

BCSP 小学高年级组基础知识测评样卷

(C++语言 满分: 100 分 考试时间: 120 分钟)

一、 单项选择题 (每题只有一个正确选项, 每题 2 分, 共 30 分)

1. 根节点高度为 1, 一棵拥有 2024 个节点的二叉树的高度至少为 ()

A. 8 B. 10 C. 11 D. 12 E. 13 F. 14

2. 楼道里有 9 盏灯, 为了节约用电, 需关掉 3 盏互不相邻的灯, 为了行走安全, 第一盏和最后一盏不关, 则关灯方案的种数为 ()

A. 15 B. 10 C. 20 D. 24 E. 8 F. 30

3. 一个链表中, 每一个结点有一个 next 指针指向下一个结点, 现在想要删除链表中结点 p 的下一个结点, 正确的操作是?

A. `free(p->next);`

B. `temp = p->next; p->next = temp->next; free(temp);`

C. `temp = p->next; temp->next = p->next; free(temp);`

D. `temp = p->next; free(temp);`

E. `free(temp); temp = p->next;`

F. `temp = p->next; free(temp); temp->next = p->next;`

4. 对于一个含有 2024 个节点的完全二叉树, 其中有 () 个结点没有子结点()

A. 1011 B. 1013 C. 2024 D. 1012 E. 1010 F. 1024

5. 利用一个长度为 N 的一维数组, 实现循环队列。可存储的最多元素个数是()个?

A. N-1 B. N-2 C. N D. N/2 E. N/3 F. N-3

6. 一个二维数组定义为 `char a[3][10];`, 则这个二维数组占用内存的大小为 ()。

A. 3B B. 30B C. 120B D. 240B E. 0.3KB F. 0.03MB

7. 一个二叉树的后序遍历序列为 ECDAB, 中序遍历序列为 ECBDA, 则先序遍历序列为:

A. BCEDA

B. BCEAD

C. BCEAD

D. BCAED

E. ACBED

F. ABDEC

8. 下列关于栈与队列描述错误的是 ()

- A. 队列进行删除操作总是删除最早入队的元素
- B. 栈只允许在栈顶的一端进行入栈与出栈的操作
- C. 只要确定了入栈次序, 即可确定出栈次序
- D. 队列只允许在表的一端进行插入, 而在表的另外一端进行删除的线性表
- E. 栈和队列都可以使用链表或顺序表来实现
- F. 队列和栈都不支持随机访问和随机插入

9. 下列关于树的说法正确的是 ()

- A. 一棵树的子树可以存在交集
- B. 树的根节点至少有 1 个
- C. 森林是 $m (m \geq 0)$ 棵相交的树的集合
- D. 树属于非线性数据结构, 但如果只有左儿子的树属于线性数据结构
- E. 在二叉树里, 至少存在一个节点有两个儿子。
- F. 一棵树有 $n (n > 0)$ 个结点, 则树中有 $n - 1$ 条边

10. 在一棵度为 4 的树 T 中, 若有 2 个度为 4 的结点, 3 个度为 3 的结点, 4 个度为 2 的结点, 5 个度为 1 的结点, 树中的叶子结点个数为 ()

- A. 17 B. 18 C. 19 D. 16 E. 20 F. 15

11. 八进制数字 567 和二进制数字 10101 的和为 (), 提示: 选项括号里代表进制。

- A. 110011100 (2) B. 616 (8) C. 398 (10)
D. 18E (16) E. 110001100 (2) F. 618 (8)

12. a 等于 4, b 等于 -5, b 等于 -6。表达式 $a \& b | c$ 的结果是 (), 提示: $\&$ 是按位与, $|$ 是按位或

- A. 6 B. -2 C. -3 D. -4 E. 4 F. -6

13. 在 C++ 中, 执行完下面代码之后, 输出的是 ()

```
01 int dfs(int* x){  
02     return *x**x;  
03 }  
04 int main(){  
05     int a = 10;
```

```

06    int * p = &a;
07    *p=dfs(&a);
08    cout<<*p<<' '<<a<<endl;
09    return 0;
10 }

