2021 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 模拟卷 - 6

考生注意事项：

1．全部试题答案均要求写在答卷纸上，写在试卷纸上一律无效。

2．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、单选题（共 15 题，每题 2 分，共计 30 分；每题有且仅有一个正确答案）

1． 关于图灵机下面的说法哪个是正确的？（）

A．图灵机是世界上最早的电子计算机

B．由于大量使用磁带操作，图灵机运行速度很慢

C．图灵机只是一个理论上的计算模型

D．图灵机是英国人图灵发明的，在二战中为破译德军的密码发挥了重要作用

2． 已知大写字母 A 的 ASCII 编码为 65(十进制），则大写字母 J 的十六进制 ASCII 编码为（ ）。

A．48　　B．49　　C．50　　D．以上都不是

3． 在字长为 16 位的系统环境下，一个 16 位带符号整数的二进制补码为 1111111111101101.其对应的十进制整数应该是（）。

A.19　　B．-19　　C．18　　D．-18

4． 一个包含 n 个分支结点（非叶结点）的非空满 k 叉树，k≥1,它的叶结点数目为（）。

A.nk+1　　B.nk-1　　C.(k+1)n-1　　D.(k-1)n+1

5． 表达式 a*(b+c)-d 的后缀表达式是（）。

A.abcd*+-　　B.abc+*d-　　C.abc*+d-　　D.-+*abcd

6． 最优前缀编码，也称 Huffman 编码。这种编码组合的特点是给予较频繁使用的元素较短的唯一编码，以提高通信的效率。下面编码组合哪一组不是合法的前缀编码？（）。

A.(00,01,10,11)　　B．(0,1,00,11)

C.(0,10,110,111)　　D．(1,01,000,001)

7． 关于 CPU 下面哪个说法是正确的？（）

A.CPU 全称为中央控制器

B.CPU 能直接运行机器语言

C.CPU 最早是由 Intel 公司发明的

D.同样主频下，32 位的 CPU 比 16 位的 CPU 运行速度快一倍

8． 若二叉树的高度为 n,其根节点高度为 0,则其节点数一定可以为（ ）个。

A.n　　B.n*n*n　　C.2^n　　D.n!

9． 设待排序的记录为（49，38，65，97，76， 13，27 ，48，55，4）,每次可以交换任意两个数字，则需要交换（ ）次使得序列元素可以从小到大有序。

A.5　　B.8　　C.9　　D.10

10. 在所有的 3 位数中(不含前导 0)，满足个位<=十位<=百位的数字有( )个

A. 700　　B.300　　C.299　　D.219

11. 若 3 个顶点的无权图 G 的邻接矩阵用数组存储为{{0, 1, 1},{1, 0, 1},{0, 1, 0},假定在具体存储中顶点依次为：v_1,v_2,v_3。关于该图，下面的说法哪个是错误的？（）

A.该图是有向图

B.该图是强联通的

C.该图所有顶点的入度之和减所有顶点的出度之和等于 1

D.以 v1 为起点对图 G 进行深度优先搜索遍历，有不止 1 种遍历的方案。

12. 在带尾指针（链表指针 clist 指向尾结点）的非空循环单链表中每个结点都以 next 字段的指针指向下一个结点。假定其中已经有了 2 个以上的结点。下面哪个说法是正确的？（)

A.如果 p 指向一个待插入的新结点，在头部插入一个元素的语句序列为

```
p->next=clist;clist->next=p;
```

B.如果 p 指向一个待插入的新结点，在尾部插入一个元素的语句序列为

```
p->next=clist;clist->next=p;
```

C.在头部删除一个结点的语句序列为

```
p=clist->next;clist->next=clist->next->next;delete(p);
```

