

## 2023 年 CSP-J 初赛题解

一. 单项选择题(共 15 题, 每题 2 分, 共计 30 分:每题有且仅有一个正确选项)

1.在 C++中, 下面哪个关键字用于声明一个变量, 其值不能被修改?

A. unsigned

B. const

C. Static

D. mutable

正确答案: B

解析:const 常量, 不能被修改

2.八进制数(12345670)<sub>8</sub> 和(07654321)<sub>8</sub> 的和为()。

A.(22222221)<sub>8</sub>

B.(21111111)<sub>8</sub>

C.(22111111)<sub>8</sub>

D.(22222211)<sub>8</sub>

正确答案: D

解析:列竖式, 对应位数满 8 就进一位, 计算结果为%8 的余数。

1	2	3	4	5	6	7	0
0	7	6	5	4	3	2	1
2	2	2	2	2	2	1	1

3.阅读下述代码，请问修改 data 的 value 成员以存储 3.14，正确的方式是()。

```
1 union Data{
2     int num;
3     float value;
4     char symbol;
5 };
6 union Data data;
```

- A. data.value = 3.14;      B. value.data = 3.14;
- C. data->value = 3.14;    D. value->data = 3.14;

正确答案：A

解析:变量类型和变量名

4. 假设有一个链表的节点定义如下：

```
1 struct Node {
2     int data;
3     Node* next;
4 };
```

现在有一个指向链表头部的指针：Node\* head。如果想要在链表中插入一个新节点，其成员 data 的值为 42，并使新节点成为链表的第一个节点，下面哪个操作是正确的？（ ）

A. Node\* newNode = new Node;  
newNode->data = 42; newNode->next = head;  
head = newNode;

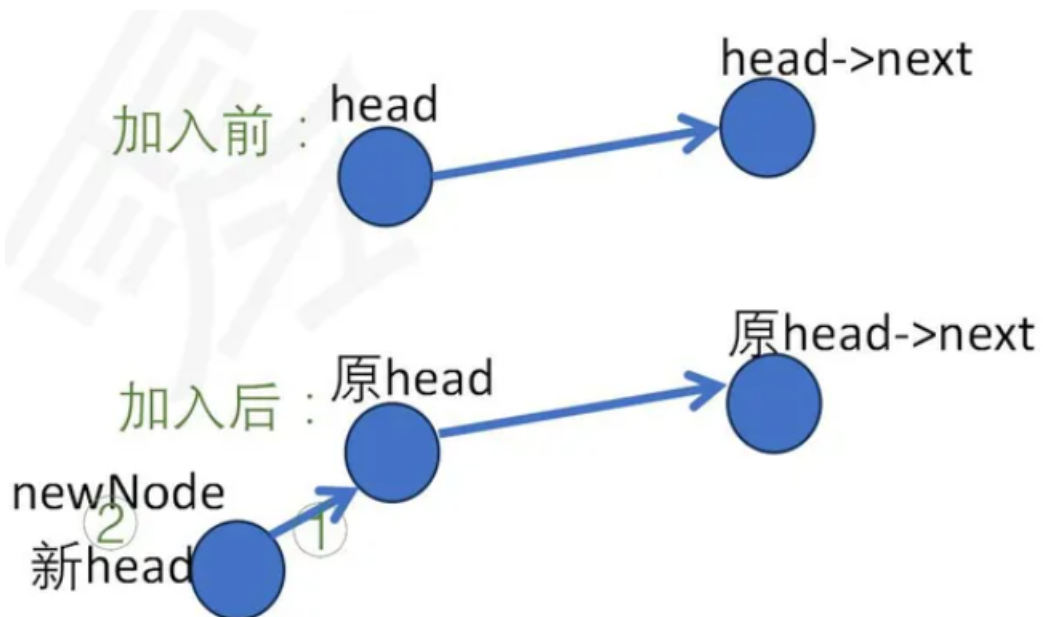
B. Node\* newNode = new Node; head->data = 42; newNode->next = head; head = newNode;

C. Node\* newNode = new Node; newNode->data = 42; head->next = newNode;

D. Node\* newNode = new Node; newNode->data = 42; newNode->next = head;

正确答案：A

解析:链表。注意①和 2 的顺序不能颠倒，



5.根节点的高度为 1，一根拥有 2023 个节点的三叉树高度至少为()

