

2023年CCF非专业级别软件能力认证第一轮（CSP-J）入门级C++语言 试题

题目总数：20 总分数：100

一、单项选择题

第1题 单选题

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

第2题 单选题

八进制数 $12345670_{(8)}$ 和 $07654321_{(8)}$ 的和为（ ）。

- A. $22222221_{(8)}$
- B. $21111111_{(8)}$
- C. $22111111_{(8)}$
- D. $22222211_{(8)}$

第3题 单选题

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（ ）。

```
union Data{  
    int num;  
    float value;  
    char symbol;  
};
```

union Data data;

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

第4题 单选题

假设有一个链表的节点定义如下：

```
struct Node {    int data;    Node* next;};
```

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;

第 5 题 单选题

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

第 6 题 单选题

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

第 7 题 单选题

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。

第 8 题 单选题

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

第 9 题 单选题

数 $101010_{(2)}$ 和 $166_{(8)}$ 的和为（ ）。

- A. $10110000_{(2)}$
- B. $236_{(8)}$

- C. 158₍₁₀₎
- D. A0₍₁₆₎

第 10 题 单选题

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

第 11 题 单选题

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. EDBGFCA
- B. EDGBFCA
- C. DEBGFCA
- D. DBEGFCA

第 12 题 单选题

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

第 13 题 单选题

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)

第 14 题 单选题

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

第 15 题 单选题

以下哪个不是操作系统？（ ）

- A. Linux
- B. Windows
- C. Android
- D. HTML

二、阅读程序题

第 16 - 20 题 组合题

```
1  #include<iostream>
2  #include<cmath>
3  using namespace std;
4
5  double f(double a,double b,double c){
6      double s=(a+b+c)/2;
7      return sqrt(s*(s-a)*(s-b)*(s-c));
8  }
9
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：

第 16 题 判断题

当输入为“2 2 2”时，输出为“1.7321”（ ）

- A. 正确
- B. 错误

第 17 题 判断题

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（ ）

- A. 正确
- B. 错误

第 18 题 判断题

程序总是输出四位小数（ ）

- A. 正确
- B. 错误

第 19 题 单选题

当输入为“3 4 5”时，输出为（ ）

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

第 20 题 单选题

当输入为“5 12 13”时，输出为（ ）

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

第 17 - 22 题 组合题

```
1  #include<iostream>
2  #include<vector>
3  #include<algorithm>
4  using namespace std;
5
6  int f(string x,string y){
7      int m=x.size();
8      int n=y.size();
9      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  }

```

第 17 题 判断题

f函数的返回值小于等于min (n,m) 。

- A. 正确
- B. 错误

第 18 题 判断题

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。 ()

- A. 正确
- B. 错误

第 19 题 判断题

当输入两个完全相同的字符串时，g函数的返回值总是true ()

- A. 正确
- B. 错误

第 20 题 单选题

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

第 21 题 单选题

当输入为“csp-j p-jcs”时，输出为： ()

- A. "0"
- B. "1"
- C. "T"
- D. "F"

第 22 题 单选题

当输入为“csppsc spsccp”时，输出为： ()

- A. "T"
- B. "F"
- C. "0"
- D. "1"

第 18 - 23 题 组合题

```
1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  int solve1(int n){
6      return n*n;
7  }
8
9  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 }
22
23 int main(){
24     int n;
25     cin>>n;
26     cout<<solve2(solve1(n))<<" "<<solve1((solve2(n)))<<endl;
27     return 0;
28 }
```

第 18 题 判断题

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（ ）

- A. 正确
- B. 错误

第 19 题 判断题

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（ ）

- A. 正确
- B. 错误

第 20 题 判断题

如果输入的n为质数，solve2（n）的返回值为n²+1（ ）

- A. 正确
- B. 错误

第 21 题 单选题

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

第 22 题 单选题

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

第 23 题 单选题

当输入为“5”时，输出为（ ）

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

三、完善程序

第 19 - 23 题 组合题

(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 的开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  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  int find missing(vector<int>& nums) (
7  int left = 0, right = nums.size() - 1;
8  while (left < right){
9      int mid = left + (right - left) / 2;
10     if (nums[mid] - mid + ① ) (
11         ② ;
12     }else{
13         ③
14     }
```



```

15     }
16     return      ④      ;
17 }
18
19 int main() (
20     int n;
21     cin >> n;
22     vector<int> nums(n);
23     for (int i= 0; i< n; i++) cin >> nums[i];
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     }
30     return 0;
31 }

```

第 19 题 单选题

①处应填 ()

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

第 20 题 单选题

②处应填 ()

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

第 21 题 单选题

③处应填 ()

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

第 22 题 单选题

④处应填 ()

- A. left+nums[0]
- B. right+nums[0]
- C. mid+nums[0]

D. right+1

第 23 题 单选题

⑤处应填 ()

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

第 20 - 24 题 组合题

(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  #include <iostream>
2  #include <string>
3  #include <vector>
4  using namespace std;
5
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<<"Mininum number of operation:"
34 <<edit_dist_dp(str1,str2)<<endl;
```

```
35 | return 0;  
36 | }
```

第 20 题 单选题

①处应填 ()

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

第 21 题 单选题

②处应填 ()

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

第 22 题 单选题

③处应填 ()

- A. str1[i-1]==str2[j-1]
- B. str1[i]==str2[j]
- C. str1[i-1]!=str2[j-1]
- D. str1[i]!=str2[j]

第 23 题 单选题

④处应填 ()

- A. dp[i-1][j-1]+1
- B. dp[i-1][j-1]
- C. dp[i-1][j]
- D. dp[i][j-1]

第 24 题 单选题

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]