

2024 北京市中小学信息学能力测评活动第一轮

基础知识测评（小学高年级组）

一、单项选择题（共 15 题，每题 2 分，共计 30 分）

1. 计算机在工作过程中突然停电，（ ）中的信息不会丢失。
A. 显存
B. 寄存器
C. RAM
D. ROM
2. 中缀表达式 $a*(b+c)-d$ 的后缀形式是（ ）。
A. $abcd*+-$
B. $abc+*d-$
C. $abc*+d-$
D. $-+*abcd$
3. 设栈 S 的初始状态为空，元素 a, b, c, d, e 依次入栈，以下出栈序列不可能出现的有（ ）。
A. a, b, c, e, d
B. b, c, a, e, d
C. a, e, c, b, d
D. d, c, e, b, a
4. 已知 7 个结点的二叉树的先根遍历是 1 2 4 5 6 3 7（数字为结点编号），中根遍历是 4 2 6 5 1 7 3，则该二叉树的后根遍历是（ ）。
A. 4 6 5 2 7 3 1
B. 4 6 5 2 1 3 7
C. 4 2 3 1 5 4 7
D. 4 6 5 3 1 7 2
5. 在 C++ 中，若变量 x 为 int 类型且已被赋值为 40，则 $x \& (x - 1)$ 的值为（ ）。
A. 79
B. 47
C. 32
D. 0
6. 有一个等比数列，共有奇数项，其中第一项和最后一项分别是 2 和 118098，中间一项是 486，请问以下那个数是可能的公比（ ）。
A. 2
B. 3

- C. 4
D. 5
7. 设变量 x 为 `float` 类型且已赋值, 则以下语句能将 x 中的数值四舍五入到小数点后第 2 位的是 ()。
- A. $x = (x * 100 + 0.5) / 100.0$;
B. $x = x * 100 + 0.5 / 100.0$;
C. $x = (x / 100 + 0.5) * 100.0$;
D. $x = (\text{int})(x * 100 + 0.5) / 100.0$;
8. 十六进制下, $7*7$ 的运算结果为 ()。
- A. 31
B. 3B
C. 41
D. 4B
9. () 是一种选优搜索法, 按选优条件向前搜索, 以达到目标。当搜索到某一步时, 发现原先选择并不优或达不到目标, 就退回一步重新选择。
- A. 回溯法
B. 枚举法
C. 动态规划
D. 贪心
10. 1TB 代表的字节数量是 ()。
- A. 2 的 10 次方
B. 2 的 20 次方
C. 2 的 30 次方
D. 2 的 40 次方
11. 原字符串中任意一段连续的字符所组成的新字符串称为子串。则字符 “AAABBBCCC” 共有 () 个不同的非空子串。
- A. 3
B. 12
C. 36
D. 45
12. 1958 年以前的第一代计算机主要用于科学计算、军事研究。这些计算机以 () 为主要的逻辑元件。
- A. 晶体管
B. 电子管
C. 集成电路
D. 大规模集成电路
13. 链表不具备的特点是 ()。
- A. 可用 $O(1)$ 时间随机访问任何一个元素。

- B. 插入、删除操作不需要移动元素。
 - C. 存储单元在内存中的地址可以不连续。
 - D. 无需事先估计存储空间大小。
14. 以下排序算法中，() 属于稳定排序算法。
- A. 堆排序
 - B. 选择排序
 - C. 冒泡排序
 - D. 快速排序
15. 完全二叉树的顺序存储方案，是指将完全二叉树的结点从上到下、从左到右依次存放到一个顺序结构的数组中。假定根结点存放在数组的 1 号位置上，则第 k 号结点的父结点如果存在的话，应当存放在数组中的 () 号位置。
- A. $2k$
 - B. $2k + 1$
 - C. $\lfloor k / 2 \rfloor$
 - D. $\lceil k / 2 \rceil$

二、阅读程序（判断题正确填 A，错误填 B，选择题每题有且仅有一个正确选项；除特殊说明外，判断题 1.5 分，选择题 3 分，共计 40 分）

(1)

```

001 #include <iostream>
002 using namespace std;
003 int main() {
004     int n, a[105], avg = 0, ans = 0;
005     cin >> n;
006     for (int i = 1; i <= n; i++) {
007         cin >> a[i];
008         avg += a[i];
009     }
010     avg /= n;
011     for (int i = 1; i < n; i++)
012         if (a[i] < avg) {
013             a[i + 1] -= (avg - a[i]);
014             ans += (avg - a[i]);
015         }
016         else if (a[i] > avg) {
017             a[i + 1] += (a[i] - avg);
018             ans += (a[i] - avg);
019         }
020     cout << ans << endl;