```

A. 100 100 B. 10 100 C. 10 100 D. 10 10 E. 0 10 F. 10 0

14. 下列哪个是合法的 C++ 变量名 ()。

A. friend B. sizeof C. template D. #ab E. _ab F. namespace

15. 假设你有两个一模一样的水杯，一栋 100 层的高楼扔下来。从某一层楼以下，水杯会摔碎，而超过这一层楼，水杯会保持完好。你最少需要扔几次水杯才能确定水杯摔碎的临界楼层？提示：你只有两个水杯，水杯没摔碎还可以继续用，不会改变临界楼层。

A. 6 B. 10 C. 99 D. 100 E. 14 F. 7

二、程序阅读理解题（共 3 大题。程序输入不超过数组或字符串定义的范围，除特殊说明外，判断题 1.5 分，选择题 3 分，共计 40 分）

(1) (14.5 分)

```

01    #include<bits/stdc++.h>
02    using namespace std;
03    int c[1005][1005];
04    int g(int x,int y)
05    {
06        if(x<y) return 0;
07        if(y == 0 || y == x) return 1;
08        if(y == 1 || y == x - 1) return x;
09        return g(x-1,y-1)+g(x-1,y);
10    }
11    int main()
12    {
13        int x,y;
14        cin>>x>>y;
15        cout<<g(x,y)<<endl;
16        memset(c,0,sizeof(c));
17        for(int i=0;i<=x;i++){
18            c[i][0] = c[i][i] = 1;
19            for(int j = 1; j < i; j++)
20                c[i][j] = (c[i-1][j] + c[i-1][j-1]);

```

```

21      }
22      cout<<c[x][y]<<endl;
23  }

```

题目保证输入均为 1000 以内的正整数

● 判断题

16. 此题时间复杂度 $O(x * y)$ 。
17. 程序的输出结果有两行。
18. 程序输出的第一行和第二行永远相等。
19. 第 7 行代码改成 \leq ，程序的结果不变。
20. 当输入的 $x \geq y$ ，第 16 行代码改为 `memset(c,1,sizeof(c));`，输出结果不变。

● 选择题

21. 当输入的 x, y 为 6, 4 时，输出第二行的结果是 ()
 A. 5 B. 20 C. 25 D. 30 E.35 F.15
22. 当输入的 x, y 为 1000,3 时，输出第一行的结果是 () (4 分)
 A. 166167000 B. 499500 C. 1000 D. 3 E.161700 D. 0

(2) (12 分)

```

01.  #include<iostream>
02.  #include<cstring>
03.  using namespace std;
04.  int a[2010],b[2010],c[2018],i,y,x,l1,l2,z;
05.  char n[2001],m[2001];
06.  void change(string m,int* b,int k){
07.  for(int i=0; i < k; i++)
08.      if(z > 10 && m[i] >= 'A')
09.          b[k - i] = m[i] - 'A' + 10;
10.      else b[k - i] = m[i] - '0';
11.  }
12.  int main(){
13.  cin>>z;
14.  cin>>n>>m;
15.  l1=strlen(n); l2=strlen(m);
16.  change(n,a,l1);
17.  change(m,b,l2);
18.  while (x <= l1 || x <= l2){

```

```

19.      x++;
20.      c[x] = y + a[x] + b[x];
21.      y=c[x]/z;
22.      c[x]%=z;
23.  }
24.  while (c[x] == 0 && x > 1) x--;
25.  for (int i = x; i>=1; i--){
26.      if(c[i] < 10)
27.          cout<<c[i];
28.      else
29.          cout<<(char)(c[i]+'A'-10);
30.  }
31.  return 0;
32.}

```

输入保证 z 小于等于 36，大于等于 2。 n, m 字符串长度小于 2000，仅包含字符 0 到 9 和大写字母。

● 判断题

23. 不存在任何输入样例，在删除 24 行前后，输出相同。
24. 存在一组输入样例，在改变 20 行为 $c[x] = [x] + b[x]$ 前后，输出相同。
25. 输出的字符串长度一定会大于 11 或 12。
26. 输出的结果可能存在小写字母。

