中国矿业大学计算机学院实验报告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 课程名称 | 高级语言程序设计 | | 实验名称 | 高级语言程序设计实践 | |
| 班级 | 信息安全2019-1班 | 姓名 | 李春阳 | 学号 | 10193657 |
| 仪器组号 |  | | 实验日期 | 2020.12.1 | |
| 实验报告要求：1.实验目的 2.实验内容（题目描述，源代码，运行截图，调试情况） 3.实验体会 | | | | | |
| 一、实验目的  1. 认识了解c++基本语法 2. 掌握条件语句和循环语句 3. 掌握数组和字符串的表达。  二、实验内容1、第一题1.1题目描述  1. 问题描述   在进行文章重复度检查时，经常需要统计一段英文中的单词数量，并找出长度最长的单词。  设有如下定义：char str[500];  编写程序，通过利用cin.getline(str,500);实现从键盘输入一小段英文（其中可以包含空格，但在同一行），利用函数统计该段英文中包含几个单词，输出统计出的单词数量、最长单词的长度以及长度最长的单词，空格隔开。  注意：函数声明使用void split(char \*str);如果有最长的单词不只一个，输出最先找到的那个。   1. 输入   一小段英文，不要超过500个字符   1. 输出   单词数量、最长单词的长度以及长度最长的单词，空格隔开。 1.2 源代码 #include <iostream>  #include <cmath>  #include <cstring>  using namespace std;  void split(char\* str)  {  int i, j, t = 0, l[20], max = 0, m = 0;  for (i = 0; \*(str + i); i++)  {  if (\*(str + i) == ' ')  {  l[t] = i;  t++;  }  }  l[t] = i;  max = l[0];  for (j = 0; j < t; j++)  {  if (max < (l[j + 1] - l[j]))  {  max = l[j + 1] - l[j];  m = j;  }  }  cout << t + 1 << ' ' << max - 1 << ' ';  for (i = l[m] + 1; i < l[m + 1]; i++)  cout << \*(str + i);  }  int main()  {  char str[500];  cin.getline(str, 500);  split(str);  return 0;  } 1.3 运行截图  1.4 调试情况2、第二题2.1题目描述  1. 问题描述、   编程序，按如下方法求A矩阵的转置矩阵B：输入两个正整数m和n，而后通过使用指针配合new运算符生成一个m行n列的二维动态数组A以及另一个n行m列的二维动态数组B，之后为A输入数据（A矩阵数据），逐行逐列输入，进而求出其转置矩阵B（即将A中的行存放成B中的列）并输出结果，逐行逐列输出，每一行数字占一行，同一行的数空格隔开。  注意：输入保证全是整数。   1. 输入   输入m和n,逐行逐列输入数组A的数据   1. 输出   逐行逐列输出矩阵B。 2.2 源代码 #include <iostream>  #include <cmath>  #include <cstring>  using namespace std;  int main()  {  int m, n, i, j;  cin >> m >> n;  int(\*p)[100] = new int[100][100];  for (i = 0; i < m; i++)  for (j = 0; j < n; j++)  cin >> \*(\*(p + i) + j);  for (j = 0; j < n; j++)  {  for (i = 0; i < m; i++)  cout << \*(\*(p + i) + j) << ' ';  cout << endl;  }  return 0; }2.3 运行截图  2.4 调试情况3、第三题3.1题目描述  1. 问题描述   编写如下原型的函数：  void split(double x,int\*iPart,double\*fPart);  提取出数据x的整数部分与小数部分，分别放于\*iPart与\*fPart处，由于形参iPart与fPart都是指针，从而可实现将这两个结果“带回”到主函数中。  在主函数中  输入一个数  输出它的整数部分和小数部分，用空格隔开。  提示：一个double类型数，强制类型转换后就是int，也就是整数部分，差为小数部分。这两个值用指针iPart和指针fPart带回（通过修改指针的目标变量值。）   1. 输入   一个数   1. 