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 - 5

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．若一个 8 位整数的原码为(A7)H，则其对应的补码为（ ）。（H 是 16 进制数的标志）

A．(D7)H B．(A7)H C．(D9)H D．(07)H

2．在 C 语言程序中，表达式(35^18|10)的值是（ ）。

A．56 B．48 C．59 D．49

3．计算机处理信息的精度取决于（ ）。

A．CPU 的主频 B． 硬盘的容量 C． 系统总线的传输速率 D． CPU 字长

4．采用 32*32 的 01 点阵的汉字字模，存放 1600 个汉字信息的存储容量约是（ ）KB。

A．25 B．200 C．800 D．1600

5．与十进制数 1770.625 对应的八进制数是（ ）

A． 3352.5 B．3350.5 C．3352.1161 D．前 3 个答案都不对

6．下列哪个是数组具有的特点（ ）

A． 可随机访问任一元素 B． 插入不需要移动元素

C． 不必事先估计存储空间 D． 删除不需要移动元素

7．已知队列(13,2,11,34,41,77,5,7,18,26,15),第一个进入队列的元素是 13,则最后一个出队列的元素是（ ）

A．13 B．77 C．15 D．18

8．设 G 是由 5 个顶点构成的完全图，则从 G 中删去（ ）边可以得到树？

A．6 B．5 C．8 D．4

9．3 张不同的电影票全部分给 10 个人，每人至多一张，一共有多少种情况？（ ）

A． 2160 B．120 C．720 D． 240

10．要排一张有 5 个独唱和 3 个合唱的节目表，如果合唱节目不能排第一个且不能相邻，一共有多少种排法（ ）。

A． $A_5^5A_5^3$ B． $A_5^5C_3^3C_5^3$ C． $C_5^5A_3^3C_5^3$ D． $A_5^5A_3^3$

11. 某公司派 ABCDE 五人出国学习，选派条件是：(1)若 A 去，B 也去；(2)D、E 两人必有一人去；(3)如 E 去，则 A、B 也同去；　　(4)C、D 二人同去或同不去
如何选他们出国?

A. CAE 去　　B.ABE 去　　C.DEC 去　　D.BC 去

12. 已知如下代码块，则语句执行结束时 i 的值和 s 的值分别是（ ）

```
int i, s=0;
for(i=1; i<=5; i+=2)
{
    s = s+i;
}
```

A. 5 和 9　　B. 7 和 9　　C. 5 和 7　　D. 9 和 7

13. 已知 6 个结点的二叉树的先序遍历是 1 2 3 4 5 6（数字为结点的编号，以下同），后序遍历是 3 2 5 6 4 1，则该二叉树的可能的中序遍历可能是（　　）

A. 3 2 1 4 6 5　　B. 3 2 1 5 4 6
C. 3 1 2 5 4 6　　D. 2 3 1 4 6 5

14. 如一棵完全二叉树，共有 1234 个节点，其叶子结点的个数为（）

A.615　　B.616　　C.617　　D.210

15. 二维数组 A 中,每个数据元素占 4 个字节,行下标从 0 到 4,列下标从 0 到 5,按行优先存储时元素 A[3][5]的地址域和按列优先存储时元素（）的地址相同.

A. A[2][4]　　B. A[3][4]　　C. A[3][5]　D.A[4][4]

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×；除特殊说明外，判断题一题 2 分，选择题一题 2.5 分，共计 40 分）

1.

```
1  #include<cstdio>
2  bool pd(long long n)
3  {
4      if(n==1)
5          return false;
6      for(long long i=2;i<n;i++)
7          if(n%i==0)  return false;
8      return true;
9  }
10 int main()
11 {
12     long long n,i,c=0;
13     int INF=1<<30;
14     scanf("%lld",&n);
15     for(i=2;i<=INF;i++)
16     {
17         if(pd(i))
18         {
19             c++;
20             if(c==n)
21             {
22                 printf("%lld",i);
23                 return 0;
24             }
25         }
26     }
27     printf("\nover");
28     return 0;
29 }
```

