 \&\& y>20 \&\& z>20$ , 程序可能会运行错误。( )

A.正确 B.错误

2) 将第 7 行的  $x>20 \parallel y>20 \parallel z>20$  改为  $x>=20 \parallel y>=20 \parallel z>=20$ , 程序运行结果一定不变。( )

A.正确 B.错误

3) 若将第八行删除, 输入 2 2 2, 程序的运行结果一定不变。( )

A.正确 B.错误

单选题:

4) 若输入 1 2 3, 则输出为 ( )

A.1 B.2 C.3 D.4

5) 若输入 3 3 3, 则输出为 ( )

A.6 B.7 C.8 D.9

6) 当 x, y, z 相等且都等于 n, 则输出的结果为 ( )

A. $n^2$  B. $n^2-1$  C. $2^n$  D. $2^n-1$

4.

```
1 #include <iostream>
2 #include <cmath>
3 using namespace std;
4 int n, k, t, d[510], w[510];
5 int main() {
6     cin >> n >> k;
7     for(int i = 1; i <= n; i++){
8         d[i] = 0;
9         w[i] = 1;
10    }
11    for(int i = 2; i * i <= n; i++){
12        if(d[i] == 0){
13            t = i * i;
14            while(t <= n){
15                d[t] = i;
16                t += i;
17            }
18        }
19    }
20    for(int i = n; i >= 1; i--){
21        if(d[i]){
22            w[d[i]] += w[i];
23            w[i/d[i]] += w[i];
24            w[i] = 0;
25        }
26    }
27    cout << w[k];
28    return 0;
29 }
```

判断题:

4) 将第 8 行删除, 程序运行结果不变。( )

A.正确 B.错误

5) 将第 11 行的  $i*i \leq n$  改为  $i \leq \text{int}(\sqrt{n})+1$ , 程序运行结果改变。( )

A.正确 B.错误

6) 第 22 行的  $d[j]$ 一定是质数。( )

A.正确 B.错误

单选题:

7) 若输入的  $n$  为 5,  $k$  为 5, 则输出为 ( )

A.0 B.1 C.2 D.3

8) 若输入的  $n$  为 10,  $k$  为 3, 则输出为 ( )

A.1 B.2 C.3 D.4

9) 若输入的  $n$  为 20,  $k$  为 2, 则输出为 ( )

A.10 B.14 C.18 D.20

5、

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  int a[555555],n,k;
4  void qsort(int l,int r){
5      int m=a[(l+r)/2];
6      int i=l,j=r;
7      while(i<=j){
8          while(a[i]<m) i++;
9          while(a[j]>m) j--;
10         if(i<=j) swap(a[i++],a[j--]);
11     }
12     if(l<j&&k<=j) qsort(l,j);
13     if(i<r&&k>=i) qsort(i,r);
14 }
15 int main(){
16     cin>>n>>k;
17     for(int i=0;i<n;i++) scanf("%d",&a[i]);
18     qsort(0,n-1);
19     cout<<a[k]<<endl;
20     return 0;
21 }

```

1.程序运行之后,a 数组里面是一定从小到大(可以相等).

2.程序运行之后,a 数组里面是一定从大到小(可以相等).

3.第 10 行的条件可以改成 if(i<j),程序结果不会改变.

4.如果输入 4 2 7 5 3 9 则输出是();

A.3    B.7    C.5    D.9

5.如果输入 50 3 2 4 6 8 ...100 则输出是();

A.3    B.4    C.6    D.8

6.如果输入 50 1 100 98 96.....2 则输出是();

A.100    B.98    C.2    D.4

6、

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  long long dp[1000001],a,n,p=0;
4  int main(){
5      cin>>n;
6      for(long long i=1;i<=n;i++){
7          cin>>a;
8          p=max(p,a);
9          for(long long j=1;j*j<=a;j++){
10             if(a%j==0){
11                 dp[j]++;
12                 if(j!=a/j) dp[a/j]++;
13             }
14         }
15     }
16     for(long long i=1;i<=n;i++){
17         while(dp[p]<i) p--;
18         cout<<p<<endl;
19     }
20 }

```

已知所有输入都是  $10^6$  以内的正整数.

