**String**

1. 到现在这个阶段了，我们就讲解基础的部分了，我们直接上代码了呀
   * 1. 头文件

#pragma once

#include <iostream>

#include <vector>

#include <stdio.h>

class String {

char \*s;

int \_size;

public:

//因为我们不知道用户传进来得字符串有多长，所以我们用copy得方式

String(const char\* str) {

if (!str) {

return;

}

//计算字符长度，然后开辟s的空间

int n = 0;

while (str[n++]);

n--;

s = new char[n + 1];

//进行赋值操作

for (int j = 0; j <= n; j++)

s[j] = str[j];

\_size = n;

}

//返回长度

int size() {

return \_size;

}

//取值操作

char operator[](int i)const {

return s[i];

}

char& operator[](int i) {

return s[i];

}

//求pos位置开始，n个字符

String sub\_string(int pos, int n) {

if (pos<0 || pos + n >= \_size || pos >= \_size) {

return NULL;

}

char \*p = new char[n + 1];

for (int i = 0, j = pos; i < n; i++, j++) {

p[i] = s[j];

}

p[n] = '\0';

String str(p);

delete[] p;

return str;

}

// 拼接操作

void operator +=(String str) {

int len = \_size + str.size(); //获取到长度

char\* p = new char[len];

int i = 0;

//复制第一个字符串

while (i < \_size) {

p[i] = s[i];

i++;

}

//复制第二个字符串

for (int j = 0; j < str.size(); j++, i++) {

p[i] = str[j];

}

delete[] s;

s = p;

\_size = len;

}

//暴力匹配字符串,从起始pos的位置开始查找

void find(String T, int pos = 0) {

int i = pos;

int j = 0;

while (i < \_size && j < T.size()) {

if (s[i] == T[j]) {

i++;

j++;

}

else {

i = i - j + 1;

j = 0;

}

}

if (j >= T.size()) {

std::cout << "在" << i - j << "处开始匹配" << std::endl;

}

}

//输出重载

friend std::ostream& operator<<(std::ostream& os, String& str);

};

//输出重载

std::ostream& operator<<(std::ostream& os, String &str) {

return os << str.s;

}

* + 1. Main

#include <iostream>

#include "String.h"

#include <stdio.h>

int main()

{

String s1("weierLin");

String s2("hello ");

String s3("er");

std::cout << s1 << std::endl;

String sub = s1.sub\_string(2, 3);

std::cout << sub << std::endl;

s2 += s1;

std::cout << s2 << std::endl;

s1.find(s3);

}