

Camtree Digital Library



AI-Driven Teaching Innovation: A Practical Exploration of the Programming Internship and Image Processing Undergraduate Programs

| | |
|------------------|--|
| Author | Yang, Shuai |
| Title | AI-Driven Teaching Innovation: A Practical Exploration of the Programming Internship and Image Processing Undergraduate Programs |
| Publication date | 2026 |
| Download date | 2026-05-14 04:52:12 |
| Link to Item | https://hdl.handle.net/20.500.14069/1336 |

程序设计实习

课程介绍

杨帅

课前多叭歪

■ 关于作业

- 2周后提交, 具体布置及要求见教学网
- 要自己好好写!!! 为啥?

■ 关于大模型

- 什么时候用
- 怎么用

■ DeepSeek 与 ChatGPT

- 所有作业都能直接搜到答案
- 鼓励辅助
- 学到真本事为目标

题目1 合并编码

题源：程序设计实习2024期末考试

Deepseek回答正确

D:合并编码

总时间限制：13000ms 单个测试点时间限制：1000ms 内存限制：262144kB

描述

给定一个合并编码后的字符串，返回原始字符串。

编码规则为：k[string]，表示其中方括号内部的string重复 k 次，k 为正整数且 $0 < k < 20$ 。

可以认为输入数据中所有的数字只表示重复的次数k，且方括号总是满足要求的，例如不会出现像 3a 、 2[b 或 2[4] 的输入。

输入保证符合格式，且不包含空格。

输入

一行，一个长度为n($1 \leq n \leq 500$)的字符串，代表合并编码后的字符串。

输出

一行，代表原始字符串。长度m满足 $0 \leq m \leq 1500$ 。

样例输入

样例1:
3[abc]1[o]2[n]

样例2:
3[a2[c]]

样例输出

样例1:
abcabcabconn

样例2:
accaccacc

Deepseek-R1思维链：

嗯，我现在得解决这个问题。题目是说给定一个合并编码后的字符串，要求返回原始字符串。编码规则是k[string]，也就是方括号内的字符串重复k次。比如输入3[abc]1[o]2[n]，输出应该是abcabcabconn。那我得怎么做呢？

首先，我得理解问题。这个编码可能有嵌套的情况，比如样例2中的3[a2[c]]，应该先处理里面的2[c]，得到acc，再重复3次变成accaccacc。这说明处理的时候需要处理嵌套的结构，可能要用栈来保存当前的状态，比如当前的重复次数和已经处理的部分字符串。

那我的思路应该是这样的：使用栈来处理嵌套的结构。每次遇到数字的时候，可能需要解析这个数字，然后遇到左括号时，将当前的字符串和重复次数压入栈中，然后重置当前的状态。当遇到右括号时，弹出栈顶的元素，将当前的字符串重复相应的次数，然后拼接到之前的字符串后面。

那如何具体实现呢？我需要遍历输入字符串的每个字符。在遍历过程中，处理数字、左括号、右括号以及普通字符。

步骤分解

- 初始化一个栈，用于保存每次遇到左括号前的状态（之前的字符串和重复次数）。
- 维护当前正在构建的字符串current_str，以及当前的数字current_num。
- 当遇到数字时，需要处理多位数的情况。比如，输入中有12[ab]，那么current_num需要累加为12。
- 当遇到左括号时，将当前的current_str和current_num压入栈，然后重置current_str和current_num。
- 当遇到右括号时，弹出栈顶的(prev_str, num)，将current_str重复num次，然后拼接回prev_str后面，形成新的current_str。
- 当遇到普通字符时，直接添加到current_str中。

那现在代码的大体结构应该没问题。那现在需要考虑栈的结构。在C++中，可以用stack来保存pair<string, int>。每次压栈的时候，保存当前的字符串和数字，然后重置。



cpp

```
#include <iostream>
#include <stack>
using namespace std;

string decodeString(string s) {
    stack<pair<string, int>> st;
    string current_str = "";
    int current_num = 0;

    for (char c : s) {
        if (isdigit(c)) {
            current_num = current_num * 10 + (c - '0');
        } else if (c == '[') {
            st.push({current_str, current_num});
            current_str = "";
            current_num = 0;
        } else if (c == ']') {
            auto top = st.top();
            st.pop();
            string temp;
            for (int i = 0; i < top.second; ++i) {
                temp += current_str;
            }
            current_str = top.first + temp;
        } else {
            current_str += c;
        }
    }
    return current_str;
}

int main() {
    string s;
    cin >> s;
    cout << decodeString(s) << endl;
    return 0;
}
```

