

C++ 程序设计实验

实验一 函数和输入输出

2022 年 ~ 2023 年 第 1 学期

专业年级: 统计 学生学号: _____ 学生姓名: _____

一、设计函数完成数组元素相应的求和

- 1) 从键盘输入若干整数, 存储为一维数组(也可以使用向量存储),
- 2) 设计求和函数: 计算并输出每对相邻元素的和, 如果读入的个数为奇数, 则提示用户最后一个元素没有求和, 并输出其值。
- 3) 设计求和函数: 头尾元素两两配对(第一个和最后一个, 第二个和倒数第二个, 以此类推), 计算每对元素得和, 并输出。
- 4) 在主函数调用 2) 和 3), 测试函数功能是否实现。
- 5) 从文件读入若干整数, 完成以上功能。

1 题目分析

考察知识点: 数组, 函数定义, If 语句, 文件的读取

具体分析: (1) 本题目要求不高, 具体难点在于题目要求的理解和文件读取的用法; (2) 先写一个数组存入键盘输入的整数, 再设计函数 2, 判断所输入的数字的数量再进行相加, 函数 3 为首尾相加; (3) 在主函数中调用, 设计读取文件的操作并再次代入计算。

2 程序代码

```
#include <iostream>
#include <vector>
#include <fstream>
using namespace std;
void sum1(const vector<int>&nums){
    int i;
    if(nums.size()%2==0){
        //size 函数返回所输入数字的个数
        for(i=0;i<nums.size();i=i+2){
            cout<<"求和结果为"<<nums[i]+nums[i + 1]<<" ";}}
    else{
```

C++ 程序设计实验

```
    cout<<"读入数据为奇数，最后一位数字没有求和"<<endl;
    for(i=0;i<nums.size()-1;i=i+2){
        cout<<"求和结果为"<<nums[i]+nums[i+1]<<" ";}
    i=nums.size()-1;
    cout<<"没有求和的数据： "<<nums[i]<<" ";}}

void sum2(const vector<int>&nums){
    int i,j;
    if(nums.size()%2==0){
        for(i=0,j=nums.size()-1;i<nums.size()/2;i++,j--){
            cout<<"求和结果为"<<nums[i]+nums[j]<<" ";}}
    else{
        for(i=0,j=nums.size()-1;i<(nums.size()-1)/2;i++,j--){
            cout<<"求和结果为"<<nums[i]+nums[j]<<" ";}}}

int type(){
    vector<int>nums;
    cout<<"请输入一组数（当输入-1 时结束）:";int n;
    while(cin>>n&& n!=-1){
        nums.push_back(n);}
    cout<<"两两相加: ";sum1(nums);cout<<endl;
    cout<<"首尾相加: ";sum2(nums);cout<<endl;
    return 0;}

int wenjian(){
    vector<int>nums;ifstream infile;int n;
    infile.open("C:\\Users\\86156\\Desktop\\c++\\numbers.txt");
    if(!infile){
        cout<<"文件读取失败"<<endl;
        return 0;}
    while(!infile.eof()){
        infile>>n;
        nums.push_back(n);}
    infile.close();
    cout<<"两两相加: ";sum1(nums);cout<<endl;
    cout<<"首尾相加: ";sum2(nums);cout<<endl;
    return 0;}

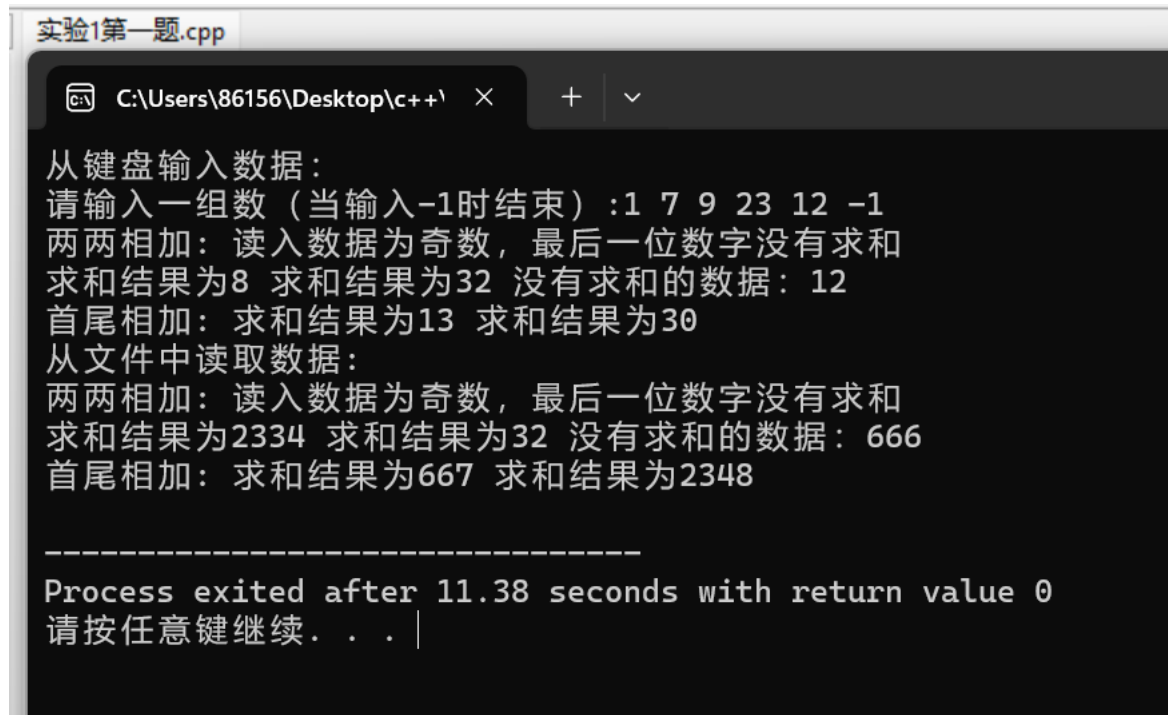
int main(){
```