A.6      B. 7      C.8      D.9

正确答案：C

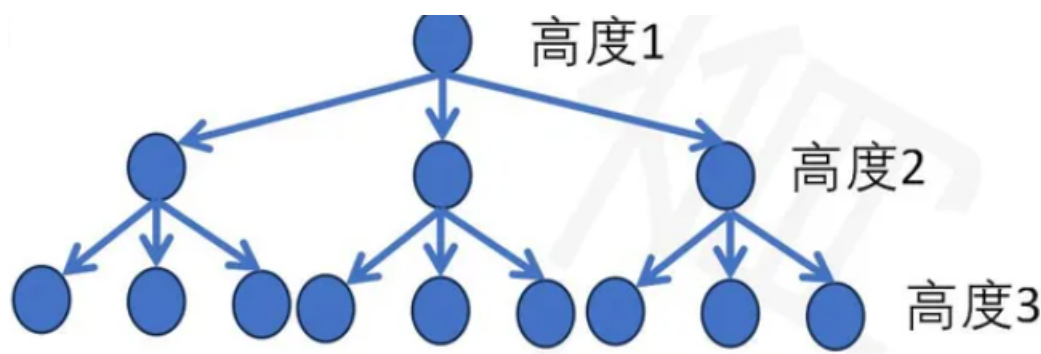
解析:当层数为  $h$  层时，总的结点数为  $1+3+3^2+\dots+3^{h-1}$ ，根据等比数列求和公式  $1+3+3^2+\dots+3^{h-1}$ ，这个式子的答案相当于  $(3^h-1)/(3-1)=(3^h-1)/2$ 。

当  $h=7$  时， $3^h=(2187-1)/2=1093$ 。

当  $h=8$  时， $3^h=6561/2=3280$

2023 介于这两个数值之间，因此高度应该为 8。

当然，如果每层不排满，那么层数会更多。所以至少为 8。



6. 小明在某一天中依次有七个空闲时间段，他想要选出至少一个空闲时间段来练习唱歌，但他希望任意两个练习的时间段之间都有至少两个空闲的时间段让他休息，则小明一共有（ ）种选择时间段的方案。

A. 31      B. 18      C. 21      D. 33

正确答案：B

解析:可以采用枚举法，1 代表唱歌，0 代表休息。那么可以有如下方案。

1001001	0100100	0010010	0001001
1001000	0100010	0010001	0001000
1000100	0100001	0010000	
1000010	0100000		
1000001			
1000000			
0000100	0000010	0000001	

一共 18 种排列方式

7. 以下关于高精度运算的说法错误的是（ ）。

A. 高精度计算主要是用来处理大整数或需要保留多位小数的运算。

B. 大整数除以小整数的处理的步骤可以是，将被除数和除数对齐，从左到右逐位尝试将除数乘以某个数，通过减法得到新的被除数，并累加商。

C. 高精度乘法的运算时间只与参与运算的两个整数中长度较长者的位数有关。

D. 高精度加法运算的关键在于逐位相加并处理进位。

正确答案：C

解析:高精度乘法的时间复杂度是:两个乘数的位数之积。

具体做法是让第一个数的第  $i$  位乘以 第二个数的第  $j$  位，可以贡献到结果的第  $i+j-1$  位。 $i$  和  $j$  要进行嵌套枚举。

8. 后缀表达式“6 2 3 + - 3 8 2 / + \* 2 ^ 3 +”对应的中缀表达式是 ( )

A.  $((6 - (2 + 3)) * (3 + 8 / 2)) ^ 2 + 3$

B.  $6 - 2 + 3 * 3 + 8 / 2 ^ 2 + 3$

C.  $(6 - (2 + 3)) * ((3 + 8 / 2) ^ 2) + 3$

D.  $6 - ((2 + 3) * (3 + 8 / 2)) ^ 2 + 3$

正确答案：A

解析:对于后缀表达式，从左到右扫描，遇到数字进栈，遇到符号，就把栈顶的两个元出栈，进行符号相应的运算，再把运算结果入栈

9. 数 $(101010)_2$  和 $(166)_8$  的和为 ( )。

A.  $(10110000)_2$

B.  $(236)_8$

C.  $(158)_{10}$

D.  $(A0)_{16}$

正确答案：D

解析: $166(8)=1110110(2)101010(2)+1110110(2)$

$=10100000(2)=A0(16)$

10. 假设有一组字符{a,b,c,d,e,f} ,对应的频率分别为 5% , 9% , 12% , 13% , 16% , 45%。请问以下哪个选项是字符 a,b,c,d,e,f 分别对应的一组哈夫曼编码？ ( )

A. 1111 , 1110 , 101 , 100 , 110 , 0

B. 1010 , 1001 , 1000 , 011 , 010 , 00

C. 000 , 001 , 010 , 011 , 10 , 11

D. 1010 , 1011 , 110 , 111 , 00 , 01

正确答案：A

解析:哈夫曼编码的原理是:让频率小的字母编码长度长,频率大的字母编码长度短因此达到文章总体编码长度更短的效果。具体过程如下(非题目样例):



11.给定一棵二叉树,其前序遍历结果为:ABDECFG,中序遍历结果为:DEBACFG。请问这棵树的正确后序遍历结果是什么?()