思维链结束后的具体输出：

要解决这个问题，我们需要解码给定的字符串，其中包含嵌套的重复模式。编码规则为 `k[string]`，表示字符串 `string` 重复 `k` 次。我们将使用栈来处理嵌套结构，确保正确解析重复次数和对应的字符串。

方法思路

1. **栈的使用**：利用栈来保存当前处理的状态（当前的字符串和重复次数），以处理嵌套的重复模式。
2. **遍历字符**：逐个字符处理输入字符串，分为四种情况处理：
 - **数字字符**：累积当前重复次数。
 - **左括号 [**：将当前状态压入栈，并重置当前字符串和重复次数。
 - **右括号]**：弹出栈顶状态，生成重复后的字符串，并与之前的字符串拼接。
 - **普通字符**：直接添加到当前字符串中。

代码解释

- **栈结构**：使用 `stack<pair<string, int>>` 来保存当前字符串和重复次数。
- **遍历字符**：逐个处理字符，遇到数字时累积重复次数，遇到左括号时保存当前状态并重置，遇到右括号时生成重复字符串并拼接，普通字符直接追加到当前字符串。
- **重复字符串生成**：通过循环将当前字符串重复指定次数，并与之前的字符串拼接。

这种方法有效处理了嵌套的重复模式，确保在正确的时间点保存和恢复处理状态，从而高效解码输入字符串。

Deepseek既会输出正确代码，也会给出问题的方法思路以及代码的具体解释

题目2 编程填空：整数整合问题

题源：程序设计实习周末上机练习

Deepseek回答正确

08:编程填空：整数整合问题

总时间限制: 1000ms 内存限制: 65536kB

描述

写出函数中缺失的部分，使得函数返回值为一个整数,该整数的后x位和前y位与n相同，其他位和m相同。请补全calc函数使得程序能达到上述的功能

```
#include<iostream>
using namespace std;
unsigned int calc(unsigned int n,unsigned m,int x,int y) {

// 在此处补充你的代码

}

int main() {
    int x, y, t;
    unsigned int n, m;
    cin >> t;
    while (t--) {
        cin >> n >> m >> x >> y;
        cout << calc(n, m, x, y) << endl;
    }
    return 0;
}
```

输入

第一行一个整数N，表示多组测试数据。
接下去每行4个整数，分别为n, m ,x, y。 (x+y <= 32)

输出

每组测试数据，输出一个整数，表示题目中的结果。

样例输入

```
2
258079 4064 5 20
5 2 1 30
```

样例输出

```
262143
7
```

Deepseek-R1思维链：

需要补全的calc函数，其目标是组合整数n和m的位段。具体来说，生成的整数应具有n的低x位和m的其余高位中的前y位，中间部分则由m填补。

要解决这个问题，可以分以下步骤：

步骤分析

1. 获取n的后x位：

- 使用掩码 $n \& ((1U \ll x) - 1)$ 获取n的最右x位。

2. 获取m的高y位（相对剩余位）：

- 总共有32位，保留除去低x位后的高位，其中前y位需要保留，其余部分置0。
- 移位并掩码处理：将m右移 $(32 - y)$ 位，再左移x位，得到高位部分的正确位置。

3. 中间位覆盖：

- 处理需覆盖m中间位的部分，这可能需具体分析。

核心代码：

```
unsigned int calc(unsigned int n, unsigned m, int x, int y) {
    // 取得n的低x位
    unsigned int n_low = n & ((1U << x) - 1);

    // 取得m的高y位
    unsigned int m_high = (m >> (32 - y)) << x;

    // 组合这两个部分，中间部分由m的相应位填补
    return (m_high | n_low);
}
```

修正思考中的错误

错误修正：

原代码未正确处理32位全范围，且中间部分应由m的位组成。正确的处理方式是通过掩码提取m的高位，并确保y位的正确位置。

重新审视题目

例如，原题要求结果的“前y位”应该与n相同。这里的“前y位”可能指的就是整数的高y位。例如，对于无符号整数32位来说，最高的位是第31位，那么前y位是指最高的y位，比如最高三位是指第31-29位。

例如， $m=2$ (32位为000...0010)。

$\text{mask_high} = \text{for } y=30, (0xFFFFFFFF \ll (32 - y)) = 0X3FFFFFF \ll 2$?可能不正确，需要更深入位操作的理解。

