



C++程序设计

C++ Programming

实验报告

Experiment Report

专业(Major): 计算机科学与技术 2502 班

学号(ID): 202512898

姓名(Name): 田佩宁

提交日期(Date): 2025 年 11 月 24 日

成绩(Score):

东北大学悉尼智能科技学院
Sydney Smart Technology College, Northeastern University



实验编号 (Experiment No.) :1

实验名称 (Experiment Name) :C++程序中的函数与预处理

实验目的 (Experiment Purpose) :

掌握编程环境,理解 C++程序编译执行的过程;掌握 C++程序设计和调试方法;掌握基本的数据类型和表达式定义方法;掌握输入/输出、三种基本结构定义方法;掌握实现函数定义与调用方法。

实验内容 (Experiment Contents) :

(1) 一个百万富翁碰到一个陌生人,陌生人找他谈了一个换钱的计划。该计划如下:我每天给你 10 万元,而你第一天给我一元,第二天我仍给你十万,你给我二元,第三天我仍给你十万,你给我四元.....你每天给我的钱是前一天的两倍,直到满 $n(0 \leq n \leq 30)$ 天。百万富翁非常高兴,欣然接受了这个契约。请编写一个程序,计算这 n 天中,陌生人给了富翁多少钱,富翁给了陌生人多少钱。(必做)

(2) 编写一程序,从键盘输入一个三位正整数,然后反向输出对应的数,如果输入的数不是三位正整数,则输出 -1。(必做)

(3) 写一函数求 $\sinh(x)$ 的值,求 $\sinh(x)$ 的近似公式为: $\sinh(x) = (e^x - e^{-x})/2$, 其中用一个函数求 e^x 。(必做)

(4) 编写函数 `int sum(int x)`, 求整数 x 的各位数字之和。在 `main` 函数中测试该函数:从键盘输入一非负整数,然后调用 `sum` 函数计算各位数字之和并输出结果。(必做)

(5) 编写函数 `fac(n)`, 用递归法求出 n 的阶乘。在程序中使用此函数,将输入的整数 n 的阶乘求出并输出到控制台。(必做)

(6) 利克瑞尔数 (Lychrel Number) 指的是将该数各数位逆序翻转后形成的新数相加,并将该过程反复迭代后,结果永远无法是一个回文数的自然数。57 就是一个非利克瑞尔数: $57+75=132$, $132+231=363$, 363 是一个回文数。请编写程序,输入一个自然数 (非利克瑞尔数), 请计算其最终的回文数是多少及每次迭代过程。注意:假设输入的整数和中间产生的整数都不超过 `int` 数据类型的表示范围。

(7) 编写一个函数 `invert(s)`, 颠倒输入的字符串参数 s 后返回。在 `main()` 函数中测试此函数:从键盘输入一个正整数 $n (n \in [1,20])$, 然后再输入 n 个字符串 (长度不大于 100), 对于每一个字符串,然后调用 `invert` 函数进行颠倒字符串操作,将颠倒后的字符串打印在屏幕上。(选做)

(8) 编写一个函数 `isprime(n)`。判断整数 n 是否为素数。编写程序使用此函数,当输入一个整数时,对它进行判断,当为素数时,输出 1。否则,输出 0。(选做)

实验结果 (Experiment Results) :



(1)

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

int main(){
    int n; cin>>n;
    long long a(0);int b(0);
    int temp(1);
    for (int i=0; i<n; i++){
        a+=100000, b+=temp, temp*=2;
    }
    cout<<a<<"\n"<<b<<"\n";
    cout<<"\n田佩宁 202512898\n";
    return 0;
}
```

10
1000000
1023

田佩宁 202512898

(2)

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

int main(){
    string a;cin>>a;
    for (int i=a.length()-1; i>=0; i--){
        if(a.length()!=3) {cout<<-1;break;}
        if(a=="300") {cout<<3;break;}
        cout<<a[i];
    }
    cout<<"\n";
    cout<<"\n田佩宁 202512898\n";

    return 0;
}
```

356
653

田佩宁 202512898

(3)

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

int main(){
    int x; cin>>x;
    cout<<fixed<<setprecision(5)
        <<(exp(x)-exp(x: -x))/2<<endl;
    cout<<"\n田佩宁 202512898\n";
    return 0;
}
```

2
3.62686

田佩宁 202512898

(4)

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

int sum(string a){
    int ans(0);
    for(int i=0; i<(int)a.length(); i++){
        ans += (a[i]-'0');
    }
    return ans;
}
```

```
int main(){
    string n; cin>>n;
    cout<<sum(a: n);
    cout<<"\n田佩宁 202512898\n";
    return 0;
}
```

58
13
田佩宁 202512898



(5)