● 选择题

27. 如果输入为 10 5 6，则运行结果为

- A. 1 B. 30 C. -1 D. 0 E. 11 F. 3

28. 如果输入为 16 AF2 1AB，则运行结果为

- A. 9D B. CAD C. C9D D. C8A E. C8B F. C8C

(3) (13.5 分)

```

01  #include <bits/stdc++.h>
02      using namespace std;
03      vector<int> ans;
04      int n, g[65][65], d[65];
05      char str[5];
06      char tochar(int x){

```

```

07         if(x >= 1 && x <= 26) return x - 1 + 'A';
08         else return x - 27 + 'a';
09     }
10     int getnum(char ch){
11         if(ch >= 'A' && ch <= 'Z') return ch - 'A' + 1;
12         else return 27 + ch - 'a';
13     }
14     void dfs(int now){
15         for(int i=1;i<=52;++i){
16             if(g[now][i]){
17                 g[now][i] = g[i][now] = 0;
18                 dfs(i);
19             }
20         }
21         ans.push_back(now);
22     }
23     int main()
24     {
25         int cnt = 0, u1 = 1e9 + 7, u2 = 1e9 + 7;
26         cin >> n;
27         for(int i=1;i<=n;++i){
28             cin >> str;
29             int u = getnum(str[0]), v = getnum(str[1]);
30             g[u][v] = g[v][u] = 1;
31             d[u]++;
32             d[v]++;
33         }
34         for(int i=1;i<=52;++i){
35             if(d[i] & 1){
36                 cnt++;
37                 u1 = min(u1, i);
38             } else {
39                 if(d[i] != 0) u2 = min(u2, i);
40             }
41         }
42         if(cnt != 0 && cnt != 2){
43             cout << "No Solution" << endl;
44             return 0;
45         }
46         if(!cnt) dfs(u2);
47         else dfs(u1);
48
49         reverse(ans.begin(), ans.end());
50

```

```

51         for(int i=0;i<ans.size();++i) cout <<
           tochar(ans[i]);
52         return 0;
53     }

```

(约定每次输入的 str 均由 2 个大写或小写字母拼接而成, 如"aa", "Xt", "RE")

● 判断题

29.若 n=1, 则输出结果和输入的 str 相同。

30.若每次输入的 str 均为"aa", 则输出结果为 n+1 个'a' 拼接而成的字符串。

31.若 n>1, 且每次输入的 str 共 2n 个字符两两不同, 则输出"No Solution".

32.若 n=26, 且输入的 str 依次为"ab", "bc", "cd", ..., "za", 则输出结果为"abcd...za".

(2 分)

● 选择题

33.在所有 str 中, 出现奇数次的字母个数为 () 时, 程序不会输出 "No Solution".

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

34.当输入为"9 ab cd hh od ob bb ba bh ha" 时, 输出字符串的长度为 () (4 分)

A. 10 B. 9 C. 8 D. 7 E. 6 F. 5

三、程序完善题 (共 2 大题, 每个选择题 3 分, 共计 30 分)

1.给出一棵二叉树的中序与后序排列。求出它的先序排列。(约定树结点用不同的大写字母表示)。

举例来说, 中序遍历为 BADC, 后序遍历为 BDCA, 先序遍历结果为 ABCD

```

01     #include <iostream>
02     using namespace std;
03     string ldr_str, lrd_str;
04     void build(int l1, int r1, int l2, int r2)
05     {
06         if(l1 > r1) return;
07         cout << ____2____;
08         int p = l1;
09         while(ldr_str[p] ____3____ lrd_str[r2]) p++;
10         int cnt = p-l1;
11         build(l1, p-1, l2, ____4____);
12         build(p+1, r1, l2+cnt, r2-1);
13     }
14     int main(){
15         cin >> ldr_str >> lrd_str;

```

```

16     int n = ____1____;
17     build(0, n, 0, n);
18 }

```

35. 1 处应填

- | | |
|---------------------|-------------------------|
| A. ldr_str.size()-1 | B. strlen(ldr_str) - 1 |
| C. ldr_str.size() | D. strlen(ldr_str) - 1 |
| E. ldr_str.size()*2 | F. ldr_str.size()*2 - 1 |