- 1.如果某次输入的  $a$  大于了 1000001,则程序可能会 RE;
- 2.程序第 3 行的  $p$  不需要赋初值(即把  $p=0$  改为  $p$ ).
- 3.已知此程序时间限制 1s,输入在题目允许范围内,程序可能会超时;
- 4.如果输入 4 1 2 3 4 则 16 行循环执行前, $dp[1] \sim dp[4]$ 的值分别是();
  - A.4 2 1 1
  - B.1 1 2 4
  - C.2 1 4 1
  - D.1 2 4 1
- 5.如果输入 4 18 81 54 63 则 16 行循环执行前, $p$  的值是();
  - A.4

B.18

C.81

D.63

6.如果输入 4 18 81 54 63 则输出是();

A.81 27 9 9

B.9 9 9 9

C.9 9 27 81

D.81 81 81 81

7、

```
1  #include<bits/stdc++.h>
2  using namespace std;
3  int n,a,dp[1100010]={9999999},ans;
4  int find(){
5      int l=1,r=ans,t;
6      while(l<=r){
7          int m=(l+r)/2;
8          if(dp[m]<a) t=m,r=m-1;
9          else l=m+1;
10     }
11     return t;
12 }
13 int main(){
14     cin>>n;
15     for(int i=1;i<=n;i++){
16         cin>>a;
17         if(a<=dp[ans]) dp[++ans]=a;
18         else dp[find()]=a;
19     }
20     cout<<ans;
21     return 0;
22 }
```

给定一长度为 n 的数列，请在不改变原数列顺序的前提下，从中随机的取出一定数量的整

数，并使这些整数构成单调不上升序列。输出这类单调不上升序列的最大长度。

1.程序第 3 行的 dp 赋初值是把整个 dp 数组都赋值为 9999999.

2.程序第 7 行的  $l+r$  的和必须是偶数,否则会导致程序结果错误.

3.程序的复杂度是  $O(n^2)$ ;

4.如果输入 5 2 2 3 2 2 则输出是();

A.2     B.3     C.4     D.5

5.如果输入 500 2 4 6...998 1000 则 dp[1]存的值是()

A.500     B.1000     C.2     D.1

6.如果输入 5 10 8 9 11 2 则 dp[1]~dp[5]存的值是();

A.11 9 2 0 0

B.10 8 2 0 0

C.10 9 2 0 0

D.1 3 5 0 0

6.如果输入 1000 1 2 3...999 1000 则

8、

```

1 #include <iostream>
2 #include <cstring>
3 using namespace std;
4 char str[107];
5 bool IsPrime(int num){
6     if(num==0||num==1) return false;
7     for (int i=2; i*i<=num; i++) {
8         if(num%i==0){
9             return false;
10        }
11    }
12    return true;
13 }
14 int main(){
15     cin>>str;
16     int n=(int)strlen(str);
17     for(int i=0;i<n;i++){
18         if(IsPrime(i)){
19             if(str[i]>='A'&&str[i]<='Z'){
20                 str[i]=str[i]+'a'-'A';
21                 continue;
22             }
23             if(str[i]>='a'&&str[i]<='z'){
24                 str[i]=str[i]-('a'-'A');
25             }
26         }
27     }
28     cout<<str;
29     return 0;
30 }

```

判断题:

- (1) 输入的字符串只能由大写或者小写字母组成。( ) A.正确 B.错误
- (2) 第 2 行的代码如果删除的话对于程序没有影响。( ) A.正确 B.错误
- (3) 第 21 行的代码删除的对于程序没有影响。( ) A.正确 B.错误
- (4) 如果输入大写字母的话，输出的也是大写字母。( ) A.正确 B.错误

选择题:

(5) 如果输入“AbCdEfG”那么输出结果应该是多少?

A. AbcDEFG B. aBcDeFg C. Abcdefg D. AbcDefg

(6) 如果输出的字符串是“abcd”，那么下列说法正确的是

A. 输入的字符串中为 ABCD

B. 输入的字符串中为 abCd

C. 输入的字符串中为 aBCD

D. 输入的字符串中为 abCD

9、

```
1 #include <iostream>
2 using namespace std;
3 const int mod=1024;
4 long long pow(int x,int y){
5     long long res=1;
6     while (y) {
7         if(y&1){
8             res=(res*x)%mod;
9         }
10        x=(x*x)%mod;
11        y>>=1;
12    }
13    return res;
14 }
15 int main(){
16     int x,y;
17     cin>>x>>y;
18     long long ans=((pow(x-1,y)+x*pow(-1,y))+mod)%mod;
19     cout<<ans<<endl;
20 }
```

判断题:

(1) 将程序中的 "y>=1" 改为 "y\*=0.5",程序输出结果一定不变 ( ) A.正确 B.错误

(2) 将程序中的 "y&1" 改为 "y%2==1",程序输出结果一定不变 ( ) A.正确 B.错误

(3) 此程序的时间复杂度为  $O(\log N)$  ( ) A.正确 B.错误

(4) 如果输入"4 4",则输出为 "75"。( ) A.正确 B.错误

选择题:

(5) 如果输入 "3 4" 那么输出结果应该是多少?

