

苏州市人工智能学会
青少年人工智能核心算法素养考核（SACCC）

2 级参考答案

一、选择题，共 20 题，每题 2 分，共 40 分

1. 人工智能领域经常提到的“大模型”，最贴切的理解是（ ）

- A. 大型计算机硬件模型
- B. 大规模集成电路
- C. 大语言模型（Large Language Model）
- D. 大型数据库模型

答案：C

解析：在 AI 领域，“大模型”特指“大语言模型”（LLM），如 ChatGPT、文心一言等，是基于海量数据训练的大规模人工智能模型。

2. 计算机存储单位中，1 GB 等于（ ）

- A. 1000 MB
- B. 1024 MB
- C. 1000 KB
- D. 1024 KB

答案：B

解析：1 KB = 1024 B, 1 MB = 1024 KB, 1 GB = 1024 MB, 1 TB = 1024 GB。

3. 下列选项中，不属于 C++ 基本内置数据类型的是 ()

A. int B. double C. string D. char

答案: C

解析: 基础数据类型区分, string 为字符串类非基础内置类型

4. C++ 中, 整型变量 int 占用的内存字节数为 ()

A. 1 字节 B. 2 字节 C. 4 字节 D. 8 字节

答案: C

解析: GESP 标准 int 字节大小

5. 定义变量 `int a=5,b=2;`, 表达式 `a/b` 的运行结果是 ()

A. 2.5 B. 2 C. 3 D. 2.0

答案: B

解析: 整型整除向下取整规则

6. 机器学习中用来训练模型的数据叫 ()

A. 测试集 B. 训练集 C. 验证集 D. 输出集

答案: B

7. 已知 `int m=8,n=5`; 表达式 `m-n*2` 值为 ()

A. 6 B. -2 C. 16 D. 3

答案: B

解析: 混合表达式优先级运算

8. 关于编译和解释的说法, 正确的是 ()

- A. 编译型语言逐行执行, 调试方便
- B. 解释型语言一次性生成可执行文件, 执行速度快
- C. C++是编译型语言, 执行效率较高
- D. Python 是编译型语言

答案: C

解析: 编译型语言 (如 C++) 一次性编译成可执行文件, 执行快; 解释型语言 (如 Python) 逐行执行, 调试方便但速度较慢。

9. 下列属于人工智能基础概念的是

- A. 二进制换算 B. 神经网络属于机器学习分支
- C. 数组内存连续 D. 左移等价乘 2

答案: B

解析: AI 体系: 人工智能包含机器学习, 机器学习包含深度学习 (神经网络)

10. 下面代码用来找出输入的 N 个正整数中最大的一个。如果将代码

段用流程图来表示,则 下面 if 语句的代码行应 该使用的图形是()。

```
int N, max=0, val;
```

```
cin >> N;
```

```
while(N){
```

```
cin >> val;
```

```
if(val > max)
```

```
max = val;
```

```
N--;
```

```
}
```

```
cout << max;
```

A. 圆形框

B. 椭圆形框

C. 平行四边形框

D. 菱形框

答案: D

11. 以下代码运行输出结果为 ()

```
int a=2;
```

```
if(a>5)
```

```
    cout<<1;
```

```
else
```

```
cout<<2;
```

A. 1 B. 2 C. 无输出 D. 编译错误

答案：B

解析：双分支 if 程序阅读

12. 以下字符的 ASCII 码值最小的是（ ）

A. '0'

B. 'A'

C. 'a'

D. 空格

答案：D

解析：ASCII 码值：空格=32，'0'=48，'A'=65，'a'=97。空格最小。

13. 执行 `int a = 3.9;`后，变量 a 的值是（ ）

A. 3

B. 4

C. 3.9

D. 编译错误

答案：A

解析：浮点数转整型时直接截断小数部分（不是四舍五入），3.9 截断为 3。

14. 若要遍历数组 `int a[10]` 的所有元素，正确的循环条件是 ()

A. `for(int i=0; i<=10; i++)`

B. `for(int i=0; i<10; i++)`

C. `for(int i=1; i<=10; i++)`

D. `for(int i=1; i<10; i++)`

答案: B

解析: 索引范围 0~9，所以循环条件应为 `i<10`。

15. 以下代码片段的功能是 ()

```
int sum = 0;
```

```
for(int i=0; i<n; i++) {
```

```
sum += a[i];
```

```
}
```

A. 查找数组最大值

B. 计算数组元素之和

C. 查找数组最小值

D. 反转数组

答案: B

解析: 循环遍历数组并将每个元素累加到 `sum` 中，实现求和。

16. 以下会造成**死循环**的代码是 ()

A. `for(int i=1; i<5; i++)` B. `int i=0; while(i++<10)`

C. `int i=1;while(i>0)i++;` D. `do{}while(0);`

答案：C

解析：死循环判定、循环变量迭代

17. 以下函数调用后，变量 x 的值是（ ）

```
void func(int a) {
```

```
    a = a + 5;
```

```
}
```

```
int main() {
```

```
    int x = 10;
```

```
    func(x);
```

```
    cout << x;
```

```
}
```

A. 5

B. 10

C. 15

D. 编译错误

答案：B

解析：C++默认是值传递，函数内修改的是参数的副本，不影响原变量。

18. 代码 `int s=0;for(int i=1;i<=4;i++) s+=i;` s 最终值为（ ）

A. 4 B. 8 C. 10 D. 16

答案: C

解析: 循环累加求和程序阅读

19. 下面 4 个选项中, 与下面 C++ 代码段具有相同效果的是()。

```
i = 0;
```

```
while (i*i<25){
```

```
    cout << i;
```

```
    i += 1;
```

```
}
```