C++ 程序设计实验

```
cout<<"从键盘输入数据: "<<endl;type());  
cout<<"从文件中读取数据:"<<endl;wenjian();return 0;}
```

3 运行结果以及分析

运行结果:



The screenshot shows a Windows command prompt window titled "实验1第一题.cpp". The command bar shows the file path "C:\Users\86156\Desktop\c++\". The program's output is as follows:

```
从键盘输入数据:  
请输入一组数 (当输入-1时结束):1 7 9 23 12 -1  
两两相加: 读入数据为奇数, 最后一位数字没有求和  
求和结果为8 求和结果为32 没有求和的数据: 12  
首尾相加: 求和结果为13 求和结果为30  
从文件中读取数据:  
两两相加: 读入数据为奇数, 最后一位数字没有求和  
求和结果为2334 求和结果为32 没有求和的数据: 666  
首尾相加: 求和结果为667 求和结果为2348  
  
-----  
Process exited after 11.38 seconds with return value 0  
请按任意键继续. . .
```

文件内容:



The screenshot shows a text editor window with a menu bar containing "文件", "编辑", and "查看". The main text area contains the following content:

```
1 2333 17 15 666
```

分析: 具体实现过程中, 从熟练度来讲, 困难点在于 vector 和 const 的使用方法的学习; 同时, 在函数的实现过程中, 拓展学习了 size() 函数, 了解到 size 函数在 vector 中的运用, 深入学习了 c++ 用法中函数的定义与使用。

二、设计函数完成字母的大小转换

- 1) 可以从键盘或者文件中读入一段文本到字符数组(或 `vector<string>` 对象), 每个单词存储为数组中的一个元素,
- 2) 设计函数, 把数组中每一个单词首字母转换为大写字母, 输出转化后的单词, 每个单词为一行输出。

1 题目分析

本题目运用知识点: 数组, 输入输出, `vector`, 函数的定义。

具体分析: (1) 提取单词并转换为大写的程序思维难度是不高的, 具体难点在于对 `c++` 代码及函数的运用; (2) 理解各个头文件的使用方向; (3) 理解程序流的意义。

2 程序代码

```
#include <iostream>
//使用的是输入输出流因此省略 sstream 头文件
#include <vector>
using namespace std;
string bigword(const string&word){
    string a=word;
    if(!a.empty()){
        //empty 函数用来判断是否为空
        if(a[0]>='a'&&a[0]<='z'){
            a[0]=a[0]-32;}}
    return a;}
void bigarr(vector<string>&words){
    for(string&word:words){
        //遍历 words 中的字符串存入 word 数组中
        word=bigword(word);}}
int main(){
    vector<string>words;//创建一个字符串容器 words
    cout<<"请输入:";string input;getline(cin,input);string word;
    for(char c:input){
        if(c==' '){
            if(!word.empty()){
```

