

2025信息素养初中组复赛真题 — 题目解析

选择题

第 1 题

[题目] C++ 表达式 $(5 > 3) * 3$ 的值是？

[答案] D (3)

[解析]

- 先计算 $5 > 3$ ，结果为真。
- 在 C++ 中，`true` 参与整数运算时可看作 1。
- 所以原式变为 $1 * 3 = 3$ 。
- 各选项对应结果：
 - A (0)：错误， $5 > 3$ 不会得到 0。
 - B (true)：错误，虽然 $5 > 3$ 是真，但题目求的是整个表达式 $(5 > 3) * 3$ 的值。
 - C (1)：错误， $5 > 3$ 的确可看作 1，但还要继续乘以 3。
 - D (3)：正确， $1 * 3 = 3$ 。

[知识点]

- 关系表达式；
- `bool` 与整数的转换。

第 2 题

[题目] 以下说法错误的是？

```
int a, b;  
cin >> a >> b;  
a = a * b;
```

```
b += a;  
cout << a << " " << b << endl;
```

[答案] B

[解析]

- 输入 2 3 时：
 - $a = 2 * 3 = 6$
 - $b = 3 + 6 = 9$
 - 输出 6 9，A 正确。
- 输入 1 1 时：
 - $a = 1$
 - $b = 1 + 1 = 2$
 - 输出应为 1 2，不是 2 2，所以 B 错误。
- $a = a * b$ 与 $a *= b$ 等价，C 正确。
- $b += a$ 与 $b = b + a$ 等价，D 正确。
- 各选项分析：
 - A：正确。输入 2 3 时，程序输出 6 9。
 - B：错误。输入 1 1 时，程序输出应是 1 2，不是 2 2。
 - C：正确。 $a = a * b$ 与 $a *= b$ 效果一样。
 - D：正确。 $b += a$ 与 $b = b + a$ 效果一样。

[知识点]

- 复合赋值；
- 代入法验证程序输出。

第 3 题

[题目] 想输出 'a' ~ 'z' 26 个字母，横线处可以填写 ()。

```
for (int i = 1; i <= 26; i++)  
{  
    cout << _____ << endl;  
}
```

[答案] B (`char('a' - 1 + i)`)

[解析]

- 当 `i = 1` 时，应输出 `'a'`。
- 当 `i = 26` 时，应输出 `'z'`。
- `'a' - 1 + i` 恰好对应 `'a'` 到 `'z'` 的字符编码。
- 再用 `char(...)` 转回字符输出即可。
- 各选项分析：
 - A：错误。`char('a' + i)` 在 `i = 1` 时输出的是 `'b'`。
 - B：正确。`i = 1` 时得到 `'a'`，`i = 26` 时得到 `'z'`。
 - C：不合适。虽然 `'a' - 1 + i` 是字符编码值，但题目要求直接输出字母，标准写法应转成 `char(...)`。
 - D：错误。`'a' + i` 同样会从 `'b'` 开始。

[知识点]

- 字符编码；
- 强制类型转换。

第 4 题

[题目] 以下说法错误的是 ()。

```
int n;
cin >> n;
while (n > 0)
{
    cout << n % 10;
    n /= 10;
}
```

[答案] C

[解析]

- 程序每次输出个位，再删去个位，所以会把数字倒着输出。
- A：输入 345，输出 543，正确。

- B: 输入 100，依次输出 0、0、1，即 001，正确。
- C: 说“任意非负整数”都可以，但输入 0 时循环不会执行，不能输出 0，所以错误。
- D: 若删去 $n \neq 10$ ，则 n 不会变化，只要初值大于 0 就会死循环，正确。
- 各选项分析：
 - A: 正确。输入 345 时输出 543。
 - B: 正确。输入 100 时输出 001。
 - C: 错误。输入 0 时，循环一次都不执行，因此不能说“任意非负整数”都适用。
 - D: 正确。去掉 $n \neq 10$ 后， n 不变，会死循环。

[知识点]

- 数位分离；
- 特殊值 0 的判断。

第 5 题

[题目] 运行程序并输入 9，输出为（）。

```
#include <iostream>
using namespace std;
int a[15];
int main()
{
    int n;
    cin >> n;
    a[1] = a[2] = 1;
    for (int i = 3; i <= n; i++)
    {
        a[i] = a[i - 1] + a[i - 2];
    }
    cout << a[n];
    return 0;
}
```

[答案] C (34)

[解析]

- 这是斐波那契数列递推：
 - $a[1]=1, a[2]=1$
 - $a[i]=a[i-1]+a[i-2]$
- 依次得到：1, 1, 2, 3, 5, 8, 13, 21, 34
- 所以输入 9 时输出 34。
- 各选项分析：
 - A (13)：错误，这是 $a[7]$ 的值。
 - B (21)：错误，这是 $a[8]$ 的值。
 - C (34)：正确，这是 $a[9]$ 的值。
 - D (55)：错误，这是 $a[10]$ 的值。

[知识点]

- 数组递推；
- 斐波那契数列。

编程题

第 8 题

[题目概括] 地宫共有 n 个区域，第 i 个区域有宝藏 $a[i]$ ，机关伤害值为 $b[i]$ 。

- 若 $b[i] = 0$ ，说明该区域无机关，可以获得该区域宝藏；
- 若 $b[i] > 0$ ，说明该区域有机关，不能获得该区域宝藏，并且会受到 $b[i]$ 点伤害。