```

```
021     return 0;
022 }
```

假设输入的所有数是不超过 100 的正整数，完成下面的判断题和单选题：

● 判断题

16. 该程序的算法时间复杂度为 $O(n)$ 。()
17. 若将第 004 行的代码改为 “int n, a[105], avg, ans;” (声明变量时不赋值)，程序仍可以被正常执行，且对于相同的输入数据，输出结果不变。()
18. 若将第 004 行的代码改为 “int n, a[100], avg = 0, ans = 0;” (更改数组 a 的大小)，程序仍可以被正常执行，且对于相同的输入数据，输出结果不变。()
19. 若将第 011 行的 for 循环执行条件改为 “i <= n”，程序仍可以被正常执行，且对于相同的输入数据，输出结果不变。()
20. 若将第 016 行的代码改为 “else {” (去掉 else 之后的条件判断)，程序仍可以被正常执行，且对于相同的输入数据，输出结果不变。()
21. 该程序的输出结果不可能为负数。()

● 单选题

22. 若输入数据第一行为 “7”，第二行为 “1 9 2 8 12 2 8”，则程序将输出 ()。
- A. 2
- B. 16
- C. 21
- D. 26

(2)

```
001 #include <iostream>
002 using namespace std;
003 int n, h[1000005], L, R;
004 long long m;
005 int main() {
006     cin >> n >> m;
007     for (int i = 1; i <= n; i++) {
008         cin >> h[i];
009         if (h[i] > R) R = h[i];
010     }
011     while (L < R) {
012         int mid = (L + R + 1) / 2;
013         long long tmp = 0;
014         for (int i = 1; i <= n; i++)
015             if (h[i] > mid)
```

```

016             tmp += h[i] - mid;
017         if (tmp < m)
018             R = mid - 1;
019         else
020             L = mid;
021     }
022     cout << L << endl;
023     return 0;
024 }

```

假设输入的所有数是正整数，其中 n 以及数组元素 $h[1], h[2], \dots, h[n]$ 均不超过 1000000， m 不超过 $h[1], h[2], \dots, h[n]$ 之和，完成下面的判断题和单选题：

● 判断题

23. 若将第 011 行的 while 循环执行条件改为 “ $L \leq R$ ”，程序仍可以被正常执行，且对于相同的输入数据，输出结果不变。（ ）
24. 若将变量 m 和变量 tmp 的数据类型都改为 int ，程序仍可以被正常执行，且对于相同的输入数据，输出结果不变。（ ）
25. 程序的输出结果有可能是 0。（ ）

● 单选题

26. （本题 4 分）将第 012 行代码改为（ ），程序执行的效果不变。
 - A. $int\ mid = (L + R) / 2;$
 - B. $int\ mid = (L + R) * 2;$
 - C. $int\ mid = L + R + 1 << 2;$
 - D. $int\ mid = L + R + 1 >> 2;$
27. 若输入数据第一行为 “5 20”，第二行为 “4 42 40 26 46”，则程序将输出（ ）。
 - A. 35
 - B. 36
 - C. 37
 - D. 38
28. 若输入的 n 为 10000，程序输出结果的最大可能值是（ ）。
 - A. 9999
 - B. 10000
 - C. 999999
 - D. 1000000

(3)

```

001 #include <stdio>
002 #include <cstring>
003 using namespace std;
004 const int MAXN = 5000 + 5;
005 int n, a[MAXN], d[MAXN];
006 int check(int k) {
007     memset(d, 0, sizeof(d));
008     int res = 0;
009     for (int i = 1, s = 0; i <= n; i++) {
010         s += d[i];
011         if ((a[i] + s) % 2 == 1)
012             continue;
013         if (i + k > n + 1)
014             return 1e9;
015         d[i]++, s++, res++;
016         d[i + k]--;
017     }
018     return res;
019 }
020 int main() {
021     scanf("%d", &n);
022     for (int i = 1; i <= n; i++)
023         scanf("%d", &a[i]);
024     int ans = 1e9;
025     for (int k = 1; k <= n; k++) {
026         int tmp = check(k);
027         if (tmp < ans)
028             ans = tmp;
029     }
030     printf("%d\n", ans);
031     return 0;
032 }

```

假设输入的 n 是不超过 5000 的正整数，数组元素 $a[1]$, $a[2]$, \dots , $a[n]$ 均是不超过 1 的非负整数，完成下面的判断题和单选题：

● 判断题

29. 若将第 009 行的 for 循环执行条件改为 “ $i + k - 1 \leq n$ ”，程序仍可以被正常执行，且对于相同的输入数据，输出结果不变。（ ）
30. 若将第 015 行的代码改为 “ $s++$, $res++$;”（去掉 $d[i]++$ ），程序仍可以被正常执行，且对于相同的输入数据，输出结果不变。（ ）
31. 若将第 015 行中的 “ $s++$ ” 改为 “ $s--$ ”，程序仍可以被正常执行，且对于相同的输入数据，输出结果不变。（ ）