C++ 程序设计实验

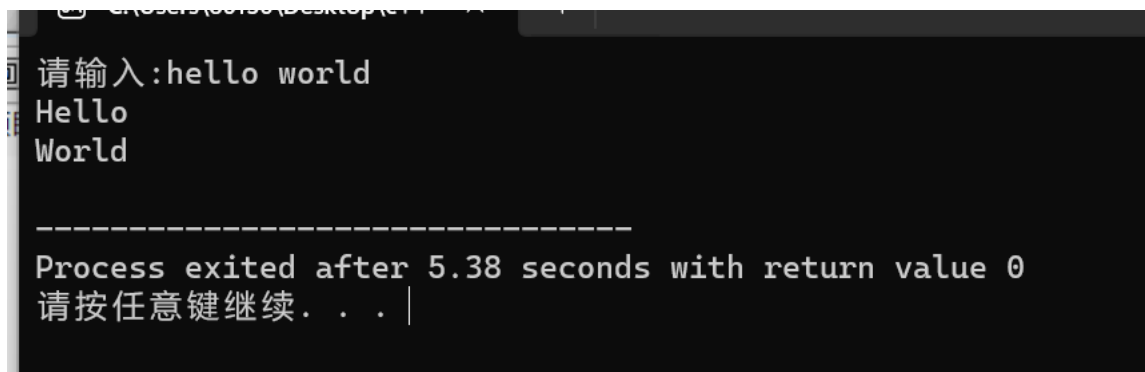
```
words.push_back(word);  
word.clear();}  
else{word+=c;}
```

//该函数表示提取一个完整的单词并存入 words 数组中，其中 word 是在每一次循环中都会清空的临时存储处

```
if(!word.empty()){  
    words.push_back(word);  
//再次检查循环，将最后一个单词也存入数组，避免遗漏  
bigarr(words);  
for(const string&word:words){  
    cout<<word<<endl;}  
//遍历数组中已转换的单词并输出  
return 0;}
```

3 运行结果以及分析

运行结果展示：



```
请输入:hello world  
Hello  
World  
  
-----  
Process exited after 5.38 seconds with return value 0  
请按任意键继续. . . |
```

分析：具体完成过程中，拓展学习了 `empty()`，`clear()`，`push_back()` 等函数，并加强了 `vector`，`const`，`getline` 的进一步使用，对比学习了遍历函数在 `python` 与 `c++` 的异同并学习了其具体用法。

三、实现矩阵的加法：

- 1) 在主函数中定义 3 个 4*4 矩阵。如 `result`，`source1`，`source2`
- 2) 调用 `add` 函数，实现矩阵的加法。 `result=source1+source2`
- 3) 调用 `showMatrix` 函数，输出 3 个矩阵