A. 24 B. 12 C.18 D.19

(6) 执行  $\text{pow}(1028,13)$ 返回的结果是什么

A.1024

B.1022

C.0

D.2

## 完形填空专题

1 (高精度除法) 输入a和b两个字符串表示两个除数与被除数(两者都是正整数),循环执行a-b的操作,没运行依次ans增大1,直到a<b为止。最后输出a乘以b的结果和余数。由于ab都是字符串,在执行a-b操作是,先将两者转换为数组,使得数组从左到右,数位依次增大,(比如下标0的位置记录个位数,下标1记录十位数)。提示:完成依次a-b的操作后,需要更新a字符串。

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  string a,b;
4  int c[10];
5  int d[10];
6  int ans;
7  bool cmp(string str1,string str2){
8      if(str1.size()<str2.size()||(str1.size()==str2.size()&& __①__ )){
9          return true;
10     }
11     return false;
12 }
13 bool subtraction(){
14     if(__②__){
15         return false;
16     }else{
17         int ci=0,di=0;
18         for (int i=a.size()-1; i>=0; i--) {
19             c[ci++] = a[i]-'0';
20         }
21         for (int i=b.size()-1; i>=0; i--) {
22             d[di++] = b[i]-'0';
23         }
24         for (int i=0;i<di;i++) {
25             __③__;
26         }
27         for(int i=0;i<ci;i++){
28             if(c[i]<0){
29                 c[i]+=10;
30                 __④__;
31             }
32         }
33         while(c[ci]==0&&ci>0){
34             ci--;
35         }
36         string tmp="";
37         while(ci>=0){
38             tmp+=(char)(__⑤__);
39         }
40         a=tmp;
41         return true;
42     }
43 }
44 int main() {
45     cin>>a>>b;
46     cout<<a<<"/"<<b<< "=";
47     if(cmp(a,b)){
48         cout<< 0 << "... " << a <<endl;
49     }else{
50         while(subtraction()){
51             ans++;
52         }
53         cout<< ans << "... " << a <<endl;
54     }
55     return 0;
56 }

```

1) ①处应填 ( )

- A `str1<=str2`
- B `str1< str2`
- C `str1 >= str2`
- D `str1>str2`

2) ②处应填 ( )

- A `cmp(a,b)`
- B `cmp(b,a)`
- C `cmp(c,d)`
- D `cmp(d,c)`

3) ③处应填 ( )

- A `d[i] -= (c[i] + '0' )`
- B `c[i] -= (d[i] + '0' )`
- C `c[i] -= d[i]`
- D `d[i]-=c[i]`

4) ④处应填 ( )

- A `c[i+1] --`
- B `c[i+1] ++`
- C `c[i-1] --`
- D `c[i-1] ++`

5) ⑤处应填 ( )

- A `c[ci]`

- B c[ci--]
- C c[ci]+'0'
- D c[ci--]+'0'

2、阶乘的定义如下： $n! = 1 * 2 * 3 * \dots * (n-1) * n$ 。

例如： $5! = 120$ ，从右向左的第一个非0数字为2，后面有1个0

输入：5

输出：2 1

```

1  #include <iostream>
2  using namespace std;
3  int n, w[1000005];
4  int main() {
5      cin >> n;
6      int cnt1 = __1__;
7      for(int i = 1; i <= n; i++){
8          w[i] = i;
9          while(w[i] % 5 == 0){
10             cnt1++;
11             w[i] /= 5;
12         }
13     }
14     int cnt2 = __2__;
15     for(int i = 1; i <= n; i++){
16         while(cnt2 && __3__){
17             __4__;
18             w[i] /= 2;
19         }
20         if(cnt2 == 0) break;
21     }
22     int s = 1;
23     for(int i = 1; i <= n; i++){
24         s = __5__;
25     }
26     cout << s << " " << cnt1 << endl;
27     return 0;
28 }

```

1) 1 处应该填 ( )

- A.0 B.1 C.2 D.n

2) 2 处应该填 ( )

A.0 B. 1 C.cnt1 D.n

3) 3 处应该填 ( )

A.cnt1 B. w[i] C.w[i]%2!=0 D.w[i]%2==0

4) 4 处应该填 ( )

A.cnt1++ B. cnt1-- C.cnt2++ D.cnt2--

5) 5 处应该填 ( )

A. w[i]%10 B. (s+w[i])%10

C.(s\*w[i])%10 D.(s\*10+w[i])%10

3、dfs:把 m 个同样的苹果放在 n 个同样的盘子里, 允许有的盘子空着不放, 问共有多少种不同的分法。(例如输入 7 3,输出 8. 5,1,1 和 1,1,5 是同一种方法)

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 int m,n,ans;
4 void dfs(int m,int n,int c){
5     if(__1__){
6         ans++;
7         return;
8     }
9     for(int i=c;i<=__2__;i++)
10        dfs(__3__,n-1,__4__);
11 }
12 int main(){
13     cin>>m>>n;
14     dfs(m,n,__5__);
15     cout<<ans<<endl;
16     return 0;
17 }
```

1.①处应填 ( )

A.n==1    B.n==0    C.m==0    D.m==0&&n==0

2.②处应填 ( )

A.m/n    B.n/m    C.m    D.n

3.③处应填 ( )

A.m-i    B.i    C.m    D.m-1

4.④处应填 ( )

A.i    B.i+1    C.i-1    D.c+1

5.⑤处应填 ( )

A.0    B.1    C.-1    D.m

4、求最小质因数;

输入一个 n,输入 n 个数,求每个数的最小质因数.