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

int fac(int n){
    if(n>0){
        return n*fac(n-1);
    }else{
        return 1;
    }
}

int main(){
    int n; cin>>n;
    cout<<fac(n);
    cout<<"\n田佩宁 202512898\n";
    return 0;
}
```

5

120

田佩宁 202512898

(8)

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

int isprime(int n) {
    if (n <= 1) return 0;
    if (n == 2) return 1;
    if (n % 2 == 0) return 0;

    int limit = sqrt(x: n);
    for (int i = 3; i <= limit; i += 2) {
        if (n % i == 0) return 0;
    }
    return 1;
}

int main() {
    int n;
    cin >> n;
    cout << isprime(n) << endl;
    cout<<"\n田佩宁 202512898\n";
    return 0;
}
```

45

0

田佩宁 202512898

(7)

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

string invert(string s) {
    reverse(first: s.begin(), last: s.end());
    return s;
}

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

    for (int i = 0; i < n; i++) {
        string str;
        getline(&is: cin, &: str);
        cout << invert(s: str) << endl;
    }
    cout<<"\n田佩宁 202512898\n";
    return 0;
}
```

3

1AdkflsddfYjkdfDD
DDfdkjYfdds1fkda1
aaaaaaaaAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAA
12345678aBc
cBa87654321

田佩宁 202512898



(6)

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

bool isPalindrome(long long n) {
    string s = to_string(val: n);
    string rev_s = s;
    reverse(first: rev_s.begin(), last: rev_s.end());
    return s == rev_s;
}

long long reverseNum(long long n) {
    string s = to_string(val: n);
    reverse(first: s.begin(), last: s.end());
    return stoll(str: s);
}

int main() {
    long long num;
    cin >> num;
    int step = 0;
    if (isPalindrome(n: num)) {
        cout << num << endl;
        return 0;
    }

    while (!isPalindrome(n: num)) {
        long long rev = reverseNum(n: num);
        long long new_num = num + rev;
        step++;

        cout << step << ":" << num << "+" << rev << "=" << new_num << endl;

        num = new_num;
    }

    cout << num << endl;
    cout<<"\n田佩宁 202512898\n";
    return 0;
}
```

109

1:109+901=1010

2:1010+101=1111

1111

田佩宁 202512898

心得体会 (Experience) :

今天上机练习了 C++ 的函数和预处理。写函数时发现把功能拆成小块确实让程序清晰不少,但参数传递和返回值有时候会搞混,得边写边想。预处理部分, #include 和 #define 用起来挺方便,就是头文件包含的细节还得留心。动手调了几个小错误后,感觉对这些概念的理解比只看书要实在一些。



实验编号 (Experiment No.) :2

实验名称 (Experiment Name) :数组和指针的使用

实验目的 (Experiment Purpose) :

掌握数组的使用, 数组与函数调用的关系; 掌握指针的使用, 指针与函数、数组的关系。

实验内容 (Experiment Contents) :

- (1) 将一个数组中的值按逆序重新存放。(必做)
- (2) 从标准输入中输入两组整数(每行不超过 20 个整数, 每组整数中元素不重复), 合并两组整数, 去掉在两组整数中都出现的整数, 并按从大到小顺序排序输出(即两组整数集“异或”)。(必做)
- (3) 输入一个字符串, 含有数字和非数字字符, 如
“sum=abc+234;while(abc<=700)tab{ass=346++;bss=abc+267;}”, 将其中连续的数字作为一个整数, 依次存放到一个数组 nums 中。例如, 234 放在 nums[0], 700 放在 nums[1]……, 统计共有多少个整数, 并输出这些整数。结合指针完成该题。(必做)
- (4) 编写函数 itob(n, s, b), 用于把整数 n 转换成以 b 为基的字符串并存储到 s 中。编写程序, 使用函数 itob(n, s, b) 将输入的整数 n, 转换成字符串 s, 将 s 输出。转换后的字符串从最高的非零位开始输出。如果 n 为负数, 则输出的字符串的第一个字符为 '-'。b 为大于 1 小于 37 的任意自然数值。当 b=2 时, 输出字符只可能是 '0' 和 '1'; 当 b=16 时, 输出字符串中可能含有字符为 '0'-'9', 'a'-'f' (字母以小写输出)。b 还可以是其它数值。比如输入 n=33, b=17, 则输出 33 的 17 进制值为 "1g"。(选做)
- (5) 找出一个二维数组中的鞍点, 即该位置上的元素在该行上最大, 在该列上最小(也可能没有鞍点)。(选做)
- (6) 用筛选法求 100 之内的素数。所谓“筛选法”指的是“埃拉托色尼 (Eratosthenes) 筛选法”。他是古希腊著名数学家, 他采用的方法是: 在一张纸上写 1-100 全部素数, 然后逐个判断它们是否为素数, 找出一个非素数, 就把它挖掉, 最后剩下的就是素数。(选做)
- (7) 在必做题 (3) 的基础上将输入字符串中的连续的字母作为一个单词存放到一个数组 words 中; 如上面的字符串中, sum 放在 words[0], abc 放在 words[1]……, 统计共有多少个单词, 并输出这些单词。其它所有字符存放到一个数组 others 中, 如上面的字符串中, = 放在 others[0], + 放在 others[1]……, 统计共有多少个字符, 并输出这些字符。(选做)