A. for (i = 0; i < 5; i++)

```
    cout << i;
```

B. for (i = 1; i < 25; i++)

```
    cout << i;
```

C. for (i = 0; i*i < 26; i++)

```
    cout << i;
```

D. for (i = 1; i < 6; i++)

```
    cout << i;
```

答案: A

20. 下面 C++ 代码执行后输出是 ()。

```
#include<bits/stdc++.h>
```

```
using namespace std;

int main() {

    int n = 8;

    while (n >= 0) {

        n -= 1;

        if (n % 2 == 0)

            continue;

        if (n == 7)

            break;

    }

    cout << n;

}
```

A. 0 B. 5 C. 6 D. 7

答案： D

二、编程题，共 4 题，每题 15 分，共 60 分

1. 智能温控助手

【题目描述】

SACCC 实验室里有一个智能空调。根据传感器传回的温度，空调需要自动决定运行模式。请编写程序，帮助空调做出判断。

逻辑规则：

如果温度 t 小于 20，说明需要制热，所以输出 Heating 。

如果温度 t 大于 26，说明需要制冷，所以输出 Cooling 。

如果温度在 20 到 26 之间（包含），说明比较舒服，所以输出 Comfortable 。

【输入格式】

一行，一个整数 t ($0 \leq t \leq 50$)，表示当前温度。

【输出格式】

一行，表示空调模式的字符串。

【样例输入 1】

10

【样例输出 1】

Heating

【样例输入 2】

40

【样例输出 2】

Cooling

参考代码：

```
#include <iostream>

using namespace std;

int main() {
    int t;
    cin >> t;
```

```
    if (t < 20) cout << "Heating";  
  
    else if (t <= 26) cout << "Comfortable";  
  
    else cout << "Cooling";  
  
    return 0;  
  
}
```

2. 回文数探测器

【题目描述】

在人工智能的数据清洗中，有时需要识别特殊的数字规律。回文数是一种正着读和反着读都一样的数字（例如 121）。请编写程序，判断一个多位数是否是回文数。

【输入格式】

一行，一个整数 n ($100 \leq n \leq 100000000$)。

【输出格式】

一行，如果是回文数，输出 Yes，否则输出 No。

【样例输入 1】

100

【样例输出 1】

No

【样例输入 2】

12321

【样例输出 2】

Yes

参考代码:

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
// 定义判断是否是回文数的函数
```

```
bool is_palindrome(int n) {
```

```
    int m = 0, t = n;
```

```
    while (t > 0) {
```

```
        m = m * 10 + t % 10;
```

```
        t /= 10;
```

```
    }
```

```
    return m == n;
```

```
}
```

```
int main() {
```

```
    int n;
```

```
    cin >> n;
```

```
    if (is_palindrome(n)) {
```

```
        cout << "Yes";
```

```
    } else {
```

```
        cout << "No";

    }

    return 0;

}
```

3. 数据清洗器

【题目描述】

SACCC 的偶数传感器采集了一组数据（整数），但其中混入了数值为奇数的无效数据。请编写程序，去掉其中的奇数，然后输出去掉的奇数的个数，并将剩下的有效数据按从大到小的顺序输出。

【输入格式】

$n + 1$ 行。

第一行，一个整数 n ($5 \leq n \leq 10000$)。

接下来的 n 行，每行一个整数 x ($0 \leq x \leq 10000$)。

【输出格式】

两行。

第一行输出奇数的个数，第二行输出剩下的有效数据。

【样例输入 1】

5

1 2 3 4 5

【样例输出 1】

3

4 2

【样例输入 2】

10

99 21 45 88 100 124 999 159 2 147

【样例输出 2】

6

124 100 88 2

参考代码：

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int main() {
```

```
    int n;
```

```
    cin >> n;
```

```
    int a[n];
```

```
    // 数组下标从 0 开始，到 n-1 结束
```

```
    for (int i = 0; i < n; i++) {
```

```
        cin >> a[i];
```

```
    }
```

```
    // 找到所有的奇数并统计
```

```
    int count = 0;
```

```
    for (int i = 0; i < n; i++) {
```

```

        if (a[i] % 2 == 1) {
            count++;
        }
    }

    cout << count << endl;

    // 排序，从小到大
    sort(a, a + n);

    for (int i = n - 1; i >= 0; i--) {
        if (a[i] % 2 == 0) {
            cout << a[i] << " ";
        }
    }

    return 0;
}

```

4. 哥德巴赫猜想验证

【题目描述】

SACCC 的 AI 数学家正在验证哥德巴赫猜想：任一大于 2 的偶数都可写成两个质数之和。请编写程序，输入一个偶数 n ，找到两个质数 p_1 和 p_2 ，使得 $p_1 + p_2 = n$ ，并且 $p_1 * p_2$ 的值最大。输出这个最大乘积。

【输入格式】

一个偶数 n ($4 \leq n \leq 10000$)。

【输出格式】

一个整数，表示最大乘积。

【样例输入 1】

8

【样例输出 1】

15

【样例输入 2】

50

【样例输出 2】

589

参考代码：

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
// 定义判断是否是质数的函数
```

```
bool is_prime(int n) {
```

```
    if (n <= 1) return false;
```

```
    if (n == 2) return true;
```

```
    for (int i = 2; i * i <= n; i++) {
```

```
        if (n % i == 0) return false;
```

```
    }

    return true;
}

int main() {
    int n;

    cin >> n;

    for (int p1 = n / 2; p1 >= 2; p1--) {
        int p2 = n - p1;

        if (is_prime(p1) && is_prime(p2)) {
            cout << p1 * p2;

            break;
        }
    }

    return 0;
}
```