例:

输入 5    31    32    33    34    35

输出 31    2    3    2    5

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  const int N=300000;
4  bool isprime[N];
5  int prime[N],ans[N],n,a;
6  void euler(){
7      int sum=0;
8      memset(isprime+2,1,sizeof(isprime)-2);
9      for(int i=2;i<=N;i++){
10         if(isprime[i]) prime[++sum]=i,ans[i]=__1__;
11         for(int j=1;j<=sum;j++){
12             if(__2__) break;
13             isprime[i*prime[j]]=0;
14             ans[__3__]=__4__;
15             if(__5__) break;
16         }
17     }
18 }
19 int main(){
20     euler();
21     cin>>n;
22     for(int i=1;i<=n;i++){
23         cin>>a;
24         cout<<ans[a]<<endl;
25     }
26 }

```

1.①处应填 ( )

A.0    B.1    C.sum    D.i

2.②处应填 ( )

A.i>N    B.j>N    C.i\*j>N    D.i\*prime[j]>N

3.③处应填 ( )

A.i    B.j    C.prime[j]    D.i\*prime[j]

4.④处应填 ( )

A.i    B.j    C.sum    D.prime[j]

5.⑤处应填 ( )

A.prime[j]%i!=0    B.i%prime[j]!=0    C.prime[j]%i==0    D.i%prime[j]==0

5、一场瘟疫来袭。瘟疫从四周向 Q 城市传染，导致该城市的很多地区被瘟疫感染，幸好该城市有些重要的地方四周盖好了隔离病毒墙，高墙内的地区瘟疫无法到达。给定 Q 城市的地图情况，计算城市中没有被瘟疫感染的地区面积。

**【输入】**

第一行：n, m, 表示村子的大小。

以下是 n 行 m 列的矩阵，描述村子的情况，“0”表示可以感染的地方，“1”表示墙，“0”和“1”之间无空格。

**【输出】**

城市没被洪水淹没的平地面积。

根据以下提示补全代码。

```
1 #include <iostream>
2 using namespace std;
3 int mp[507][507];
4 bool vis[507][507];
5 int n,m,cnt=0;
6 char c;
7 void dfs(int x,int y){
8     if( ④ ) return;
9     mp[x][y]=1;
10    if(x>n||y>m||x<1||y<1) return;
11    if(vis[x][y]==1) return;
12    dfs(x+1,y);
13    dfs(x,y+1);
14    ⑤
15 }
16 int main(){
```

```

19 int main(){
20     cin>>n>>m;
21     for (int i=1;i<=n;i++ ) {
22         for (int j=1; j<=m;j++ ) {
23             cin>>c;
24             ①;
25         }
26     }
27     for(int i=1;i<=n;i++) dfs(i, 1);
28     for(int i=1;i<=n;i++) ②;
29     for(int i=1;i<=m;i++) dfs(1, i);
30     for(int i=1;i<=m;i++) ③;
31     for (int i=1;i<=n;i++ ) {
32         for (int j=1; j<=m;j++ ) {
33             if(mp[i][j]==0) cnt++;
34         }
35     }
36     cout<<cnt;
37     return 0;
38 }

```

(1) ①应该填写

A.mp[i][j]=c B.mp[i][j]=1 C.mp[i][j]=0 D. mp[i][j]=c-'0';

(2) ②应该填写

A.dfs(i, m); B.dfs(i, n); C.dfs(1, i); D.dfs(m, i);

(3) ③应该填写

A.dfs(m, i) B.dfs(n, i); C.dfs(i, 1) D.dfs(n, n)

(4) ④应该填写

A.mp[i][j]==0 B.vis[i][j]=1 C.mp[i][j]=1 D.vis[i][j]=0

(5) ⑤应该填写

A.  $\text{dfs}(x-1, y); \text{dfs}(x+1, y);$

B.  $\text{dfs}(x+1, y); \text{dfs}(x, y-1);$

C.  $\text{dfs}(x-1, y); \text{dfs}(x, y+1);$

D.  $\text{dfs}(x-1, y); \text{dfs}(x, y-1);$