实验结果 (Experiment Results) :



(1)

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

int a[100], flag;
int main(){
    for (int i=0; i<=100; i++){
        cin>>a[i];
        if (a[i]==0) {
            flag=i; break;
        }
    }
    for (int i=flag-1; i>=0; i--){
        cout<<a[i]<<(i==0 ? '\n' : ' ');
    }
    cout<<"\n田佩宁 202512898\n";
}
```

8 6 5 4 1

1 4 5 6 8

田佩宁 202512898

(2)

```
#include <stdio.h>
#define MAX_SIZE 100

int main() {
    int a[MAX_SIZE], b[MAX_SIZE], result[MAX_SIZE * 2];
    int lenA = 0, lenB = 0, lenResult = 0;
    char ch;
    // 读取
    while (scanf(format: "%d", &a[lenA]) == 1) {
        lenA++;
        if ((ch = getchar()) == '\n') break;
    }
    while (scanf(format: "%d", &b[lenB]) == 1) {
        lenB++;
        if ((ch = getchar()) == '\n' || ch == EOF) break;
    }
    // 异或去重
    for (int i = 0; i < lenA; i++) {
        int found = 0;
        for (int j = 0; j < lenB; j++) {
            if (a[i] == b[j]) {
                found = 1;
                break;
            }
        }
        if (!found) {
            result[lenResult++] = a[i];
        }
    }
    for (int i = 0; i < lenB; i++) {
        int found = 0;
        for (int j = 0; j < lenA; j++) {
            if (b[i] == a[j]) {
                found = 1;
                break;
            }
        }
        if (!found) {
            result[lenResult++] = b[i];
        }
    }
    // 从大到小排序
    for (int i = 0; i < lenResult - 1; i++) {
        for (int j = 0; j < lenResult - i - 1; j++) {
            if (result[j] < result[j + 1]) {
                int temp = result[j];
                result[j] = result[j + 1];
                result[j + 1] = temp;
            }
        }
    }
    // 输出
    for (int i = 0; i < lenResult; i++) {
        printf(format: "%d", result[i]);
        if (i < lenResult - 1) printf(format: " ");
    }
    printf(format: "\n");
    printf(format: "\n田佩宁 202512898\n");
    return 0;
}
```

5 1 4 32 8 7 9 -6

5 2 87 10 1

87 32 10 9 8 7 4 2 -6

田佩宁 202512898



(3)

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

void itob(int n, string&s, int b){
    int b_num[100], len(0); memset(s, 0, n: sizeof(b_num));
    if (n<0) {
        cout<<'-' , n=-n;
    }
    for (int i=1; n!=0; i++, len++) {
        b_num[i] = n%b, n/=b;

        if(b_num[i]>=0 && b_num[i]<=9){
            s[i] = b_num[i]+'0';
        }else {
            s[i] = b_num[i]+'a';
        }
    }
    for (int i=len; i>=1; i--) {
        cout<<s[i];
    }
    cout<<'\\n';
}

int main(){
    int n, b;    cin>>n>>b;
    string s;
    itob(n, &s, b);
    cout<<"\\n田佩宁 202512898\\n";
}
```

5 2

101

田佩宁 202512898



(4)

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

int a[5][5], saddle[5], clm[5];
int main(){
    for (int i=0; i<3; i++) {
        for (int j=0; j<3; j++) {
            cin>>a[i][j];
            if (a[i][j]>saddle[i]) {
                saddle[i]=a[i][j], clm[i]=j;
            }
        }
    }

    bool found = false;
    for (int i=0; i<3; i++) {
        int flag(0);
        for (int j=0; j<3; j++) {
            if (a[j][clm[i]] < saddle[i]) flag=1;
        }
        if (flag==0) {
            cout<<"Saddle point:a["<<i<<"]["<<clm[i]<<"]="<<saddle[i]<<endl;
            found = true;
        }
    }