D.在尾部删除一个结点的语句序列为 `p=clist;clist=clist->next;delete(p);`

13. 设散列表的地址空间为 0 到 10,散列函数为 h(hk)=k mod 11,用线性探查法解决碰撞（线性探查法指发生冲突时，往后移动一位继续检测的方法）。现从空的散列表开始，依次插入关键码值 95, 14, 27, 68, 82,则最后一个关键码 82 的地址为（）

A.4　　B.5　　C.6　　D.7

14. 排序算法是稳定的意思是关键码相同的记录排序前后相对位置不发生改变，下列哪种排序算法不是稳定的？（）

A.插入排序　　B.基数排序　　C.归并排序　　D.堆排序

15. 计划展出 10 幅不同的画，其中 1 幅水彩画、4 幅油画、5 幅国画，排成一行陈列，要求同一品种的画必须连在一起，并且水彩画不放在两端，那么不同的陈列方式有（）种。

A.2880　　B.17280　　C.8640　　D.5760

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 4 分，共计 40 分）

1.

```
1   #include <cmath>
2   #include <cstdio>
3
4   int a1[1001];
5   int i, j, t, t2, n;
6   int main() {
7       scanf("%d", &n);
8       for (i = 2; i <= sqrt(n); i++) {
9           if (a1[i] == 0)
10          {
11              t2 = n / i;
12              for (j = 2; j <= t2; j++)
13                  a1[i * j] = 1;
14          }
15      }
16      t = 0;
17      for (i = 2; i <= n; i++)
18          if (a1[i] == 0)
19          {
20              printf("%d", i);
21              t++;
22              if (t % 10 == 0)
23                  printf("\n");
24          }
25      printf("\n");
26      return 0;
27  }
```

数据约定输入的 n 为正整数。

- 判断题

  1)若将第 8 行的"i=2"改为"i=1",程序结果会发生变化。（）

  2)若将第 17 行的"i=2"改为"i=1",程序结果会发生变化。（）

  3)若去掉第 9 行的 if 语句及其括号（不去掉大括号内语句），程序结果会发生变化（）

  4)当 n 大于 2 时，程序运行到第 25 行时，t 的值为$\sqrt{n}$(向下取整）。（）

- 选择题

  5)(3 分）若程序最终 t 的值为 25,则 n 的输入可以有（）种方案。

  A. 1　　B. 2　　C. 3　　D. 4

  6)(3 分）若输入的 n 为 50,那么输出的所有数字中最大值为（）。

  A. 47　　B. 48　　C. 49　　D. 50

2.

```
#include  <ctype.h>
#include  <stdio.h>
void expand(char s1[], char s2[])
{
    int i, j, a, b, c;
    j = 0;
    for (i = 0; (c = s1[i]) != '\0'; i++)
        if(c == '-')
        {
            a = s1[i - 1]; b = s1[i + 1];
            if (isalpha(a) && isalpha(b) || isdigit(a) && isdigit(b))
            {
                j--;
                do
                    s2[j++] = a++;
                while (tolower(a) < tolower(s1[i + 1]) && isalpha(a));
            }
            else s2[j++] = c;
        } else s2[j++] = c;
    s2[j] = '\0';
}
int main()
{
    char s1[100], s2[300];
    gets(s1);
    expand(s1, s2);
    printf("%s\n", s2);
}
```

函数 isalpha(a)用于判断字符 a 是否为字母，isdigit(b)用于判断字符 b 是否为数字，如果是，返回 1,否则返回 0.