32. 程序的输出结果有可能为 0。()
33. 程序的输出结果必然小于输入的 n 。()

● 单选题

34. 该程序的算法时间复杂度为 ()。
- A. $O(n)$
- B. $O(n \log n)$
- C. $O(n^2)$
- D. $O(n^2 \log n)$
35. 若输入数据第一行为“7”，第二行为“0 0 1 0 1 0 0”，则程序将输出 ()。
- A. 3
- B. 4
- C. 5
- D. 6

三、完善程序（单选题，每小题 3 分，共计 30 分）

(1) 给定长度 n 的整数序列 a_1, a_2, \dots, a_n ，以及 q 个询问。每个询问将指定两个整数 l, r ，请判断下标在 l 与 r 之间的序列元素（包括 a_l 与 a_r ）是否互不相同。输入数据保证 $1 \leq n, q \leq 100000, 1 \leq a_i \leq n, 1 \leq l \leq r \leq n$ 。

试用最优选项补全程序：

```
001 #include <iostream>
002 #include <algorithm>
003 using namespace std;
004 const int MAXN = 1e5 + 5;
005 int n, q, a[MAXN], last[MAXN], tmp[MAXN], mxlast[MAXN];
006 int main() {
007     cin >> n >> q;
008     for (int i = 1; i <= n; i++) {
009         cin >> a[i];
010         ①;
011         ②;
012         ③;
013         if (last[i] > mxlast[i])
014             mxlast[i] = last[i];
015     }
016     while (④) {
017         int l, r;
018         cin >> l >> r;
019         if (⑤)
```

```

020             cout << "No" << endl;      // 区间内存在相同元素
021         else
022             cout << "YES" << endl;      // 互不相同
023     }
024     return 0;
025 }

```

36. ①处应填 ()。

- A. last[tmp[i]] = a[i]
- B. last[i] = tmp[a[i]]
- C. tmp[last[i]] = i
- D. tmp[a[i]] = i

37. ②处应填 ()。

- A. last[tmp[i]] = a[i]
- B. last[i] = tmp[a[i]]
- C. tmp[last[i]] = i
- D. tmp[a[i]] = i

38. ③处应填 ()。

- A. mxlast[i] = MAXN
- B. mxlast[i] = tmp[i]
- C. mxlast[i] = 0
- D. mxlast[i] = mxlast[i - 1]

39. ④处应填 ()。

- A. q--
- B. --q
- C. !(q--)
- D. !(--q)

40. ⑤处应填 ()。

- A. mxlast[l] <= r
- B. mxlast[r] <= l
- C. mxlast[l] >= r
- D. mxlast[r] >= l

(2) 给定正整数 n ，请编写程序计算它的全部正约数之和。输入数据保证 $n \leq 10000000$ 。

试用最优选项补全程序：


```

001 #include <iostream>
002 #include <algorithm>
003 using namespace std;
004
005 const int MAXN = 10000005;
006 int n, d[MAXN];
007 bool flag[MAXN];
008
009 long long value(int x, int k) {
010     long long ans = 1;
011     for (int i = 1; i <= k + 1; i++)
012         ans *= x;
013     return ①;
014 }
015
016 int main() {
017     for (②) {
018         if (flag[i]) continue;
019         for (int j = i; j < MAXN; j += i) {
020             if (flag[j]) continue;
021             flag[j] = true;
022             ③;
023         }
024     }
025
026     cin >> n;
027     long long ans = 1;
028     int last = -1, t = 0;
029     for (; n > 1; ④) {
030         if (d[n] == last) t++;
031         else {
032             ans *= value(last, t);
033             ⑤;
034         }
035     }
036     ans *= value(last, t);
037     cout << ans << endl;
038     return 0;
039 }

```

41. ①处应填 ()。

- A. $\text{ans} / (x - 1)$
- B. $\text{ans} * (x - 1)$

C. $(ans - 1) * (x - 1)$

D. $(ans - 1) / (x - 1)$

42. ②处应填 ()。

A. `int i = 1; i < MAXN; i++`

B. `int i = 1; i <= MAXN; i++`

C. `int i = 2; i < MAXN; i++`

D. `int i = 2; i <= MAXN; i++`

43. ③处应填 ()。

A. `d[j] = i`

B. `d[i] = j`

C. `d[i] = i`

D. `d[j] = j`

44. ④处应填 ()。

A. `n -= d[n]`

B. `n -= last`

C. `n /= d[n]`

D. `n--`

45. ⑤处应填 ()。

A. `last = d[n], t = 0`

B. `last = d[n], t = 1`

C. `last = d[n], t--`

D. `last = -1, t = 0`

-----题目结束-----