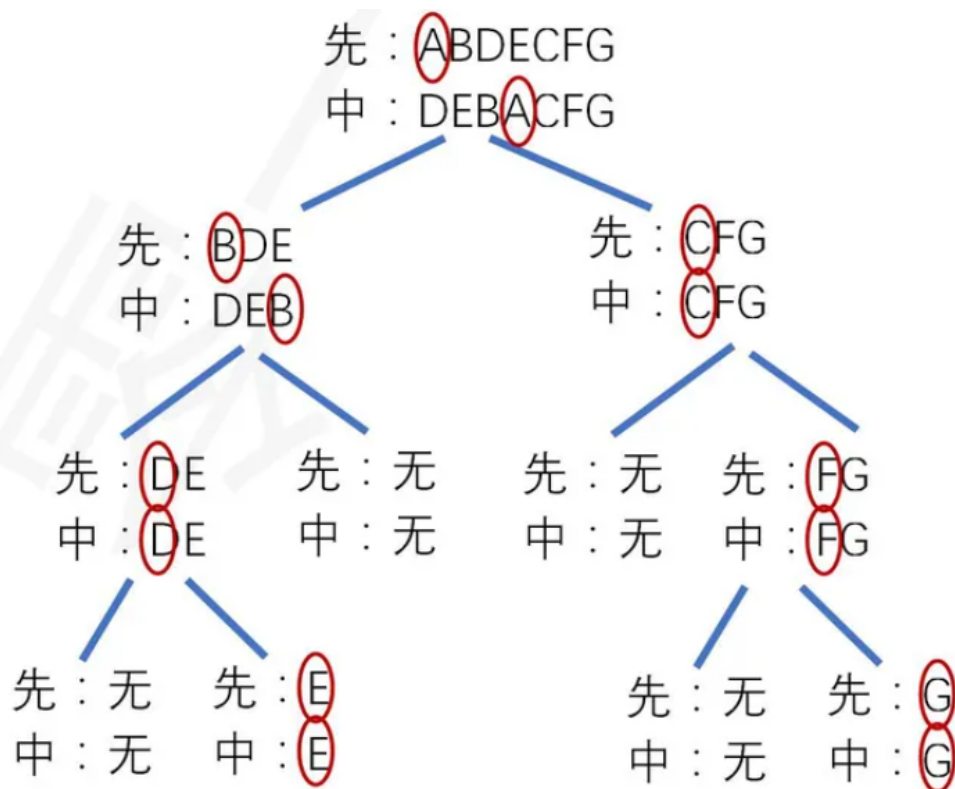
- A. EDBFGCA      B. EDBGCF A
- C. DEBGFCA      D. DBEGFCA



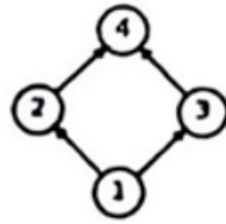
正确答案：A

解析:原则:对于每一个结点都满足先序:根左右中序:左根右后序:左右根无论哪种,都是先左后右。先序的第1个一定是根。先序和中序里,左侧结点的数量相等右侧结点的数量也相等可以通过中序里根的位置,判断左右子树的大小。

下面的红圈标出每一个结点体系的根。



12.考虑一个有向无环图,该图包括4条有向边:(1,2), (1,3), (2,4),和(3,4)。以下哪个选项是这个有向无环图的一个有效的拓扑排序?()



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

正确答案：B

解析:图上能顺利递推的方向 1234 或者 1324 都可以

13.在计算机中，以下哪个选项描述的数据存储容量最小?()

- A.字节(byte)      B.比特(bit)  
C.字(word)        D.千字节(kilobyte)

正确答案：B

解析:1 个比特只占 1 位。

14.一个班级有 10 个男生和 12 个女生。如果要选出一个 3 人的小组，并且小组中必须至少包含 1 个女生，那么有多少种可能的组合?()

- A. 1420    B. 1770    C.1540    D.2200

正确答案:A

解析:解决反问题更简单小组里一个女生都没有的方案

数: $C^3_{10}$  不考虑男女，总共的方案数: $C^3_{22}$ 。

至少 1 个女生的方案数: $C^3_{22} - C^3_{10} = 1420$

15.以下哪个不是操作系统?

A. Linux

B. Windows

C. Android

D.HTML

正确答案：D

解析:HTML:超文本标记语言，用于网页

阅读程序(程序输入不超过数组成字符串定义的范围:

判断题正确填 V，错误填 x;除特殊说明外，判断

题 1.5 分，选择题 3 分，共计 40 分)

```
01 #include<iostream>
02 #include<cmath>
03 using namespace std;
04
05 double f(double a,double b,double c){
06     double s=(a+b+c)/2;
07     return sqrt(s*(s-a)*(s-b)*(s-c));
08 }
09
10 int main(){
11     cout.flags(ios::fixed);
12     cout.precision(4);
13
14     int a,b,c;
15     cin>>a>>b>>c;
16     cout<<f(a,b,c)<<endl;
17     return 0;
18 }
```

假设输入的所有数都为不超过 1000 的正整数，完成下面的  
判断题和单选题

### 判断题

16.(2 分)当输入为“22 2”时，输出为“1.7321”()

17.(2 分)将第 7 行中的“(s-b)\*(s-c)”改为“(s-c)\*(s-b)”不会影响  
程序运行的结果()

18.(2 分)程序总是输出四位小数()

### 单选题

19.当输入为“3 4 5”时，输出为()

A."6.0000" B."12.0000"C."24.0000"D."30.0000"