综上，或许问题的正确解法并不如之前所想，可能需要重新构建各部分的掩码。

不断进行自身纠错以得到最终正确答案

可能的正确步骤应为：

结果的组成部分：

- 最高位的y位由n的最高y位构成。
- 最低的x位由n的最低x位构成。
- 中间的其余位数由m的中间位数构成。

这三个部分的相对位置是：最高前y位，中间 $(32 - y - x)$ 位，最后x位。例如，当 $y+x \leq 32$ 时，这三者不存在冲突，各自占据不同的位位置。因此：

构造三个部分：

例如，假设结果的高y位为a，中间为b，低x为c。那么：

$\text{result} = (a) | (b) | (c)$

思维链结束后的具体输出：

```
#include<iostream>
using namespace std;

unsigned int calc(unsigned int n, unsigned int m, int x, int y) {
    unsigned int mask_high = y ? ~((1U << (32 - y)) - 1) : 0;
    unsigned int a = n & mask_high;
    unsigned int mask_low = (1U << x) - 1;
    unsigned int c = n & mask_low;
    unsigned int mask_mid = ~mask_high & ~mask_low;
    unsigned int b = m & mask_mid;
    return a | b | c;
}

/* 代码解析：
1. **mask_high** 用于提取n的高y位：通过构造掩码`~((1U << (32 - y)) - 1`
2. **a** 保留n的高y部分后的值。
3. **mask_low** 用于提取n低x位的掩码。
4. **c** 保留n的末x位值。
5. **mask_mid** 构造中间位的掩码，将m中不属于高y位和低x位的部分保留。
6. **b** 取自m的中间位置部分。
7. 最后拼接a、b、c部分得到最终结果。
*/
```

生成代码中也会有较为充分的注释，以便于阅读和理解

题目3 编程填空：Dog! Dog! Dog!

题源：程序设计实习周末上机练习

Deepseek回答错误

03:编程填空：Dog! Dog! Dog!

总时间限制：1000ms 内存限制：65536kB

描述

程序填空，编写Dog类，使得程序按样例进行输出

```
#include <iostream>
using namespace std;
class Mammal {
public:
    Mammal() { cout << "Mammal constructor...\n"; }
    virtual ~Mammal() { cout << "Mammal destructor...\n"; }
    Mammal (const Mammal & rhs);
    virtual void Speak() const { cout << "Mammal speak!\n"; }
    virtual Mammal* Clone() { return new Mammal(*this); }
};

class Dog : public Mammal {
// 在此处补充你的代码

};

Mammal::Mammal (const Mammal & rhs) {
    cout << "Mammal Copy Constructor...\n";
}

Dog::Dog(){ cout << "Dog constructor...\n"; }
Dog::~Dog(){ cout << "Dog destructor...\n"; }
void Dog::Speak()const { cout << "Woof!\n"; }

int main() {
    Mammal* x=new Dog, *y=new Mammal;
    Mammal* x1=x->Clone(), *y1=y->Clone();
    x1->Speak(); y1->Speak();
    return 0;
}
```

样例输出

```
Mammal constructor...
Dog constructor...
Mammal constructor...
Mammal Copy Constructor...
Dog copy constructor...
Mammal Copy Constructor...
Woof!
Mammal speak!
```

Deepseek-R1思维链：

首先看题目描述，需要编写Dog类，使得程序的输出符合样例。样例的输出是：

```
Mammal constructor...
Dog constructor...
Mammal constructor...
Mammal Copy Constructor...
Dog copy constructor...
Mammal Copy Constructor...
Woof!
Mammal speak!
```

这应该是运行main函数中的代码后的输出结果。我需要仔细分析main函数里的步骤以及需要这些步骤生成什么样的函数调用。

首先看main函数的内容：

```
Mammal* x=new Dog, *y=new Mammal;
```

这会动态创建两个对象，一个Dog和一个Mammal。然后：

```
Mammal* x1=x->Clone(), *y1=y->Clone();
```

这行调用各个对象的Clone方法，得到两个新的指针x1和y1，最后：

```
x1->Speak(); y1->Speak();
```

这会导致调用Speak方法，预期输出分别为"Woof!"和"Mammal speak!"。

所以要使得当用Dog对象调用Clone时，返回的是一个新的Dog对象的拷贝，这样才能在x1->Speak()时显示"Woof!"。

因此，在Dog类中，需要override Clone函数：

```
virtual Dog* Clone() { return new Dog(*this); }
```

