2022 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 - 3

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答案）

1．某网站的网址为 cxy.edu.cn，则其是一个（ ）的网站。

A． 政府官方 B． 教育部门 C． 个人性质 D． 商业性质

2．表达式 a+b*c-(d+e)的前缀形式（ ）。

A． -+a*bc+de B． -+*abc+de

C． abc*+de+- D． abcd*++-

3．下列选项中与$(1011)_2$ * $(1101)_2$的值相等的是（ ）。

A． $(11000)_2$ B． $(1001111)_2$ C． $(10001111)_2$ D． $(1001101)_2$

4．设 A=true，B=false，C=true，D=false，以下逻辑运算表达式值为真的是（ ）。

A.(A ∧ B) ∨(C ∧ D ∨ ¬A) B.((¬A ∧ B) ∨ C) ∧¬D

C.(B ∨ C ∨ D) ∧ D ∧ A D. A ∧ (D ∨ ¬C) ∧ B

5．一个正整数可以表示为若干个不同的正整数的加和，如数字 3 可表示为 3=3，3=1+2，其中 3=2+1 认为与 3=1+2 是一种表示方法，则数字 10 的表示方法有（ ）种。

A． 10 B． 7 C． 8 D． 9

6．现有两个 5 层的二叉树（根节点为第 1 层），则这两个二叉树节点数差的最大可能为（ ）。

A． 15 B． 31 C． 22 D． 26

7．在一个结点数为 n 的无向图中，至少需要（ ）条边才可以保证任意两点都可以相互到达。

A． 1 B． n-1 C． n D． (n-1)*n/2

8．通过二叉树的（ ）就可以唯一的确定二叉树的形态。

A． 前缀和后缀表达式 B．中缀表达式

C． 后缀表达式 D．前缀和中缀表达式

9．3^{2020}的个位数字为（ ）。

A． 3 B． 9 C． 7 D． 1

10．线性表采用链式存储时，结点的存储地址（ ）。

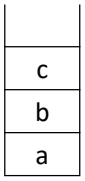
A． 必须不连续 B． 连续与否均可

C． 必须是连续的　　D． 和头节点的存储地址相连续

11．周末小明和爸爸妈妈三人一期做了三道菜，小明负责洗菜，爸爸负责切菜，妈妈负责炒菜，假设每道菜的顺序都是：先洗菜 10 分钟，再切菜 10 分钟，最后炒菜 10 分钟，那么做一道菜需要 30 分钟，注意，两道不同菜的相同步骤不能同时进行，例如第一道菜和第二道菜不能同时洗菜，也不能同时切菜，那么做完这三道菜最短时间是（ ）分钟。

A． 90　　B． 60　　C． 50　　D． 40

12. 如果一个栈初始时为空，且当前栈中的元素从栈底到栈顶依次为 a，b，c（如右图所示），另有元素 d 已经出栈，则可能的入栈顺序是（ ）。



A. a, d, c, b　　B. b, a, c, d

C. a, c, b, d　　D. d, a, b, c

13．归并排序在最坏的情况下时间复杂度为（ ）。

A． $O(n)$　　B． $O(n^2)$　　C． $O(n\log_2 n)$　　D． $O(n^2\log_2 n)$

14．下列选项中与$(0.8)_{16}$相等的是（ ）。

A． $(0.1)_3$　　B． $(0.4)_8$　　C． $(0.6)_{10}$　　D． $(0.7)_{13}$

15．int 为 32 位整数数据类型，如果定义 int a[1024*1024]; a 数组占据内存大小为（ ）。

A． 1MB　　B． 4KB　　C． 4MB　　D． 4GB

二、阅读程序（程序输入不超过数组或字符串定义的范围；判断正确填√，错误填×；共计40分）

1.

```
1  #include <iostream>
2  using namespace std;
3  int n, m, a[100], b[100], cnt = 0;
4  int main() {
5      cin >> n >> m;
6      for(int i = 1; i <= n; i++) cin >> a[i];
7      for(int j = 1; j <= m; j++) cin >> b[j];
8      int h1 = n, h2 = m;
9      while(h2 > 0) {
10         int t = a[h1];
11         a[h1] = b[h2];
12         b[h2] = t;
13         h1--;
14         h2--;
15     }
16     for(int i = 1; i <= min(n,m); i++) {
17         if (a[i] - b[i] > 0) cnt++;
18         if (a[i] - b[i] < 0) cnt--;
19     }
20     cout << cnt << endl;
21     return 0;
22 }
```