20.当输入为“5 12 13”时，输出为()

A."24.0000"B."30.0000"C."60.0000" D."120.0000"

题目解析:本题是求三角形面积的海伦公式，即给出三角形的  
三边长  $a$ ， $b$ ， $c$ ，利用 07 句给出的公式，即可方便地求出三  
角形的面积。

### 1. 判断题解析

16.(2 分)当输入为“22 2”时，输出为“1.7321”()

正确答案：正确

解析:数据不大，直接代入数据计算即可， $4 * \sqrt{3/4} = \sqrt{3}$

17.(2 分)将第 7 行中的" $(s-b)*(s-c)$ "改为" $(s-c)*(s-b)$ "不会影响程序运行的结果()

正确答案：正确

解析:乘法交换律是两个变量交换位置，乘积不变

18.(2 分)程序总是输出四位小数()

正确答案：错误

解析:第 12 句指明了保留 4 位小数。但是题目并没有保证输入的  $abc$  一定能构成三角形那么这时根号下可能为负数，就不能正常运行

## 2. 选择题解析

19.当输入为“3 4 5”时，输出为()

A."6.0000" B."12.0000"C."24.0000"D."30.0000'

正确答案：A

解析:3,4,5 构成等边三角形，所以面积是  $3*4/2 = 6$

20.当输入为"5 12 13"时，输出为()

A."24.0000"B."30.0000"C."60.0000" D."120.0000

正确答案：B

解析:5,12,13 构成等边三角形，所以面积是  $5*12/2 = 30$

(2)

```
01 #include<iostream>
02 #include<vector>
03 #include<algorithm>
04 using namespace std;
05
06 int f(string x,string y){
07     int m=x.size();
08     int n=y.size();
09     vector<vector<int>>v(m+1,vector<int>(n+1,0));
10     for(int i=1;i<=m;i++){
11         for(int j=1;j<=n;j++){
12             if(x[i-1]==y[j-1]){
13                 v[i][j]=v[i-1][j-1]+1;
14             }else{
15                 v[i][j]=max(v[i-1][j],v[i][j-1]);
16             }
17         }
18     }
19     return v[m][n];
20 }
21
22 bool g(string x,string y){
23     if(x.size() != y.size()){
24         return false;
25     }
26     return f(x+x,y)==y.size();
27 }
28
29 int main(){
30     string x,y;
31     cin>>x>>y;
32     cout<<g(x,y)<<endl;
33     return 0;
34 }
```

## 判断题

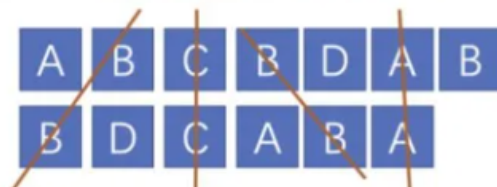
21. f 函数的返回值小于等于  $\min(n, m)$ 。( )
22. f 函数的返回值等于两个输入字符串的最长公共子串的长度。( )
23. 当输入两个完全相同的字符串时, g 函数的返回值总是 true ( )

## 单选题

24. 将第 19 行中的“v[m][n]”替换为“v[n][m]”, 那么该程序 ( )
- A. 行为不变      B. 只会改变输出
- C.一定非正常退出   D. 可能非正常退出
25. 当输入为 "csp-j p-jcs" 时,输出为( )
- A. "0"      B. "1"      C "T"      D. "F"
- 26 当输入为“csppsc spsccp”时, 输出为: ( )
- A. "T"      B. "F"      C. "0"      D. "1"

解析：题目解析本题的 f 函数为【最长公共子序列】问题的模板程序。即可以求出两个字符串共同含有的子串里，最长的子串的长度。注意子串不一定取连续的字母，但是顺序必须和原始顺序保持一致。

例如，下面两个字符串的最长公共子串“BCBA”已经用线对应地标出。



### 状态转移的部分

$\text{if}(x[i-1] == y[j-1])$  对应了图 1 的情况。

这种情况，匹配肯定相比  $v[i-1][j-1]$  增多 1 个

$\text{if}(x[i-1] \neq y[j-1])$  对应了图 2 的情况。

其中  $v[i-1][j]$  对应图 2 的绿线， $v[i][j-1]$  对应图 2 的蓝线。即如果末尾字母不等，那么答案就是这 2 种情况取优。

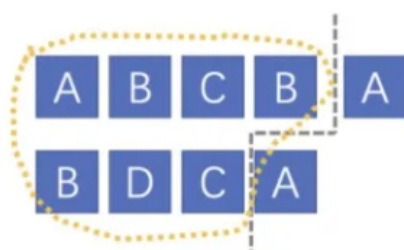


图1



图2

第 9 句: 等同定义一个  $m+1$  行乘以  $n+1$  列的二维数组

第 26 句:  $x+x$  是字符串的连接，相当于对于 2 个粘在一起的  $x$  串和 1 个  $y$  串找最长公共子序列。第 26 句要求这两个串的最长公共子序列的长度必须为  $y$  的长度，就相当于，最长公共子序