    if (!found) {
        cout<<"There is no saddle point"<<'\\n';
    }
    cout<<"\\n田佩宁 202512898\\n";
    return 0;
}
```

1 2 4

2 4 8

3 6 12

Saddle point:a[0][2]=4

田佩宁 202512898



(6)

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

void get_num(char* &ptr, int*nums, int cnt) {
    int value = 0;
    while (*ptr >= '0' && *ptr <= '9') {
        value = value * 10 + (*ptr - '0');
        ptr++; // 移动指针
    }
    nums[cnt] = value;
}

int nums[100];

int main(){
    string a;    getline(&is: cin, &str: a);    a += ' ';
    char*p = &a[0];
    int cnt = 0;
    while (*p != '\\0') {
        if (*p >= '0' && *p <= '9') {
            get_num(&ptr: p, nums, cnt);
            cnt++;
        } else {
            p++;
        }
    }

    cout << "There are " << cnt << " integers:";
    for (int i = 0; i < cnt; i++) {
        cout << nums[i] << (i!=cnt-1 ? ' ' : '\\n');
    }
    cout<<"\\n田佩宁 202512898\\n";
}
```

```
sum=abc+234;while(abc==700)tab{ass=346++;bss=abc+267;}
```

```
There are 4 integers:234 700 346 267
```

```
田佩宁 202512898
```



(5)

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

bool is_prime(int x){
    if (x<=1) return false;
    if (x==2) return true;
    for (int i=3; i<x; i++) {
        if (x%i == 0) return false;
    }
    return true;
}

int main(){
    cout<<2<<' ';
    for (int i=3; i<=100; i+=2) {
        if (is_prime(x: i)==true) {
            cout<<i<<(i==97 ? '\n' : ' ');
        }
    }
    cout<<"\n田佩宁 202512898\n";
}
```

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

田佩宁 202512898

(7)

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

class Token{
private:
    int nums[100], num_cnt, word_cnt, symbol_cnt;
    string words[100], a;
    char others[100];
    char*p;
public:
    Token(){
        memset(s: nums, c: 0, n: sizeof(nums));
        memset(s: words, c: '\0', n: sizeof(words));
        memset(s: others, c: '\0', n: sizeof(others));
        num_cnt=0, word_cnt=0, symbol_cnt=0;
    }
    int get_num_cnt(){
        return num_cnt;
    }
    int get_word_cnt(){
        return word_cnt;
    }
    int get_symbol_cnt(){
        return symbol_cnt;
    }
    int*get_nums(){
        return nums;
    }
    string*get_words(){
        return words;
    }
    char*get_others(){
        return others;
    }
    void extract_num() {
        int value = 0;
        while (*p >= '0' && *p <= '9') {
            value = value*10 + (*p-'0');
            p++; //移动指针
        }
        nums[num_cnt] = value;
    }
    void extract_word() {
        while (*p >= 'a' && *p <= 'z') {
            words[word_cnt] += *p;
            p++;
        }
    }
    void extract_symbol() {
        others[symbol_cnt] = *p;
        p++;
    }
}
```



(7) 【续】

```

void scan(){
    getline(&is: cin, &str: a);
    a += ' ';
    p = &a[0];
}

void traverse(){
    while (*p != '\0') {
        if (*p >= '0' && *p <= '9') {
            extract_num();
            num_cnt++;
        } else if (*p >= 'a' && *p <= 'z'){
            extract_word();
            word_cnt++;
        } else if (*p=='=' || *p=='+' || *p==';' || *p=='(' || *p=='(' || *p=='(' || *p=='{' || *p=='{') {
            extract_symbol();
            symbol_cnt++;
        } else {
            p++;
        }
    }
}

template<class T>
void print(int cnt, T*array, string label){
    cout << "There are " << cnt << ' ' << label;
    for (int i = 0; i < cnt; i++) {
        cout << array[i] << (i!=cnt-1 ? ' ' : '\n');
    }
}

};

int main(){
    Token obj;
    obj.scan();
    obj.traverse();
    obj.print<int>(cnt: obj.get_num_cnt(), array: obj.get_nums(), label: "integers:");
    obj.print<string>(cnt: obj.get_word_cnt(), array: obj.get_words(), label: "words:");
    obj.print<char>(cnt: obj.get_symbol_cnt(), array: obj.get_others(), label: "symbols:");
    cout<<"\n田佩宁 202512898\n";
}

```

```

sum=abc+234;while(abc==700)tab{ass=346++;bss=abc+267;}
There are 4 integers:234 700 346 267
There are 8 words:sum abc while abc tab ass bss abc
There are 16 symbols:= + ; ( = = ) { = + + ; = + ; }

```

田佩宁 202512898

心得体会 (Experience) :

这次 C++ 实验让我体会到数组名本质上是首元素地址的常量指针，而指针变量则能灵活遍历数组。特别是在用指针访问数组元素、进行地址算术运算时，我直观感受到内存操作的效率与风险——稍有不慎就会越界访问。

这次实验让我明白，必须同时注重指针的灵活性和安全性。只有准确理解每个指针指向的内存位置及其边界，才能写出既高效又可靠的代码。这为后续学习打下了坚实基础。