**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 指针与引用  班级 计算机科学与技术2019-3班 姓名 王杰永 学号 03190886  仪器组号\_\_\_\_\_\_\_  实验日期 2020.11.26  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）掌握指针变量的定义和初始化，指针的间接访问，指针的加减运算和指针的表达式。  （2）掌握用指针处理数组和字符串的方法。  （3）掌握用指针作为函数参数的方法。  （4）理解引用的概念，掌握引用的应用方法。  二、实验内容  1、第一题  1.1题目描述    1.2 源代码  #include<iostream>  using namespace std;  void split(char\* str)  {  int count = 0, len = 0, max = 0;  char\* st = str, \* end = str;  while (\*end)  {  if (\*end != ' ')  len++;  if (\*end == ' ' && len > 0)  {  count++;  if (len > max)  {  max = len;  st = end - len;  }  len = 0;  }  end++;  }  if (\*(end - 1) != ' ')  count++;  if (len > max)  {  max = len;  st = end - len;  }  if (\*str == '\0')  count = 0;  cout << count << " " << max << " ";  int i = 0;  while (i < max)  {  cout << \*(st + i);  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  在零输入时有返回值，加了len=0和对输入为空的if判断语句。输入后有空格，对最后结尾count++加了条件判断。输入前有空格，加了输入为空的if判断语句。  2、第二题  2.1题目描述    2.2 源代码  #include <iostream>  #include<string>  using namespace std;  int main()  {  int m, n;  cin >> m >> n;  int\*\* p = new int\* [m];  for (int i = 0; i < m; i++)  p[i] = new int[n];  int\*\* q = new int\* [n];  for (int i = 0; i < n; i++)  q[i] = new int[m];  for (int i = 0; i < m; i++)  for (int j = 0; j < n; j++)  cin >> p[i][j];  for (int i = 0; i < m; i++)  for (int j = 0, k = 0; j < n; j++, k++) {  q[k][i] = p[i][j];  }  for (int i = 0; i < n; i++) {  for (int j = 0; j < m; j++)  cout << q[i][j] << " ";  cout << endl;  }  return 0;  }  2.3 运行截图    2.4 调试情况  掌握了动态创建二维数组的方法。先使用指向指针的指针指向开辟的一个指针数组，后用指针数组中的每一个指针指向一片新开辟的空间。  3、第三题  3.1题目描述    3.2 源代码  #include <iostream>  #include<string>  using namespace std;  void split(double x, int\* iPart, double\* fPart) {  int z = int(x);  double xiao = x - z;  \*iPart = z;  \*fPart = xiao;  }  int main()  {  double x;  cin >> x;  int\* iPart = new int;  double\* fPart = new double;  split(x, iPart, fPart);  cout << \*iPart << endl << \*fPart << endl;  return 0;  }  3.3 运行截图    3.4 调试情况  题目较为简单，编写过程顺利。  4、第四题  4.1题目描述    4.2 源代码  #include<iostream>  using namespace std;  int length(char\* str) {  int i = 0;  while (str[i] != NULL)  i++;  return i;  }  char\* findLast(char\* sourceStr, char\* subStr) {  int i = 0, len = length(subStr);  int position = -1;  while (sourceStr[i] != NULL) {  bool r = true;  for (int j = 0, k = i; j < len; j++, k++)  if (sourceStr[k] != subStr[j]) {  r = false;  break;  }  if (r)  position = i;  i++;  }  cout << position << endl;  return NULL;  }  int main()  {  char sourceStr[100], subStr[20];  cin >> sourceStr >> subStr;  findLast(sourceStr, subStr);  return 0;  }  4.3 运行截图    4.4 调试情况  感觉题目有些问题。题目说如果返回null则输出-1，即未找到则输出-1，找到了就输出下标。于是函数类型char\*似乎有些多余，可以用void代替。  （提交的代码中，无论是否找到都会返回NULL，输出下标的操作在函数体内部）  5、第五题  5.1题目描述    5.2 源代码  #include<iostream>  using namespace std;  int f(unsigned long x, int n, int& Lxn) {  int count = 0;  int z = x;  while (z) {  z = z / 10;  count++;  }  if (count < n || n == 0) {  Lxn = 0;  return 0;  }  int y = 0, i = 1;  while (x) {  int r = x % 10;  if (i == (count - n + 1))  Lxn = r;  if (i == n)  y = r;  x = x / 10;  i++;  }  return y;  }  int main()  {  unsigned long x;  int n;  cin >> x >> n;  int Lxn;  int Rxn = f(x, n, Lxn);  cout << Rxn << " " << Lxn << endl;  return 0;  }  5.3 运行截图    5.4 调试情况  找到一个数从左到右第n位，从右向左第n位，即要求从数中找出特定的两位，所以放在了一个循环中。  三、实验体会  本次实验我学会了更好的运用函数的参数来达到参数传递的效果，函数需要多个返回值时可以通过引用传递参数的方式来实现。同时熟练了new创建二维动态数组的过程。 |