

非专业级别软件能力认证第一轮

(CSP-J1) 入门级 C++ 语言模拟卷

考生注意事项：

- 不得使用任何电子设备（如计算器、手机、电子词典等）或查阅任何书籍资料。

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 if 语句中，对于充当条件的表达式，下面描述正确的是()。

- A. 必须是关系表达式
- B. 可以是任意表达式
- C. 必须是关系表达式或逻辑表达式
- D. 必须是关系表达式或算术表达式

2. 下列函数的定义中，有语法错误的是()。

- A. `void fun(int x, int *y){*x *= *y;}`
- B. `void fun(int *x, int y){*x += y;}`
- C. `int * fun(int *x, int y){return x += y;}`
- D. `void fun(int x, int *y){x *= *y;}`

3. 某班同学一个 48 人每个人都报名了兴趣班，兴趣班一共有两种，报名书法班的有 29 人，报名绘画班的有 32 人，一有多少人两个兴趣班同时报了名()。

- A. 13
- B. 15
- C. 10
- D. 12

4. 现在有 8 个学生要颁发 3 个相同的奖杯，每个人只能领取一个，那一共有()种颁发方式。

- A. 32
- B. 56
- C. 336
- D. 72

5. 目前使用的计算机的主要部件是由()构成的。

- A. 集成电路
- B. 大规模集成电路
- C. 晶体管
- D. 电子管

6. 现在有一张分辨率为 2048*2048 的 64 位真彩色图象。请问存储这张图像,需要多大的存储空间 ()。

- A. 8MB
- B. 16MB
- C. 32MB
- D. 64MB

7. 下面函数实现了在一个升序整型数组 arr 中查找一个目标值 target 的位置, 请分析它的时间复杂度为 ()。

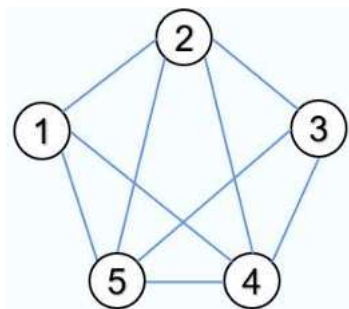
```
int search(int start, int end, int target, int *arr){
    if(start <= end) {
        unsigned int mid = start + (end - start)/2;
        if (arr[mid] == target)
            return mid;
        else if (target > arr[mid])
            return search(mid + 1, end, target, arr);
        else if (target < arr[mid])
            return search(start, mid - 1, target, arr);
    }
    return -1;
}
```

A. $O(1)$ B. $O(\log N)$ C. $O(N)$ D. $O(N\log(N))$

8. 有一个顺序有序表 $S[0,15]$ 中有 16 个元素, 则采用二分法查找元素 $S[12]$ 的过程中比较元素的顺序为()。

- A. $S[7]$, $S[12]$, $S[13]$, $S[12]$
- B. $S[8]$, $S[11]$, $S[14]$, $S[12]$
- C. $S[7]$, $S[11]$, $S[13]$, $S[12]$
- D. $S[8]$, $S[11]$, $S[13]$, $S[12]$

9. 在下图的多边形 12345 中, 从下面选项中哪一点出发, 可以遍历图上的每条边一次, 而且仅遍历一次()。



- A. 2 点 B. 3 点 C. 4 点 D. 5 点

10. 有一棵哈夫曼树中含有 n 个叶结点, 如果用下面的代码中存储该树中的结点, 总共会产生()个空指针。

```
struct NodeTree {
```

```
struct NodeTree *left;
struct NodeTree *right;
char value;
}
```

A. $4n$ B. $2n-1$ C. $2n+1$ D. $2n$

11. 以下有关链表的描述不恰当的一项是()。

- A. 链表通常包括单向链表、双向链表、单向循环链表和双向循环链表。
- B. 链表属于链式存储结构。
- C. 链表的大小可以动态的增加或减少。
- D. 链表中的每个节点只包括元素值和指向下一个节点的指针。

12. 已知后缀表达式为 $6\ 2\ +\ 3\ *\ 4\ 2\ /\ 3\ *\ -\ 5\ +$ ，其前缀表达式为()。

- A. $- + + 6 * 2 3 * / 4 2 3 5$
- B. $- + * + 6 2 3 * / 4 2 3 5$
- C. $+ - + 6 * 2 3 * / 4 2 3 5$
- D. $+ - * + 6 2 3 * / 4 2 3 5$

13. f 表示循环队列 q 的队首元素位置， r 表示 q 的队尾元素位置的下一个位置，若 q 容量为 30 (序号为 0~29)， f 的值为 17， r 的值为 4，则队列中元素个数为()。