函数 tolower(a)的功能是当字符 a 是大写字母，返回其小写字母，其余情况不变

规定：输入的字符串只包含大小写字母，数字和“—”，且保证“—”不会出现在首位和末位。

- 判断题

  1)若输入的字符串不包含“—”号，则输出的字符串和输入相同。（）

  2)若输入的字符串包含“—”号，输出的字符串长度可能与输入的相同。（）

  3)若存在一个字符，只存在于 s2 字符串中而不存在于 s1 字符串中，则 s1 字符串中一定存在“—”号。（ ）

  4)若存在一个字符，只存在于 s2 字符串中而不存在于 s1 字符串中，则该字符一定不是大写字母字符。（ ）

- 选择题

  5)若输入 6 个字符，最多输出（ ）个字符。

  A. 6　　B. 20　　C. 52　　D. 54

  6)若输出 500 个字符，至少输入（）个字符。

  A. 20　　B. 59　　C. 60　　D. 500

3.

```
1  #include<iostream>
2  #include<cstring>
3  using namespace std;
4  const int mm = 1007;
5  int map[mm][mm];
6  int n;
7  int main() {
8      cin >> n;
9      n--;
10     memset(map, 0, sizeof(map));
11     for (int i = 0; i < n; i++) {
12         for (int j = 0; j < n; j++) {
13             map[i][j] = (i + j) % n + 1;
14         }
15     }
16     for (int i = 0; i < n; i++) {
17         map[i][n] = map[n][i] = map[i][i];
18         map[i][i] = 0;
19     }
20     for (int i = 0; i <= n; i++) {
21         for (int j = 0; j <= n; j++) {
22             cout << map[i][j] << "";
23         }
24         cout << endl;
25     }
26     return 0;
27 }
```

规定：输入的 n 为偶数。

- 判断题

  1)若 n=6,程序执行结束时 map[3][6]的值是 4.( )

  2)当程序执行完第 15 行时，此时 map 数组形成一个矩阵（值不为 0 的部分），且该矩阵关于从左上到右的对角线对称。( )

  3)输出的值形成一个矩阵，且该矩阵关于从左上到右下的对角线对称。（ ）

  4)值为 x 的位置有(i1,j1)，(i2,j2)，...，则任意两个位置的 i 都不相同，任意两个位置的 j 都不相同。（ ）

- 选择题

  5)当输入 n 的值为 8 时，输出的第 4 行的值的和是（）。

  A. 28　　B. 36　　C. 8　　D. 7

  6)输出的从右上到左下对角线的值的和是（）。

  A. 0　　B. n　　C. n(n-1)/2　　D. (n+1)n/2

三、完善程序（单选题，每小题 3 分，共计 30 分）

1.(求字符的逆序）下面的程序的功能是输入若干行，每行一个整数和一个字符串，你需要将字符串逆序输出。如果输入的整数为-1，则终止程序。请将程序补充完整。

例如，输入为 abc（换行）bcd（换行）ddf（换行）-1

则输出为 cba（换行）dcb（换行）fdd（换行）

```
1   #include <bits/stdc++.h>
2   int maxline = 200, kz;
3   int reverse (char s[] ) {
4       int i, j, t;
5       for (i = 0, j = strlen(s) - 1; i < j; ①, ②)
6       {
7           t = s[i];
8           s[i] = s[j];
9           s[j] = t;
10      }
11      return (0);
12  }
13  int main () {
14      char line[100];
15      cin >> kz;
16      while (③)
17      {
18          cin >> line;
19          ④;
20          cout <<⑤<< end1;
21          cout << "continue? -1 for end. " << endl;
22          cin >> kz;
23      }
24  }
```

1) ①处应填（ ）。

A.i++　　B.i+=2　　C.i--　　D.i=s[i]

2) ②处应填（ ）。

A.j=i　　B.i-=strlen(s)　C.j--　　D.j-=i

3) ③处应填（ ）。

A.kz　　B.kz%2==1　　C.kz!=-1　　D.kz>0

4) ④处应填（）。

A.kz=0　　B.reverse(line-1)　C.reverse(line)　D.reverse(line+1)

5) ⑤处应填（）。

A.reverse(line)B.line　　C.reverse(line+1)　D.line+1

2.(棋盘覆盖问题）在一个 2x2*个方格组成的棋盘中恰有一个方格与其他方格不同（如图中标记为－1 的方格），称之为特殊方格。现用 L 形（占 3 个小方格）纸片覆盖棋盘上除特殊方格的所有部分，各纸片不得重叠，于是，用到的纸片数恰好是（4*-1)/3.在下面给出的覆盖方案例子中，k=2,相同的 3 个数字构成一个纸片。下面给出的程序使用分治法设计的，将棋盘一分为四，依次处理左上角、右上角、左下角、右下角，递归进行。
请将程序补充完整。
例：