保证所有输入均不会超出数组定义的范围，读入的所有变量不会超过其类型范围。

- 判断题（每题2分）

1) 若n、m相等且对任意i来说a[i]和b[i]为同一个数，则程序输出的结果为0（）

2) 如果n<m，则程序会出现运行错误（ ）

3) 如果n=1或m=1，则输出的结果只可能是1或者-1（ ）

4) 将第16行的min(n,m)替换为max(n,m)，程序输出的结果不会发生改变（ ）

- 选择题（每题2.5分）

5) 如果n=100，m=5，程序输出的最大值可能是多少（ ）

A. 1　B. 4　C. 5　D. 100

6) 如果n=100，m=20，程序输出的最小值可能是多少（ ）

A. -100　B. -80　C. -20　D. 0

2.

```
#include<iostream>
using namespace std;
int n;
int a[1000],b[1000],c[1000];
int main() {
    cin >> n;
    for (int i = 1; i <= n; i++) {
        cin >> a[i];
        b[ a[i] ] = i;
    }
    for (int i = 1; i <= n; i++) {
        c[i] = c[i - 1] + i * b[i];
    }
    int cnt = 0;
    for (int i = 1; i <= n; i++) {
        if(b[ a[i] ] == a[i]) cnt++;
    }
    cout << cnt << " " << c[n] << endl;
    return 0;
}
```

约定 n 为区间[1,888]的整数，数组 a 中的元素为[1,n]区间内的整数且不重复。

- 判断题（每题2分）

1) 若i属于区间[1,n],则c[i]>=c[i-1]+(i-1)一定成立。（ ）

2) 若n>2且最终cnt=n-2，a数组可能的组合有6种。（ ）

3) 若n>1,则存在某种组合使得cnt的值为0。（ ）

4) 对于任意输入n，c[n]一定大于n*n。（ ）

- 选择题（每题2.5分）

5) 若输入n为3，数组为1 2 3，则输出为（ ）

A. 3 14　　B. 0 14　　C. 3 10　　D. 0 10

6) 当输入n的值为5时，c[5]可能的最大值为（ ）

A. 25　　B. 35　　C. 55　　D. 105

3.

```
#include <bits/stdc++.h>
using namespace std;
int ans,deep,n;
int a[1000];
void fun(int L,int R) {
    if(L > R) return ;
    int xb = L;
    for (int i = L; i <= R; i++)
        if(a[i] < a[xb]) xb = i;
    deep++;
    ans += deep*a[xb];
    fun(L, xb - 1);
    fun(xb + 1, R);
}
int main() {
    cin >> n;
    for (int i = 0; i < n; i++) {
        cin >> a[i];
    }
    fun(0, n-1);
    cout << ans <<endl;
    return 0;
}
```

保证 1<=n<=888，数组 a 中的元素在[-10,10]区间内。

- 判断题（每题2分）
  1) 程序结束时，deep的值等于n。（ ）
  2) 将程序中第12行和第13行互换，程序运行的结果不会发生改变（ ）
  3) 如果a数组中 所有元素的和 恰好等于0时，输出的ans值一定是正数。（ ）
  4) 如果a数组中 所有元素 都是1，输出的ans的值等于(n+1)*n/2（ ）
- 选择题（每题3分）
  5) 最好情况下，程序的时间复杂度是（ ）

  A. O(n)　　B. $O(n^2)$　　C. O(nlogn)　　D. $O(n^2logn)$

  6) 如果n=5,且输入的a[i]依次为5 4 3 2 1时，程序的输出是（ ）

  A.25　　B. 35　　C. 55　　D. 105

三．完善程序（单选题，每小题 3 分，共计 30 分）

1.（高精度计算）由于计算机运算的数据范围表示有一定限制，如整型 int 表达范围是（-2＾31～2＾31-1），unsigned long（无符号整数）是（0～2＾32-1），都约为几十亿，因此在计算位数超过十几位的数时，不能采用现有类型，只能自己编程计算。