- A. 13 B. 14 C. 16 D. 17

14. 已知二叉树先序序列为 ABCDEFGHI，中序序列为 CDBAFEHIG，则后序序列为()。

- A. DCBFHIGEA B. DCBAFEIHG
- C. DCBFIHGEA D. DCBFEIHGA

15. 存在一个含有 12 个顶点的有向图，该图中入度为 1 的点有 5 个，入度为 2 的点有 3 个，入度为 4 的点 4 个，则该图所有顶点的出度之和为()。

- A. 12 B. 24 C. 27 D. 54

二、阅读程序(程序输入不超过数组或字符串定义的范围; 判断题正确填√, 错误填×; 除特殊说明外, 判断题 1.5 分, 选择题 3 分, 共计 40 分)

(1)

```
01 #include <iostream>
02 #include <cmath>
03 using namespace std;
04 const int N=1001;
05 int a[N],n,k,l=0;
06 int main() {
07     long long ans=0;
08     cin >> n >> k;
09     while(k)    a[++l]=k%2,k/=2;
10     for(int i=l;i>=1;i--)
11         ans+=pow(n,i-1)*a[i];
12     cout << ans;
13     return 0;
14 }
```

假设输入的所有数都为不超过 100 的正整数, 完成下面的判断题和单选题:

判断题

16. 当输入为 "5 4" 时, 输出为 "20" ()。
17. (1 分) 将第 9 行的 "++l" 改成 "l++", 对结果不会有任何影响()。
18. 第 9 行实现的功能是将 k 转换成二进制()。

单选题

19. 当输入为 "3 100" 时, 输出为()。
- A. "965" B. "981" C. "976" D. "1107"
20. (2 分) 当输入为 "6 9" 时, 输出为()。
- A. "217" B. "209" C. "246" D. "196"

(2)

```
01 #include <bits/stdc++.h>
02 using namespace std;
03
04 void func(string str, int n, int len)
05 {
06     char tmp[n][len] = {};
07     char ans[n * len + 1] = {};
08     for (int i = 0; i < n; i++)
09     {
10         for (int j = 0; j < len; j++)
```

```

11     {
12         tmp[i][j] = str[i * len + j];
13     }
14 }
15 int index = 0;
16 for (int i = 0; i < len * n; i++)
17 {
18     for (int j = 0; j < n; j++)
19     {
20         if (tmp[j][i] != '\0' && index < len * n)
21         {
22             ans[index] = tmp[j][i];
23         }
24         index++;
25     }
26 }
27 cout << ans << endl;
28 }
29
30 int main()
31 {
32     int len, n;
33     string str;
34     cin >> str >> n;
35     len = str.size();
36     func(str, n, (len + n - 1) / n);
37     return 0;
38 }

```

判断题

21. 当 $n=1$ 时,输出字符串和输入字符串相同 ()。
22. 将第 36 行的 $(len+n-1)/n$ 改成 len/n 结果不变 ()。
23. 将第 20 行 $tmp[j][i] != '\0'$ 修改成 $tmp[j][i] != 0$ 结果不变 ()。

单选题

24. 若输入 cpcpsssj 2, 输出结果是 ()。
 A. cspjcspj B. cspscspj C. pspcscjs D. scpjscps
25. 若输入 9sj8od5/n.9p 4, 输出内容为 ()。
 A. 998.jp/sdn5o B. 95s/jn8.o9dp
 C. 9859.so/jdnp D. 985.so/9jdn

(3)

```
01 #include <iostream>
02 using namespace std;
03 int a[] = {2, 3, 5, 7}, b[] = {1, 3, 7, 9};
04 int ans;
05 bool check(int x){
06     if(x == 1) return 0;
07     for(int i = 2; i * i <= x; i++){
08         if(x % i == 0){
09             return 0;
10         }
11     }
12     return 1;
13 }
14 void dfs(int x, int n){
15     if(x > n){
16         return;
17     }
18     ans++;
19     for(int i : b){
20         if(check(x * 10 + i)){
21             dfs(x * 10 + i, n);
22         }
23     }
24 }
25 int main(){
26     int num;
27     cin >> num;
28     for(int i : a){
29         dfs(i, num);
30     }
31     cout << ans;
32     return 0;
33 }
```

判断题

26. 如果将第 6 行删除，程序结果不变()。
27. 输入 num 的值为 100 和 200 时输出的结果相同()。
28. 每当第 18 行 ans++ 执行时，dfs 函数中 x 的值都为奇数()。

单选题

29. 当输入为 30 时, 输出为()。
- A. 2 B. 6 C. 10 D. 30
30. 当输入为 10 时, 第 21 行的执行次数为()。
- A. 3 B. 6 C. 9 D. 12