列就是  $y!$ 也就相当于在 2 个粘在一起的  $x$  里，看能否凑出一个  $y$

判断题

21.  $f$  函数的返回值小于等于  $\min(n, m)$ 。 ( )

正确答案：正确

解析:【最长公共子序列】的长度不可能超过  $n$  和  $m$  中较短的那个。

22.  $f$  函数的返回值等于两个输入字符串的最长公共子串的长度。 ( )

正确答案：错误

解析:根据以上分析，本题是求 2 个粘在一起的  $x$  串和 1 个  $y$  串的最长公共子序列。

23. 当输入两个完全相同的字符串时,  $g$  函数的返回值总是  $true$  ( )

正确答案：正确

解析:输入 2 个完全相同的字符串，输入  $abc$  和  $abc$ ，那么本题会求出  $abcabc$  和  $abc$  的最长公共子序列，就是  $abc$  自己。

24. 将第 19 行中的“v[m][n]”替换为“v[n][m]”，那么该程序（ ）

- A. 行为不变      B. 只会改变输出  
C. 一定非正常退出   D. 可能非正常退出

正确答案：D

解析:如果数组只有 10\*20，那么当访问 20\*10 的位置时，就会出现 RE 错误，即可能非正常退出。

25. 当输入为 “csp-j p-jcs” 时,输出为( )

- A. “0”      B. “1”      C “T”      D. “F”

正确答案：B

解析:找出 csp-jcsp-j 和 p-jcs 的最长公共子序列，看结果是不是和第二个串相等的。可见是可以成功找到的

26 当输入为“csppsc spsccp”时，输出为：（ ）

- A. “T”      B. “F”      C. “0”      D. “1”

正确答案：D

解析:找出 csppscsppcs 和 spsccp 的最长公共子序列，看结果是不是和第二个串相等的。可见是可以成功找到的。

```
01 #include <iostream>
02 #include <cmath>
03 using namespace std;
04
05 int solve1(int n){
06     return n*n;
07 }

09 int solve2(int n){
10     int sum=0;
11     for(int i=1;i<=sqrt(n);i++){
12         if(n%i==0){
13             if(n/i==i){
14                 sum+=i*i;
15             }else{
16                 sum+=i*i+(n/i)*(n/i);
17             }
18         }
19     }
20     return sum;
21 }
```

```
23 int main(){
24     int n;
25     cin>>n;
26     cout<<solve2(solve1(n))<<"
"<<solve1((solve2(n)))<<endl;
27     return 0;
28 }
```

假设输入的  $n$  是绝对值不超过 1000 的整数，完成下面的判断题和单选题。

### 判断题

27. 如果输入的  $n$  为正整数, solve2 函数的作用是计算  $n$  所有的因子的平方和 ( )
28. 第 13~14 行的作用是避免  $n$  的平方根因子  $i$  ( 或  $n/i$  ) 进入第 16 行而被计算两次 ( )
29. 如果输入的  $n$  为质数, solve2 (  $n$  ) 的返回值为  $n^2+1$  ( )

## 单选题

30. ( 4 分)如果输入的  $n$  为质数  $p$  的平方, 那么 `solve2` ( $n$ ) 的返回值为 ( )

A.  $p^2+p+1$  B.  $n^2+n+1$  C.  $n^2+1$  D.  $p^4+2p^2+1$

31. 当输入为正整数时, 第一项减去第二项的差值一定 ( )

A. 大于 0 B. 大于等于 0 且不一定大于 0

C. 小于 0 D. 小于等于 0 且不一定小于 0

32. 当输入为"5"时, 输出为 ( )

A. "651.625" B. "650.729" C. "651.676" D. "652.625"

## 题目解析

`solve2` 函数是在快速枚举  $n$  的所有因数, 同时求出了因数的平方和。例如对于  $n=6$ , 那么 `solve2` 将返回  $1^2+2^2+3^2+6^2$

26 句

```
cout<<solve2(solve1(n))<<" "<<solve1((solve2(n)))<<endl;里,
```

前者 `solve2(solve1(n))` 是:  $n^2$  的因数的平方和

后者 `solve1((solve2(n)))` 是:  $(n \text{ 的因数的平方和})^2$ 。

举个例子：

$$n=6$$

$$\text{前者} = 1^2 + 2^2 + 3^2 + 4^2 + 6^2 + 9^2 + 12^2 + 18^2 + 36^2$$

$$\begin{aligned}\text{后者} &= (1^2 + 2^2 + 3^2 + 6^2)^2 \\ &= (1^2 + 2^2 + 3^2 + 6^2) * (1^2 + 2^2 + 3^2 + 6^2)\end{aligned}$$