36. 2 处应填

- | | |
|------------------|--------------------|
| A. lrd_str[r2] | B. lrd_str[r2] - 1 |
| C. ldr_str[r2] | D. ldr_str[r2] - 1 |
| E. lrd_str[r2]+1 | F. ldr_str[r2] + 1 |

37. 3 处应填

- | | |
|-------|-------|
| A. > | B. < |
| C. >= | D. <= |
| E. != | F. == |

38. 4 处应填

- | | |
|-------------|-------------|
| A. l2 | B. l1 |
| C. l2+cnt | D. l1+cnt-1 |
| E. l2+cnt-1 | F. l1+cnt-1 |

39. 5 处应填

- | | |
|-------------|-------------|
| A. l2+cnt-1 | B. l1+cnt-1 |
| C. l2+cnt | D. l1+cnt-1 |
| E. l1 | F. l2 |

2. (逆序对)对于给定的一段正整数序列，逆序对就是序列中 $a_i > a_j$, 且 $i < j$ 的有序对。

举例来说，序列 5 1 2 3 4，其中逆序对有四个，分别是(5,1), (5,2), (5,3), (5,4)
 输入第一行包含一个正整数 n ，接下来一行包含 n 个正整数，表示序列的内容。
 序列中每个数字不超过 int 范围, n 最大 100000
 提示：使用归并排序来求解逆序对

```

01     #include<cstdio>
02     #include<cstring>
03     #include<algorithm>
04     #define N 500005
05     using namespace std;
06     int a[N],b[N],c[N];
07     long long merge(int l,int r,int mid)

```



```

08  {
09      int i,b1=0,c1=0,b2=1,c2=1;
10      for(i=1;i<=mid;++i) b[++b1]=a[i];
11      for(i=mid+1;i<=r;++i) c[++c1]=a[i];
12      long long ans=____2____;
13      for(i=1;i<=r;++i)
14          {
15              if(b1>=b2&&(c1+1==c2||b[b2]<=c[c2]))
16                  ____4____;
17              else
18                  {
19                      a[i]=c[c2++];
20                      ans+=____5____;
21                  }
22          }
23      return ans;
24  }
25  long long solve(int l,int r)
26  {
27      if(l==r) return 0;
28      int mid=(l+r)>>1;
29      long long ans=0;
30      ans+=____3____;
31      ans+=solve(mid+1,r);
32      ans+=____1____;
33      return ans;
34  }
35  int main()
36  {
37      int n,i;
38      long long ans;
39      scanf("%d",&n);
40      for(i=1;i<=n;++i)scanf("%d",&a[i]);
41      ans=solve(1,n);
42      printf("%lld",ans);
43      return 0;
44  }

```

40. 1 处应填

- | | |
|------------------------|------------------------|
| A. merge(l-1,r-1,mid); | B. merge(l-1,r+1,mid); |
| C. merge(l,r,mid); | D. merge(l,r,mid-1); |
| E. merge(l+1,r-1,mid); | F. merge(l+1,r+1,mid); |

41. 2 处应填

- A. $r-1$
- C. $c1-b1$;
- E. 0 ;

- B. $1-r$
- D. $b1-c1$;
- F. 1 ;

42. 3 处应填

- A. `solve(l, mid - 1);`
- B. `solve(l, mid + 1);`
- C. `solve(l + 1, mid);`
- D. `solve(l + 1, mid-1);`
- E. `solve(l, mid);`
- F. `solve(l, mid - 1);`

43. 4 处应填

- A. `a[i]=b[b2++]`
- C. `a[i]=b[b1++]`
- E. `a[i]=b[c1++]`
- B. `a[i]=b[b2--]`
- D. `a[i]=b[b1--]`
- F. `a[i]=b[c2++]`

44. 5 处应填

- A. $b1-b2+1$
- C. $b1-b2$
- E. $b1-b2-1$
- B. $b2-b1+1$
- D. $b2-b1$
- F. $b2-b1-1$