探险队按 $1 \sim n$ 的顺序依次经过所有区域。

- 若总伤害 **超过** 护盾防御值 k ，输出 Danger；
- 否则输出最终获得的宝藏总数。

[样例 1]

输入
5 6

4 1 9 4 2
0 3 0 1 2

输出

13

[解析]

- 区域 1: $b[1]=0$, 获得 4
- 区域 2: $b[2]=3$, 不能拿宝藏, 伤害累计 3
- 区域 3: $b[3]=0$, 获得 9
- 区域 4: $b[4]=1$, 不能拿宝藏, 伤害累计 4
- 区域 5: $b[5]=2$, 不能拿宝藏, 伤害累计 6
- 总伤害 6 没有超过 $k=6$, 所以安全。
- 宝藏总数为 $4 + 9 = 13$ 。

[思路] 顺序扫描每个区域:

- 如果 $b[i] == 0$, 把 $a[i]$ 加入答案;
- 如果 $b[i] > 0$, 把 $b[i]$ 加到总伤害里;
- 最后判断总伤害是否大于 k 。

[参考程序]

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    int n, k;
    cin >> n >> k;

    vector<int> a(n + 1), b(n + 1);
    for (int i = 1; i <= n; i++) cin >> a[i];
    for (int i = 1; i <= n; i++) cin >> b[i];

    int damage = 0;
    int treasure = 0;

    for (int i = 1; i <= n; i++) {
        if (b[i] == 0) treasure += a[i];
        else damage += b[i];
    }
}
```

```
    if (damage > k) cout << "Danger";  
    else cout << treasure;  
  
    return 0;  
}
```

[知识点]

- 顺序模拟；
- 条件判断；
- 累加统计。

第 9 题

[题目概括] 输入 n 个只含小写字母的字符串：

- 统计其中回文串的个数；
- 按输入顺序把所有回文串连接起来输出；
- 如果一个回文串都没有，第二行输出空行。

[样例]

输入

5

level

world

kayak

hello

civic

输出

3

levelkaykcivic

[解析]

- level 是回文串；
- world 不是；
- kayak 是；

- hello 不是；
- civic 是。
- 所以共有 3 个，连接结果是 levelkayakcivic。

[思路一：反转比较] 对于每个字符串 s：

- 复制一份 t = s；
- 将 t 反转；
- 若 t == s，则 s 是回文串。

[参考程序]

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    int n;
    cin >> n;

    int cnt = 0;
    string ans = "";

    for (int i = 1; i <= n; i++) {
        string s, t;
        cin >> s;
        t = s;
        reverse(t.begin(), t.end());
        if (s == t) {
            cnt++;
            ans += s;
        }
    }

    cout << cnt << '\n' << ans;
    return 0;
}
```

[知识点]

- 字符串处理；
- 回文串判断；
- reverse 的使用。

第 10 题

[题目概括] 有 n 堆宝石，第 i 堆有 $a[i]$ 颗。从某一堆开始拿，之后每隔若干堆再拿一堆。根据你提供的样例和题面截图对应程序，实际实现规则是：

- 若当前拿第 x 堆，下一次拿第 $x + k + 1$ 堆。

也就是说，选定起点后，拿取序列为：

- x
- $x + k + 1$
- $x + 2(k + 1)$
- ...

[样例]

输入

7 2

4 6 2 1 10 1 3

输出

16

[解析]

- 从第 1 堆开始：拿 1,4,7 ，总数 $4+1+3=8$
- 从第 2 堆开始：拿 2,5 ，总数 $6+10=16$
- 从第 3 堆开始：拿 3,6 ，总数 $2+1=3$
- 从第 4 堆开始：拿 4,7 ，总数 $1+3=4$
- 从第 5 堆开始：拿 5 ，总数 10
- 从第 6 堆开始：拿 6 ，总数 1
- 从第 7 堆开始：拿 7 ，总数 3
- 最大值为 16 。

[思路] 步长固定为 $k + 1$ 。

从某个起点 $start$ 出发，能拿到的所有堆下标是一个等差数列：

- start
- start + k + 1
- start + 2(k + 1)
- ...

只需枚举前 k + 1 个起点：

- 1, 2, 3, ..., k+1

分别求和，取最大值即可。

[参考程序]

```
#include <bits/stdc++.h>
using namespace std;
#define LL long long

LL a[1000010];
int n, k;

LL getSum(int start) {
    LL sum = 0;
    while (start <= n) {
        sum += a[start];
        start += k + 1;
    }
    return sum;
}

int main() {
    cin >> n >> k;
    for (int i = 1; i <= n; i++) cin >> a[i];

    LL ans = 0;
    for (int start = 1; start <= min(n, k + 1); start++) {
        ans = max(ans, getSum(start));
    }

    cout << ans;
    return 0;
}
```

[知识点]

- 枚举起点；
- 等差下标求和；
- 余数分组思想。

结论汇总

选择题核对结果

- 第 1 题答案：D
- 第 2 题答案：B
- 第 3 题答案：B
- 第 4 题答案：C
- 第 5 题答案：C

选择题最终答案为：D、B、B、C、C。