三、完善程序 (单选题, 每小题 3 分, 共计 30 分)

(1) (数字查找)

问题: 给定一个升序数组和一个目标值, 在数组中找到目标值, 并返回其索引。如果目标值不存在数组中, 返回它将会被按顺序插入的位置。

```
01 #include <bits/stdc++.h>
02 using namespace std;
03 int searchInsert(int ①, int target, int n) {
04     int left = 1, right = n, ans = ②;
05     while (left <= right) {
06         int mid = (right + left) / 2;
07         if (target ③ nums[mid]) {
08             ans = mid;
09             right = mid - 1;
10         } else {
11             left = mid + 1;
12         }
13     }
14     return ④;
15 }
16 int main() {
17     int n, nums[1001] = {}, target;
18     cin >> n;
19     for (int i = 1; i <= n; i++) cin >> nums[i];
20     cin >> target;
21     int ret = ⑤;
22     cout << ret;
23     return 0;
24 }
```

单选题

31. ①处填写的代码为()。
- A. num B. num[] C. nums D. nums[]
32. ②处填写的代码为()。
- A. 0 B. 1 C. n+1 D. n

33. ③处填写的代码为()。
- A. != B. <= C. > D. <
34. ④处应该填()。
- A. mid B. left C. right D. ans
35. ⑤处应该填()。
- A. searchInsert(nums,target,n) B. searchInsert(nums[],target,1)
C. searchInsert(nums,target) D. searchInsert()

(2) (神奇序列)

对于一个小于—亿的正整数 x , 这个 x 所有大于 1 的因子可以组成一些神奇序列, 该序列满足任意前一项元素 x_i 都能整除后一项 x_{i+1} , 求这个正整数 x 所能组成的神奇序列的最大长度, 以及满足最大长度的神奇序列的个数。

例如: $x=100$ 时, 大于 1 的约数有 2、4、5、10、20、25、50、100。

能够组成最长的神奇序列的长度是 4;

满足最大长度的神奇序列的个数有 6 个, 分别是:

2 4 20 100、2 10 20 100、2 10 50 100、5 10 20 100、5 25 50 100、5 10 50 100。

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 int dp[10010][2],ans1,ans2;
4 vector<int> get_div(int x){
5     vector<int> res;
6     for(int i=1; i<=x/i; i++){
7         if (x%i == 0){
8             res.push_back(i);
9             if(①) res.push_back(x/i);
10        }
11    }
12    ②;
13    return res;
14 }
15 void solve(vector<int> &v){
16     memset(dp,0,sizeof dp);
17     dp[0][0]=0,dp[0][1]=1;
18     ans1=0,ans2=0;
19     for(int i=1; i<v.size(); i++){
20         int a=v[i];
21         for(int j=0; ③;j++){
22             int b = v[j];
23             if(④){
```



```

24         dp[i][0]=dp[j][0]+1;
25         ____⑤____;
26     }else if(a%b==0 && dp[j][0]+1==dp[i][0]){
27         dp[i][1]+=dp[j][1];
28     }
29 }
30 if (dp[i][0] > ans1){
31     ans1 = dp[i][0];
32     ans2 = dp[i][1];
33 }
34 }
35 cout<<ans1<<" "<<ans2<<endl;
36 }
37 int main(){
38     int n = 0;
39     cin >> n;
40     vector<int> v=get_div(n);
41     solve(v);
42     return 0;
43 }

```

单选题

36. ①处应填 ()
 A. $x/i \neq i$ B. $i/x \neq i$ C. $x/i == i$ D. $x \% i \neq 0$
37. ②处应填 ()
 A. if (x > 1) res.push_back(x)
 B. sort(res.begin(), res.end())
 C. if (x / sqrt(x) == 0) res.push_back(sqrt(x))
 D. ans1 = ans2 = res.size()
38. ③处应填 ()
 A. $j \leq i$ B. $i \geq j$ C. $j > i$ D. $i > j$
39. ④处应填 ()
 A. $a \% b == 0 \ \&\& \ dp[j][0] + 1 > dp[i][0]$
 B. $a \% b == 0 \ \&\& \ dp[i][0] + 1 > dp[j][0]$
 C. $a \% b \neq 0 \ \&\& \ dp[j][0] + 1 == dp[i][0]$
 D. $a \% b \neq 0 \ || \ dp[j][0] + 1 == dp[i][0]$
40. ⑤处应填 ()
 A. $dp[i][1] = dp[i][0]$ B. break
 C. $dp[i][1] = dp[j][1]$ D. $dp[i][1] += dp[j][1]$