约定输入 n 在[1,100]范围内。

- 判断题

  1) 上述代码中，将第6行的"i=2"修改为 "i=1"，输出结果一定不变（ ）

  2) 上述代码中，将第23行"return 0"修改为"break"或"continue"后，输入不变，输出结果也一定不变。（ ）

  3) 将第23行"return 0"修改为"break"后，程序运行中变量c的值不会受到影响（ ）

  4) 变量INF无法准确存储 1<<30 这个表达式的数值（ ）

- 选择题

  1) 当输入8时，输出为（ ）

  A.17　　B.19(回车符)over　　C.19　　D.23\nover

  2) 上述代码中，将第6行的"i<n"修改为（ ）后程序执行结果不变，效率更高

  A.i*i<=n　　B.i<n/2　　C.i<n/3　　D.i<n/4

2.

```
#include<cstdio>
int n,r,num[10000];
bool mark[10000];
void print()
{
    for(int i=1; i<=r; i++)
        printf("%d", num[i]);
    printf("\n");
}
void search(int x)
{
    for(int i=1;i<=n;i++)
        if(!mark[i])
        {
            num[x]=i;
            mark[i]=true;
            if(x==r)
                print();
            search(x+1);
            mark[i]=false;
        }
}
int main()
{
    scanf("%d%d", &n, &r);
    search(1);
    return 0;
}
```

约定输入数据 n 和 r 均为区间[1,100]的整数。

- 判断题

1) 程序结束时,对任意 1 ≤ i ≤ n, mark[i]的值为false。（ ）

2) 若 n<r ,则程序无输出。（ ）

3) 若输入为 4 3，则输出中数字1和2的个数不同。（ ）

4) 此程序的时间复杂度为O(n)。（ ）

- 选择题

5) 若输入为 6 3，则函数print()的执行次数为（ ）。

A.60　　B.120　　C.6　　D.720

6) 若输入为 7 4，则输出的最后一行为（ ）。

A. 4567　　B. 7654　　C. 4321　　D. 1234

3.

```
#include<bits/stdc++.h>
using namespace std;
int a[100][100], b[100][100];
int f(int m,int n)
{
    if(m<=0||n<=0) return 0;
    a[0][0] = b[0][0];
    for(int i=1;i<n;i++)  a[0][i]=a[0][i-1]+b[0][i];
    for(int i=1;i<m;i++)  a[i][0]=a[i-1][0]+b[i][0];
    for(int i=1;i<m;i++)
    {
        for(int j=1;j<n;j++)
        {
            a[i][j]=min(a[i-1][j],a[i][j-1])+b[i][j];
        }
    }
    return a[m-1][n-1];
}
int main()
{
    int m,n;
    cin>>m>>n;
    for(int i=0;i<m;i++)
        for(int j=0;j<n;j++)
            cin>>b[i][j];
    cout<<f(m,n);
    return 0;
}
```

约定输入数据为区间[1,1000]的正整数。

- 判断题

1) 上述代码实现了对一个长度为m*n的二维数组寻找每行上的最小值进行求和（ ）

2) 将程序中对b数组的运用均改成a数组，结果不变（ ）

3) 若输入数据为: 3 3 1 2 3 4 5 6 7 8 9 则输出的结果为29（ ）

4) 我们将上述算法称为贪心。（ ）

- 选择题(每题3分)