高精度计算通用方法：高精度计算时一般用一个数组存储一个数，数组的一个元素对应于数的一位，将数由低位到高位依次存储在数组下标对应的由低到高的位置上。另外，申请数组大小时，一般考虑了最大的情况，在很多情况下表示有富裕，即高位有很多 0，可能造成无效的运算和判断，因此一般利用一个整型数据存储该数的位数。下面的程序是一个高精度整数的加法运算，请补充完整程序。

```
#include<iostream>
using namespace std;
struct HugeInt{
    int len;
    int num[100001];
};
HugeInt a, b, w;
char c[100001],d[100001];
void Scan_HugeInt(){
    cin >> c >> d;
    a.len = strlen (c);
    b.1en = strlen (d);
    for(int i=0; i<a.len; i++)
        ___(1)____;
    for(int i=0; i<b.len; i++)
        ___(2)____;
}
void Plus(){
    w.len = max(a.len, b.len);
    for(int i=1; i<=w.len; i++){
        w.num[i] += ___(3)____;
        w.num[i+1] += ____(4)_____;
        w.num[i] %= 10;
    }
    if(____(5)___)
        w.len++;
}
int main(){
    Scan_HugeInt();
    Plus();
    for(int i=w.len;i>=1;i--)
        cout<<w.num[i];
    cout<<endl;
    return 0;
}
```

(1) 处应填（ ）

A.a.num[i]=c[i]　　B.a.num[a.len-i]=c[i]

C.a.num[i]=c[i]-'0'　　D.a.num[a.len-i]=c[i]-'0'

(2)处应填（ ）

A.b.num[i]=d[i]　　B.b.num[b.len-i]=d[i]

C.b.num[i]=d[i]-'0'　　D.b.num[b.len-i]=d[i]-'0'

(3)处应填（ ）

A.(a.num[i]+b.num[i])　　B.(a.num[i]+b.num[i])%10

C.(a.num[i]%10+b.num[i]%10)　　D.(a.num[i]+b.num[i]-10)

(4)处应填（ ）

A.w.num[i]　　B.w.num[i]%10　　C.w.num[i]/10　　D.w.num[i]-10

(5)处应填（ ）

A.w.num[w.len+1]>=0　　B.w.num[w.len+1]==0

C.w.num[w.len+1]>1　　D.w.num[w.len+1]!=0

2.（打印螺旋矩阵）

xixixi 想打印一个螺旋矩阵，比如当 n=5 时，你的程序应该输出：

```
1   2   3   4   5
16  17  18  19  6
15  24  25  20  7
14  23  22  21  8
13  12  11  10  9
```

数据规模 n<100,且**保证输入的 n 是奇数，xixixi 最终使用了递归的方法来实现！**

试补全程序。

```
#include <iostream>
using namespace std;
int p, q, n, f[105][105];
void dfs(int c, int x, int y)// 填充第 x 行，y 列的值
{
    f[x][y] = c;
    if (____(1)____) return ;
    if (p != 0) {
        if (x+p>n || x+p<1 || f[x+p][y]!=0) {
            ____(2)____;
        }
    }
    else if (q != 0)
    {
        if (y+q>n || y+q<1 || f[x][y+q]!=0) {
            ____(3)____;
        }
    }
    ______(4)______;
}
```

```
21  int main() {
22      cin >> n;
23      _____(5)_____;
24      dfs(1, 1, 1);
25      for(int i = 1; i <= n; i++) {
26          for(int j = 1; j <= n; j++)
27              printf("%5d", f[i][j]);
28          printf("\n");
29      }
30      return 0;
31  }
```

1) ①处应填（ ）

A. c==n　　B. x==n/2 && y==n/2

C. x==n/2+1 && y==n/2+1　　D. x+y==n+1

2) ②处应填（ ）

A. q = p, p = 0　　B. q = -p, p = 0

C. p = 1-p, q = p+1　　D. p = 1-q, q = p-1

3) ③处应填（ ）

A. p = q, q = 0　　B. p = -q, q = 0

C. q = -p, p = q+1　　D. q = -p+1, p = q-1

4) ④处应填（ ）

A. dfs(c+1, x, y+1)　　B. dfs(c+1, x+q, y+p)

C. dfs(c+1, x+p, y+q)　　D. dfs(c+1, x+p, y+q+1)

5) ⑤处应填（ ）

A. p=0, q=0　　B. p=1, q=0

C. p=1, q=1　　D. p=0, q=1