输出   整数部分 小数部分，用空格隔开 3.2 源代码 #include <iostream>  #include <cmath>  #include <cstring>  using namespace std;  void split(double x, int\* iPart, double\* fPart)  {  int a;  double b;  a = x;  b = x - a;  \*iPart = a;  \*fPart = b;  }  int main()  {  double x, b, \* f;  int a, \* i;  cin >> x;  f = &b;  i = &a;  split(x, i, f);  cout << a << ' ' << b;  return 0;  } 3.3 运行截图  3.4 调试情况4、第四题4.1题目描述  1. 问题描述   编制具有如下原型的函数findLast：             char\*findLast(char\*sourceStr,char\*subStr);  findLast函数则要返回源串sourceStr中最后一次出现subStr子字符串的头字符位置。  而后编制主函数，输入两个字符串，将它们用作实参来调用这两个函数，如果返回NULL输出-1，否则输出子字符串出现时头字符在原字符串的下标，每个结果占一行。  要求实现程序中不可使用“string.h”头文件内有关寻找子串的标准库函数。   1. 输入   输入源串sourceStr，子字符串subStr。   1. 输出   子字符串subStr最后一次在源串sourceStr中出现的位置 4.2 源代码 #include <iostream>  #include <cmath>  #include <cstring>  using namespace std;  char\* findLast(char\* sourceStr, char\* subStr)  {  int x = 0, y = 0, i, j, t = 0;  while (\*(sourceStr + x))  x++;  x--;  while (\*(subStr + y))  y++;  y--;  for (i = x; i >= 0; i--)  {  if (\*(sourceStr + i) == \*(subStr + y))  {  t = 0;  for (j = 0; j <= y; j++)  if (\*(sourceStr + i - j) != \*(subStr + y - j))  t++;  if (t == 0)  {  return sourceStr + i - j + 1;  break;  }  }  }  return NULL;  }  int main()  {  char a[500], b[50], \* p;  cin >> a;  cin >> b;  p = findLast(a, b);  if (p == NULL)  cout << -1;  else cout << p - a;  return 0;  } 4.3 运行截图  4.4 调试情况5、第五题5.1题目描述  1. 问题描述   编写具有如下原型的函数：  int f(unsigned long x, int n,int & Lxn);  它负责将整数x的第n位（从左边数第n位，n>0）的数值放到引用Lxn中（将作为结果返回到主调函数的对应实参变量中），并将倒数第n位（从右边数第n位，n>0）的数值作为函数结果返回去。  例如当x=123456789,n=7时，执行语句Rxn=f(x,n, Lxn);将使返回的Lxn为7，并使Rxn变为3；而执行语句 Rxn=f(12345，6，Lxn)；将使Lxn与Rxn都变为0（超出数的“长度”即总位数时返回0）   1. 输入   输入x,输入n   1. 输出   输出Rxn，Lxn 5.2 源代码 #include <iostream>  #include <cmath>  #include <cstring>  using namespace std;  int lenght(unsigned long x)  {  int n = 0;  while (x != 0)  {  x /= 10;  n++;  }  return n;  }  int f(unsigned long x, int n, int& Lxn)  {  int number = x;  int i, len = lenght(x);  if (n > len)  {  Lxn = 0;  return 0;  }  else  {  Lxn = int(int(x / pow(10, len - n)) % 10);  return int(int(x / pow(10, n - 1)) % 10);  }  }  int main()  {  unsigned long x;  int n, Lxn, Rxn;  cin >> x >> n;  Rxn = f(x, n, Lxn);  cout << Rxn << " " << Lxn << endl;  return 0;  } 5.3 运行截图  5.4 调试情况三、实验体会 通过这次实验，我更加明白了c++理论中的一些编程规范和c++语言特性，掌握了基本编程知识，以后会更加认真的学习c++理论知识，并不断实践和练习，在debug中不断学习。 | | | | | |