## 1. 判断题解析

27. 如果输入的  $n$  为正整数，solve2 函数的作用是计算  $n$  所有的因子的平方和（ ）

正确答案：正确

解析:根据以上分析，是正确的。

28. 第 13~14 行的作用是避免  $n$  的平方根因子  $i$ (或  $n/i$ ) 进入第 16 行而被计算两次（ ）

正确答案：正确

解析:求  $n$  的因数的快速方法，经典优化

29. 如果输入的  $n$  为质数，solve2 ( $n$ ) 的返回值为  $n^2 + 1$ （ ）

正确答案：正确

解析:质数的因数只有 1 和自己。

## 单选题解析

30. (4 分) 如果输入的  $n$  为质数  $p$  的平方, 那么  $\text{solve2}(n)$  的返回值为 ( )

A.  $p^2+p+1$  B.  $n^2+n+1$  C.  $n^2+1$  D.  $p^4+2p^2+1$

正确答案: B

解析:

$$n=p^2$$

$\text{solve2}(n)$  为  $p^2$  的所有因数的平方和  $p^2$  的因数为 1,  $p$ , 那么答案是  $1+p^2+p^4$ , 或者是  $1+n+n^2$

31. 当输入为正整数时, 第一项减去第二项的差值一定 ( )

A. 大于 0      B. 大于等于 0 且不一定大于 0  
C. 小于 0      D. 小于等于 0 且不一定小于 0

正确答案: D

解析后者一般情况下是  $>$  前者的。当然极端状况下, 也可以相等, 比如  $n=1$  时因此前者-后者  $\leq 0$ 。

32. 当输入为“5”时，输出为（ ）

A. "651.625" B. "650.729" C. "651.676" D. "652.625"

正确答案：C

解析:代入计算即可  $n=5$  前者= $1^2+5^2+25^2$  后者= $(1^2+5^2)^2$

### 三、完善程序(单选题，每小题 3 分，共计 3 分)

(1)(寻找被移除的元素)问题:原有长度为  $n+1$  公差为 1 等升数列，将数列输到程序的数组时移除了一个元素，导致长度为  $n$  的升序数组可能不再连续，除非被移除的是第一个或最后之个元素。需要在数组不连续时，找出被移除的元素。试补全程序。

```
1 01 #include <iostream>
2 02 #include <vector>
3 03
4 04 using namespace std;
5 05
6 06 int find missing(vector<int>& nums) (
7 07 int left = 0, right = nums.size() - 1;
8 08 while (left < right){
9 09     int mid = left + (right - left) / 2;
10 10     if (nums[mid] - mid+ ①) (
11 11         ②;
12 12     }else{
13 13         ③
14 14     }
15 15 }
16 16 return ④;
17 17 }
```



```

18 18
19 19 int main() (
20 20     int n;
21 21 cin >> n;
22 22 vector<int> nums(n);
23 23 for (int i= 0; i< n; i++) cin >> nums[i];
24 24 int missing_number = find_missing(nums);
25 25 if_(missing_number == ③) {
26 26     cout << "Sequence is consecutive" << endl;
27 27 }else{
28 28     cout << "Missing number is " << ,missing number << endl;
29 29 }
30 30 return 0;
31 31 }

```

33. ①处应填 ( )

A. 1 B. nums[0] C. right D. left

34. ②处应填 ( )

A. left=mid+1 B. right=mid-1

C. right=mid D. left=mid

35. ③处应填 ( )

A. left=mid+1 B. right=mid-1

C. right=mid D. left=mid

36. ④处应填 ( )

A.left+nums[0] B.right+nums[0]

C.mid+nums[0] D.right+1

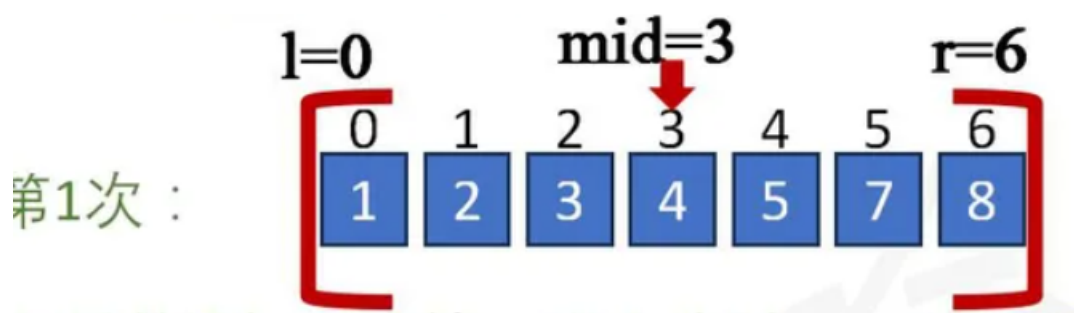
37. ⑤处应填 ( )