2	2	3	3
2	-1	1	3
4	1	1	5
4	4	5	5

－1 为特殊方格的位置，其他数字表示 L 形方格放置的顺序。

```
1   #include <bits/stdc++.h>
2   using namespace std;
3   int board[65][65], tile; /* tile 为纸片编号 */
4   void chessboard(int tr, int tc, int dr, int dc, int size)
5   /* dr,dc 依次为特殊方格的行、列号 */
6   {
7       int t, s;
8       if (size == 1)
9           return;
10      t = tile++;
11      s = size / 2;
12      if (①)
13          chessboard(tr, tc, dr, dc, s);
14      else{
15          board[tr + s - 1][tc + s - 1] = t;
16          ②;
17      }
18      if (dr < tr + s && dc >= tc + s)
19          chessboard(tr, tc + s, dr, dc, s);
20      else {
21          board[tr + s - 1][tc + s] = t;
22          ③;
23      }
24      if (dr >= tr + s && dc < tc + s)
25          chessboard(tr + s, tc, dr, dc, s);
26      else {
27          board[tr + s][tc + s - 1] = t;
28          ④;
```

```
29      }
30      if (dr >= tr + s && dc >= tc + s)
31          chessboard(tr + s, tc + s, dr, dc, s);
32      else {
33          board[tr + s][tc + s] = t;
34          ⑤; }
35  }
36  void prt1(int b[][65], int n)
37  {
38      int i, j;
39      for (i = 1; i <= n; i++)
40      {
41          for (j = 1; j <= n; j++)
42              cout << setw(3) << b[i][j];
43          cout << endl;
44      }
45  }
46  void main()
47  {
48      int size, dr, dc;
49      cout << "input size(4/8/16/64):" << endl;
50      cin >> size;
51      cout << "input the position of special block(x,y):" << end1;
52      cin >> dr >> dc;
53      board[dr][dc] = -1;
54      tile++;
55      chessboard(1, 1, dr, dc, size);
56      prt1(board, size);
57  }
58
```

1) ①处应填（ ）。

A.(dr<tr+s)&&(dc>=tc+s)B.(dr<=tr+s)&&(dc<=tc+s)

C.(dr<tr+s)&&(dc<tc+s)D.(dr>=tr+s)&&(dc<tc+s)

2)②处应填（ ）。

A.chessboard(tr+s,tc,tr+s,tc+s-1,s)

B.chessboard(tr,tc,tr+s-1,tc+s-1,s)

C.chessboard(tr,tc+s,tr+s-1,tc+s,s)

D.chessboard(tr,tc,tr+s,tc+s,s)

3)③处应填（ ）。

A.chessboard(tr,tc,tr+s-1,tc+s-1,s)

B.chessboard(tr+s,tc,tr+s,tc+s-1,s)

C.chessboard(tr+s,tc+s,tr+s,tc+s,s)

D.chessboard(tr,tc+s,tr+s-1,tc+s,s)

4)④处应填（).

A.chessboard(tr,tc+s,tr+s-1,tc+s,s)

B.chessboard(tr+s,tc+s,tr+s,tc+s,s)

C.chessboard(tr,tc,tr+s-1,tc+s-1,s)

D.chessboard(tr+s,tc,tr+s,tc+s-1,s)

5)⑤处应填（）。

A.chessboard(tr,tc,tr+s-1,tc+s-1,s)

B.chessboard(tr,tc+s,tr+s-1,tc+s,s)

C.chessboard(tr+s,tc,tr+s,tc+s-1,s)

D.chessboard(tr+s,tc+s,tr+s,tc+s,s)