



|  |  |  |  |
| --- | --- | --- | --- |
| **学号** | 17130130312 | **姓名** | 曹晨瑶 |
| **班级** | **1713013** | **任课教师** | **王献青** |
| **实验名称** | 简单类型、表达式和基本控制结构 | | |
| **实验学期** | **2018 – 2019 学年第2学期** | | |
| **实验日期** | 2019年4月13日 | **实验地点** | **G334** |
| **报告成绩** |  |  |  |

西安电子科技大学软件学院

# 实验目的

本次实验通过设计实现几道简单的程序，以熟悉 C++中的简单类型、基本表达式、基本的控制结构相关概念，并能够灵活运用相应机制，提高学生的编程水平。

# 实验环境

操作系统：Windows10

开发工具: Code::Blocks

# 实验内容

## 计算名字的相关数据

Read a sequence of possibly whitespace-separated (name,value) pairs, where the name is a single whitespace-separated word and the value is an integer or a floating-point value. Compute and print the sum and mean for each name and the sum and mean for all names. Hint: §6.1.8.

## 定义函数 atoi()

Write a function *atoi*(*const char*\*) that takes a string containing digits and returns the corresponding *int*. For example, *atoi*("*123*") is *123*. Modify *atoi*() to handle C++ octal and hexadecimal notation in addition to plain decimal numbers. Modify *atoi*() to handle the C++ character constant notation.

## 定义函数itoa（）

Write a function *itoa*(*int i*, *char b*[]) that creates a string representation of *i* in *b* and returns *b*.

# 数据结构与算法设计

## 计算名字的相关数据

**模块结构及文件组织设计：**

模块1：主控模块，仅包括文件main.cpp ，定义了 main()函数。

模块2：仅包含test.h文件，在文件中定义了wordinfo结构体。

**关键数据结构设计：**

数据结构：

const int SIZE=30;//将结构体数组大小定义为常量

struct wordinfo{

string name;//特定名字

double sum;//同一名称的总和

int name\_count;//给特定名称计数

wordinfo(){

sum=0;

name\_count=0;

}//构造函数，初始化

};

static wordinfo myDictionary[SIZE];//记录不重复名字及其信息

**算法1.1 int main()**

作 用：主控函数，也实现对题目所需其他内容的测试。

参 数：无参数。

返回值：总是返回0。

计算过程：

1. 输入一组数据。
2. 与已经存储的互不相同的名字作对比，若相同，特定姓名的计数器加1；
3. 若不同，则将其存储进结构体数组中，计数器设置为1；
4. 根据要求计算相关数据；
5. 输出结果。

## 定义函数atoi()

**模块结构及文件组织设计：**

模块1：仅包括文件main.cpp,定义了main（）函数。

模块2：仅包含task.h文件，文件对atoi()函数进行了声明。

模块3：仅包含test.cpp文件，文件对atoi()函数进行了功能实现。

**关键数据结构设计：**

const char\*str(输入的字符串)和int result返回的整数。

**算法2.1 int main()**

作 用: 主控函数 输入数据、类型转换和结果输出

参 数： 无参数

返回值： 0

计算过程：

1. 输入字符串，将字符串保存在 string str中；
2. 将str转化为常量指针 const char \*p=str.c\_str();//将输入的字符串类型转化为常量；
3. 调用算法2.2，将返回的结果赋值给result；
4. 输出结果；
5. 返回0。

**算法2.2 int atoi()**

作 用： 将字符串转化为整数，包括十进制、八进制、十六进制

参 数： const char\*str

返回值：若输入错误，返回值为0；若正确输入，则返回整数

计算过程：

1. 输入数据，判断是否输入正确；
2. 判断是否存在符号，若存在，保存符号；
3. 判断是十进制还是其他（八进制、十六进制）；
4. 进行字符串转化为整数（绝对值）；
5. 返回加上符号的整数。

## 定义1函数itoa()

**模块结构及文件组织设计：**

模块1：仅包括文件main.cpp,定义了main（）函数。

模块2：仅包含task.h文件，文件对itoa()函数进行了声明。

模块3：仅包含test.cpp文件，文件对itoa()函数进行了功能实现。

**关键数据结构设计：**

int n(输入的整数) char \*str(输出的字符串) int stack[50](堆栈数组)

**算法3.1 int main()**

作 用： 主控函数 输入数据和结果输出

参 数： 无参数

返回值： 返回0

计算过程：

1. 输入一个整数；
2. 调用函数itoa()，运算结果存储在字符串数组中；
3. 输出结果；
4. 返回0.。

**算法3.2 int itoa()**

作 用：将指定的整数转化为（十进制字符串）

参 数： int n,char\*str

返回值： 无返回值

计算过程：

1. 判断正负，若为负，转化为正数；
2. 将整数逐步转化为个位数，并存放在堆栈数组中；
3. 将每一个个位数转化为字符串。

# 测试用例与测试结果

## 5.1 计算名字的相关数据

|  |  |  |
| --- | --- | --- |
| 序号 | 测试数据 | 输出结果 |
| 1 | Xi 60  Wei 50.2  Jin 100  Wei 40.5  Wei 1.1  Jin 200.0 | 结果是：  Xi Sum=60 Avg=60  Jin Sum=91.8 Avg=30.6  Wei Sum=300 Avg=150  总计:  ALL\_Sum=41.8, ALL\_Avg=75.3 |

## 定义函数atoi()

|  |  |  |
| --- | --- | --- |
| 序号 | 测试数据 | 输出结果 |
| 1 | 123 | 123 |
| 2 | -123 | -123 |
| 3 | +123 | 123 |
| 4 | 0123 | 83 |
| 5 | -0123 | -83 |
| 6 | +0123 | 83 |
| 7 | 0x123 | 291 |
| 8 | +0x123 | 291 |
| 9 | -0X123 | -291 |
| 10 | x123 | 0 |
| 11 | 1a | 1 |
| 12 | 12.3E-10 | 12 |

## 定义函数itoa()

|  |  |  |
| --- | --- | --- |
| 序号 | 测试数据 | 输出结果（十进制） |
| 1 | 123 | 123 |
| 2 | -123 | -123 |
| 3 | 2147483647 | 2147483647 |
| 4 | -2147483648 | -2147483648 |

# 实验总结

通过本次实验，我学习了c++的基本类型、表达式和基本控制结构。在第一题中，开始是想将所有数据存储后再进行相关计算，后来改进后输入的同时就进行了一部分的计算，使代码更加简洁，逻辑更加清晰，并且掌握了判断输入结束的方法。第二题和第三题中，核心是转化的方法与异常处理，并且要考虑到多种输入情况。我们在考虑一个问题时，应考虑全面。目前，对于面向对象程序设计只是一个粗浅的理解，我们还要更多的实验加深理解。在程序设计中，我们也应该养成好的编程习惯。