A.nums[0]+n B.nums[0]+n-1

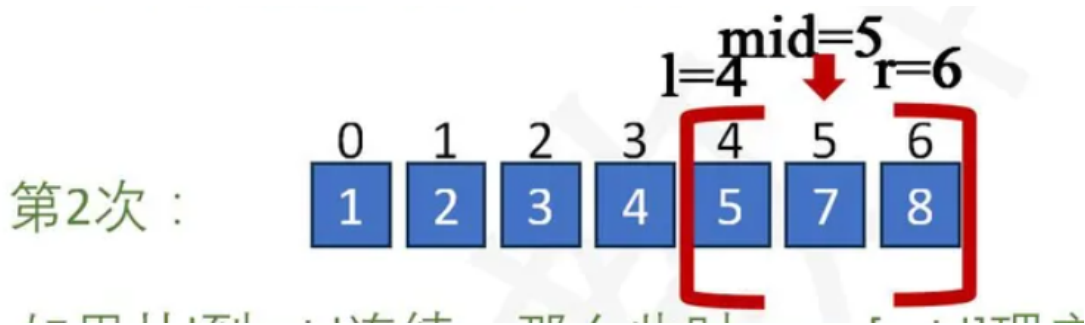
C.nums[0]+n+1 D.nums[n-1]

题

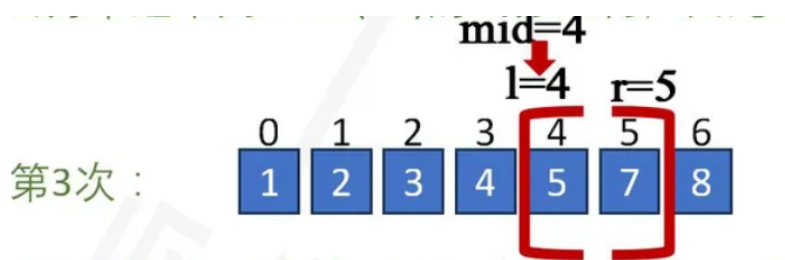
目解析:本题是用二分的方式,找到缺失的数据。让我们举个例子:



如果从  $l$  到  $mid$  连续,那么此时  $nums[mid]$  理应就=4 说明残缺的数值一定不在左边,而且一定不是  $mid$ ,因此应该排除掉  $mid$  向右找,即  $l=mid+1$ 。



如果从 1 到 mid 连续，那么此时  $\text{nums}[\text{mid}]$  理应=6 然而却等于 7，说明残缺的数值就出现在 l 到 mid 之间，因此应该向左找，即  $\text{r}=\text{mid}$ 。因为本题采用  $\text{while}(\text{l}<\text{r})$  的二分架构，因此 mid 不-1。



如果从 l 到 mid 连续，那么此时  $\text{nums}[\text{mid}]$  理应就=9 说明残缺的数值一定不在左边，而且一定不是 mid，因此应该排除掉 mid 向右找，即  $\text{l}=\text{mid}+1$ 。

33. ①处应填 ( )

A. 1 B.  $\text{nums}[0]$  C. right D. left

正确答案：B

解析:例如百个数值是 100，那么 mid 卫直处理应是  $100+\text{mid}$ 。注意 mid 从 0 开始编号

34. ②处应填 ( )

A.  $\text{left}=\text{mid}+1$  B.  $\text{right}=\text{mid}-1$

C.  $\text{right}=\text{mid}$  D.  $\text{left}=\text{mid}$

正确答案：A

解析:根据上述分析,如果  $\text{mid}$  处和理应是数值相等,那么缺失的数据一定出现在右边并且  $\text{mid}$  肯定不是缺失的数据,所以要排除掉  $\text{mid}$ 。或者可以举一个只有 2 个数据的序列,分析一下,如果不这样做就会死循环。

35. ③处应填 ( )

A.  $\text{left}=\text{mid}+1$  B.  $\text{right}=\text{mid}-1$

C.  $\text{right}=\text{mid}$  D.  $\text{left}=\text{mid}$

正确答案：C

解析:根据上述分析,如果  $\text{mid}$  处和理应是数值不相等,那么缺失的数据就出现在  $l$  到  $\text{mid}$  之间,因此向左二分。因为本题采用  $\text{while}(l < r)$  的二分架构,因此  $\text{mid}$  不-1。

36. ④处应填 ( )

A.left+nums[0] B.right+nums[0]

C.mid+nums[0] D.right+1

正确答案：A

解析:根据上述例子的分析, mid 在各种情况下并没有规律的统一性。而 l 和 r 却有规律。当含有缺失的数据时:最终的 l 或 r 指向的数, 就是比缺失的数值大 1 的数。但不含缺失的数据时:指向的数据就是应该的数据。因此这里 return 回【本应该的数据】更方便后续判断。例如最终 left=5, 那么它理应是  $\text{nums}[0]+5=1+5=6$

37. ⑤处应填 ( ) |

A.nums[0]+n B.nums[0]+n-1

C.nums[0]+n+1 D.nums[n-1]

正确答案：D

解析: missing\_number 可以理解为本应该的数值⑤可以理解为【实际的数值】由上述分析, 如果不缺失, 那么最终 1 一定=n。例如原序列从 1 到 7。

```

24 int missing_number = find_missing(nums);
25 if_(missing_number == ⑤) {
26     cout << "Sequence is consecutive" << endl;
27 }else{
28     cout << "Missing number is " << ,missing
number << endl;
29 }