因为在样例输出中，有一个“Dog copy constructor...”的行。例如，创建x1时调用了Clone，这会调用Dog的复制构造函数。而Dog的复制构造函数会先调用基类Mammal的复制构造函数，然后输出Dog自己的复制构造函数信息。

所以，Dog的复制构造函数需要定义，并且在其初始化列表中调用基类的复制构造函数，并且函数体输出"Dog copy constructor..."。

Deepseek的答案（错误）：

代码补全如下：

```
class Dog : public Mammal {
public:
    Dog(const Dog& d) : Mammal(d) {
        cout << "Dog copy constructor...\n";
    }
    virtual Dog* Clone() override {
        return new Dog(*this);
    }
};
```

错误原因：类外定义成员函数在类内缺少函数声明

AC答案：

```
class Dog : public Mammal {
public:
    Dog();
    ~Dog();
    void Speak() const;
    Dog(const Dog& d) : Mammal(d) {
        cout << "Dog copy constructor...\n";
    }
    virtual Dog* Clone() override {
        return new Dog(*this);
    }
};
```

回看Deepseek-R1的思维链：

但代码的现有部分中，用户不能改变类外的定义吗？例如，现有的部分已经定义了Dog的构造函数和析构函数？

看原题的Dog类的函数定义情况：

在用户给出的代码中，Dog的构造函数和析构函数的定义已经写在了类体外：

```
Dog::Dog(){ cout << "Dog constructor...\n"; }
```

```
Dog::~Dog(){ cout << "Dog destructor...\n"; }
```

另外，Dog的Speak函数也是在类体外定义：

```
void Dog::Speak()const { cout << "Woof!\n"; }
```

这说明，在Dog类体中，应该已经声明了这些函数。即，在类体内部，有对应的函数声明。可能原来的Dog类声明中有析构函数和Speak的声明。假设在需要补全的Dog类的代码中包含这些声明？

但是当前给出的Dog类的定义是空的，因为被留为问题需要填空的部分？

意识到了函数声明的问题，但是最终没有改正认为“应该已经声明”，但实际类内并未声明

课程介绍

- 教师介绍
- 课前多叭歪
- 课程介绍
 - 课程调整
 - 进度安排
 - 成绩评定

命运多舛 Vs. 新篇章新挑战

- 本学期课程组计划做 **翻天覆地的**调整
- 课程安排的调整
- 课程大作业的调整
 - QT技术赏析**前移**（计划在清明前）
 - QT大作业与Python大作业**任选**
 - 路演加分

课程大纲 2024

第一讲 C语言补充知识

第二讲 从C走进C++

第三讲 类和对象 1

第四讲 类和对象 2

第五讲 运算符重载

第六讲 派生与继承

第七讲 虚函数与多态

第八讲 输入输出流和文件操作

C++程序设计
基础部分

第九讲 类模板与函数模板

第十讲 标准模板库 (I)

第十一讲 标准模板库 (II)

第十二讲 标准模板库 (III)

第十三讲 C++11与高阶

第十四讲 习题课

第十五讲 QT技术分享

C++程序设计
进阶部分

第十六讲 枚举

第十七讲 递归

第十八讲 动态规划

第十九讲 深度优先搜索

第二十讲 广度优先搜索

第二十一讲 习题课

第二十二讲 大作业路演

算法设计基础

课程大纲 2025

01. 课程介绍 / 类和对象1

02. 类和对象2

03. 运算符重载

04. 派生和继承

05. 虚函数和多态

C++程序设计
基础部分

Python
程序设计部分

06. C++程序员的Python修养1

07. C++程序员的Python修养2

08. C++程序员的Python修养3

09. C++程序员的Python修养4

10. QT技术分享

应用实践篇

11. 输入输出流 & 文件操作

12. 类模板与函数模版

13. 标准模板库STL-1

14. 标准模板库STL-2

15. C++11 & 高阶

C++程序设计
进阶部分

16. Python应用实践1

17. Python应用实践2

18. Python应用实践3

应用实践篇

课程内容和目标

- 论一个C++程序员的Python修养
 - 与你们一起拭目以待 (ssfd)
 - 希望可以辅助大家AI引论/基础课程的编程技术
 - 未来支撑大家的AI技术新探索