5) 上述代码的时间复杂度为（ ）

A. O(min(m,n))　B.O(m*n+m*n+m+n)　C.O(m*n)　D.O(m*n+m*n)

6) 若输入数据为: 4 4 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 则输出的结果为（ ）

A.28　B.16　C.136　D.46

三．完善程序（单选题，每小题 3 份，共计 30 份）

1.(字符串合并）给你两个字符串 a，b，是否能不改变 a，b 字符串原有顺序合成 c。

输入为三个待检验的字符串 a，b，c，根据题目描述判断对应输出。

若 c[i]与 c[j]源于 a 字符串的 a[x]与 a[y]，且 i<j,则需要保证 x<y.否则视为打乱了 a 串的原有顺序。

例如：

输入：lab tree latrbee

输出：Yes //解释：字符串可以被拆分

输入：lab tree laterbe

输出：No //解释:不能先使用 b 串的字符 e，再使用字符 r，顺序被打乱。

```
#include <stdio.h>
#include <string.h>
#define Max 505
char a[Max],b[Max],c[Max];
int flag=0;
int vis[Max][Max];
void dfs(int i,int j,int cnt)
{
    if( ( ① ) )//  c 已经完全匹配完了；
    {
        ( ② );
        return;
    }
    if(a[i]!=c[cnt]&&b[j]!=c[cnt])
          return ;     //  剪枝
    if( ③ )
          return ;     //      剪枝
    vis[i][j] = 1;
    if( ④ )
          dfs(i+1,j,cnt+1);
    if(flag)
          return;
    if(b[j]==c[cnt])
          ( ⑤ );
}
int main(){
    flag = 0;
    memset( vis, 0, sizeof(vis) );
    scanf("%s%s%s", a, b, c);
    dfs(0, 0, 0);
    if(flag) printf("Yes\n");
    else printf("No\n");
    return 0;
}
```

1) ①处应填（ ）

A. cnt == strlen(c)　　B. cnt == strlen(c)-1

B. i == strlen(a)　　D. j == strlen(b)

2) ②处应填（ ）

A. flag=1　　B. flag=0　　C. break　　D. return

3) ③处应填（ ）

A. vis[i][j]==1　　B. vis[i][j]==0

C. a[i] != c[cnt]　　D. b[j] != c[cnt]

4) ④处应填（ ）

A. a[i+1]==c[cnt]　　B.a[i]==c[cnt]

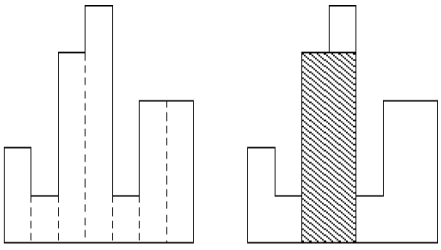
C. b[i] == c[cnt]　　D.b[i+1] == c[cnt]

5) ⑤处应填（ ）

A. dfs(i,j+1,cnt+1)　　B.dfs(i+1,j,cnt+1)

C. dfs(i,j,cnt+1)　　D.dfs(i+1,j,cnt)

2.（（最大矩形面积）给定一个矩形图，矩形图由宽度为 1 的小矩形拼成，求能够切割出来的最大矩形面积。输入 n 个数，已知从左到右每个小矩形的高度。采用分治算法，将区间从中间分成两份，即分成了两个子问题。最大矩形面积在两个子问题的最大值和由左右两部分合成的最大值中取最大值，即为最大子问题。如图所示，阴影部分即为最大值。



```
#include <stdio.h>
#define N 10005
int a[N];
int min(int a,int b)
{
    return (  ①  );
}
int max(int a,int b)
{
    if(a>b) return a;
    else return b;
}
int solve(int l,int r)
{
    if(l==r) return a[l];
    int mid= (  ②  );
```

```
17      int ans=max(solve(l,mid),solve(mid+1,r));
18      int L=mid,R=mid+1;
19      int h= (  ③  );
20      ans=max(ans,h*2);
21      while(L>l||R<r){
22          if(R<r&&(L==l||a[R+1]>a[L-1])){
23              (  ④  );
24              h=min(h,a[R]);
25          }else{
26              L--;
27              h=min(h,a[L]);
28          }
29          ans = (  ⑤  );
30      }
31      return ans;
32  }
33  int main()
34  {
35      int i,n;
36      scanf("%d",&n);
37      for(i=1;i<=n;i++)
38          scanf("%d",&a[i]);
39      printf("%d\n",solve(1,n));
40      return 0;
    }
```

1) ①处应填（ ）

A.a<b?b:a　　B.a<b?a:b　　C.a　　D.b

2) ②处应填（ ）

A.l+1+r/2　　B.(l+r+1)/2　　C.(l+r)/2　　D.(l+r-1)/2

3) ③处应填（ ）

A. min(a[L-1],a[R+1])　　B.min(a[L+1],a[R])

C. min(a[L],a[R])　　D.min(a[L],a[R+1])

4) ④处应填（ ）

A.R++　　B.R--　　C.R+1　　D.R-1

5) ⑤处应填（ ）

A. max(ans,(R-L+1)*h)　　B.max(ans,(R-L)*h)

C. max(ans,(R+L)*h)　　D.max(ans,(R+L+1)*h)