1 题目分析

C++ 程序设计实验

本题目运用知识点：二维数组，定义函数，循环，输出

具体分析：（1）理解矩阵即二维数组，学习矩阵的加法；（2）学习函数定义的使用；（3）学习循环遍历的用法。

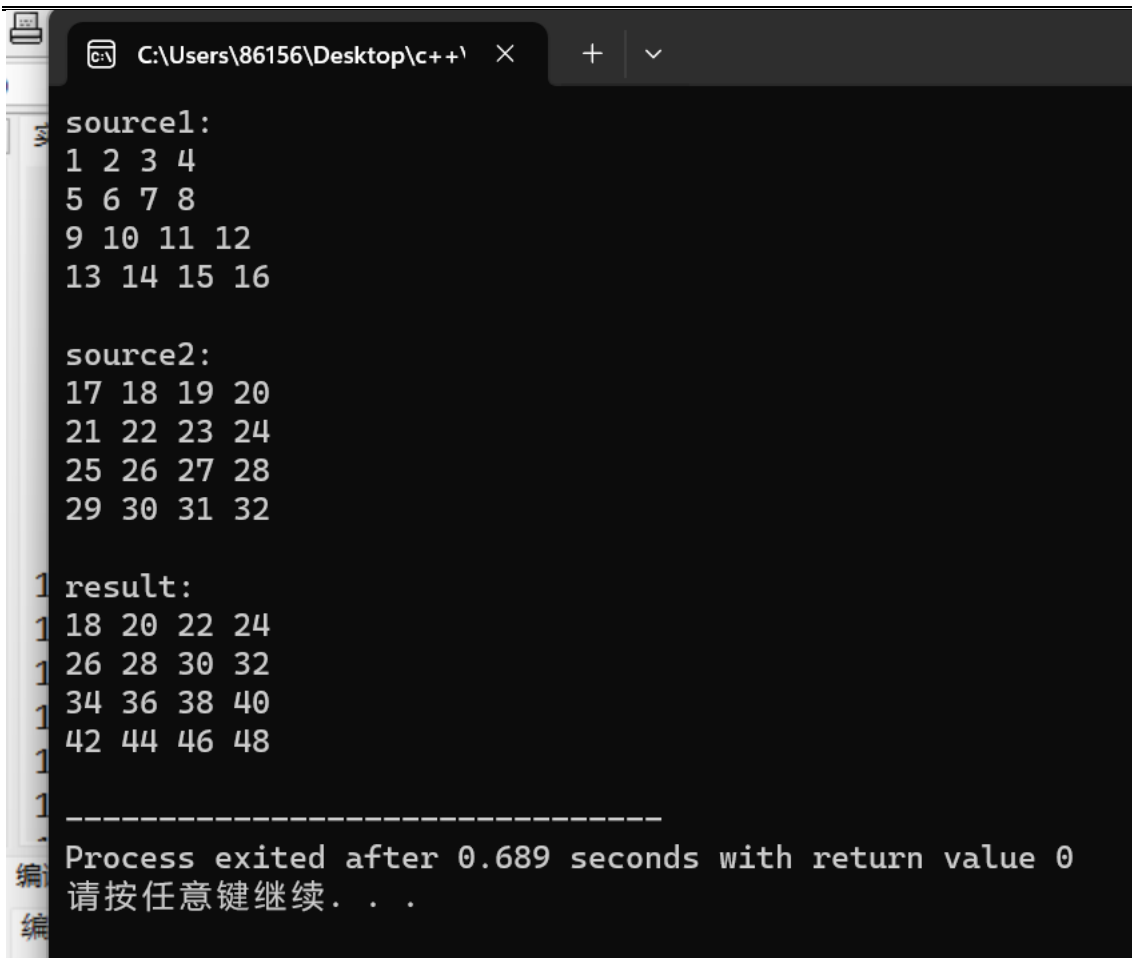
2 程序代码

```
#include <iostream>
using namespace std;
void add(int source1[][4],int source2[][4],int result[][4]){
    for(int i=0;i<4;i++){
        for(int j=0;j<4;j++){
            result[i][j]=source1[i][j]+source2[i][j];}}
void showMatrix(int show[][4]){
    for(int i=0;i<4;i++){
        for(int j=0;j<4;j++){
            cout<<show[i][j]<<" ";}
        cout<<endl;}}
int main(){
    int result[4][4];
    int source1[4][4]={
        {1,2,3,4},{5,6,7,8},{9,10,11,12},{13,14,15,16}};
    int source2[4][4] = {
        {17,18,19,20},{21,22,23,24},{25,26,27,28},{29,30,31,32}};
    add(source1,source2,result);
    cout<<"source1:"<<endl;showMatrix(source1);cout<<endl;
    cout<<"source2:"<<endl;showMatrix(source2);cout<<endl;
    //清空输出流避免出现错误
    cout<<"result:"<<endl;showMatrix(result);return 0;}
```

3 运行结果以及分析

运行结果展示：

C++ 程序设计实验



```
C:\Users\86156\Desktop\c++\ >
source1:
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16

source2:
17 18 19 20
21 22 23 24
25 26 27 28
29 30 31 32

1 result:
1 18 20 22 24
1 26 28 30 32
1 34 36 38 40
1 42 44 46 48
1
1
-----
Process exited after 0.689 seconds with return value 0
请按任意键继续. . .
```

分析:

本程序无额外困难点，主要目的在于熟练 c++ 的程序写法；定义矩阵后根据矩阵相加的规则设计循环函数，再根据循环函数输出矩阵。