```

如果返回 7，nums[n-1]也是 7，就可以判断没有缺失。如果只有从 1 开始的 7 个数，而 nums[n-1] 是 8。根据以上分析，我们返回的数字一定在 1 到 7 之间那么返回的数字是不可能为 8 的，这种情况就是有缺失。

(2) (编辑距离) 给定两个字符串，每次操作可以选择删除 (Delete)、插入 (Insert)、替换 (Replace)，一个字符，求将第一个字符串转换为第二个字符串所需要的最少操作次数。

```

1 1.#include <iostream>
2
3 2.#include <string>
4
5 3.#include <vector>
6
7 4.using namespace std;

```

```
6.int min(int x,int y,int z){
7.  return min(min(x,y),z);
8.}
9.
10.int edit_dist_dp(string str1,string str2){
11.  int m=str1.length();
12.  int n=str2.length();
13.  vector<vector<int>> dp(m+1,vector<int>(n+1));
14.
```

```
15.  for(int i=0;i<=m;i++){
16.    for(int j=0;j<=n;j++){
17.      if(i==0)
18.        dp[i][j]=①;
19.      else if(j==0)
20.        dp[i][j]=②;
21.      else if(③)
22.        dp[i][j]=④;
23.      else
24.        dp[i][j]=1+min(dp[i][j-1],dp[i-1][j], ⑤);
25.    }
26.  }
27.  return dp[m][n];
28.}
29.
30.int main(){
31.  string str1,str2;
32.  cin>>str1>>str2;
```

```
33.  cout<<"Minimum number of operation:"
34.  <<edit_dist_dp(str1,str2)<<endl;
35.  return 0;
36.}
```

题目解析:

本题的思路类似于阅读程序题里的【最长公共子序列】

$dp[i][j]$ 的意思是:把第 1 个串的前  $i$  个字母, 修改成第 2 个串的前  $j$  个字母的最小修改次数。

注意  $i$  和  $j$  并不是从 0 开始编号的,  $i=0$  和  $j=0$  只是代表把空串修改成另外一个串。

1.如果  $i=0$ , 代表第 1 个串为空, 第 2 个串有  $j$  个字母, 那么只需将第 1 个串增添  $j$  个字母即可  
2.如果  $j=0$ , 代表第 1 个串有  $i$  个字母, 第 2 个串为空, 那么只需将第 1 个串删除  $i$  个字母即可

如果字符串 1 为 xxxxA, 字符串 2 为\*\*\*\*\*A,因为他们末尾字符相同, 因此修改的次数就和前面 xxxxx 和\*\*\*\*\*的修改次数相同。

如果字符串 1 为 xxxxA, 字符串 2 为\*\*\*\*\*B

因为他们末尾字符不同, 因此修改的次数有 3 种情

况:1.删掉 B 这 1 次操作(相当于把串 1 增加 B), 加修改成相等的操作次数\_xxxxxA 和 \*\*\*\*\*即  $1 + dp[i][j-1]$ 。2.删掉 A 这 1 次操作, 加上 把 xxxxx 和 \*\*\*\*\*B 修改成相等的操作次



数，即  $1 + dp[i-1][U]$  3.把 A 修改成 B 这 1 次操作，加上把 xxxxx 和\*\*\*\*\* 修改成相等的次数，即  $1 + dp[i-1][L]$  这 3 种操作就对应了第 24 句的状态转移。

38. ①处应填 ( )

A.j B.i C.m D.n

**正确答案 A**

解析:见上述分析。 $i=0$ ，代表第 1 个串为空第 2 个串有 j 个字母，那么只需将第 1 个串增添 j 个字母即可

39. ②处应填 ( )

A.j B.i C.m D.n

**正确答案 B**

解析:见上述分析。 $j=0$ ，代表第 1 个串有个字母，第 2 个串为空，那么只需将第 1 个串删除 i 个字母即可

40. ③处应填 ( )

A.  $\text{str1}[i-1] == \text{str2}[j-1]$  B.  $\text{str1}[i] == \text{str2}[j]$

C.  $\text{str1}[i-1] != \text{str2}[j-1]$  D.  $\text{str1}[i] != \text{str2}[j]$

**正确答案 A**

解析:见上述分析。末尾相等的情况。注意  $\text{dp}[i][j]$  里的  $i$  和  $j$  并不是从 0 开始编号的, 字符串的 0 号位对应的是  $i=1$ 。 $i=0$  和  $j=0$  只是代表把空串修改成另外一个串。

41. ④处应填 ( )

A.  $\text{dp}[i-1][j-1]+1$  B.  $\text{dp}[i-1][j-1]$

C.  $\text{dp}[i-1][j]$  D.  $\text{dp}[i][j-1]$

**正确答案 B**

解析:见上述分析。末尾相等的情况, 次数就和  $\text{dp}[i-1][j-1]$  是一样的

42. ⑤处应填 ( )

A.  $dp[i][j] + 1$  B.  $dp[i-1][j-1]+1$

C.  $dp[i-1][j-1]$  D.  $dp[i][j]$

**正确答案 C**

解析:见上述分析。把末尾两个字母改成一致。