实验报告

专业班级： 软件工程2205

学号：L209220532

姓名：丹尼

实验报告成绩：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验 | 实验二 | 实验三 | 实验四 | 实验五 | 总评 |
| 成绩 |  |  |  |  |  |

批阅教师：

实验四

1. 实验目的

1、进一步加深对数组的理解，掌握数组的定义方法；

2、掌握数组的处理方法、数组作为函数参数的使用方法，以及搜索与排序的应用。 3、掌握指针的概念、指针变量定义格式以及指针的运算；

4、掌握指针与数组、函数的关系；

5、理解内存动态分配的含义、熟练掌握内存动态分配方法；

6、掌握递归函数的定义方法。

1. 实验内容与要求

程序设计

(1)编写函数检查字符串 s1 是否为字符串 s2 的子串，若是，返回第一次匹配的下标，否则返 回-1。在主程序中输入字符串 s1 与 s2，调用函数实现。 函数原型：int indexof(const char \*s1,const char \*s2);

(2)编写一个函数将以字符串形式表示的一个 16 进制数转换为 10 进制数，并在主函数中测 试。函数原型 int parseHex(const char \*const hexString); 如：调用函数 parseHex(“A5”); 返回 165

(3) 主程序中建立一动态数组（使用 new），数组元素及元素个数由键盘输入，动态调试观 察指针及指针指向的内容；设计一个函数对数组由小到大排序；主程序中用指针方式输出数组元 素；最后释放数组内存（delete）

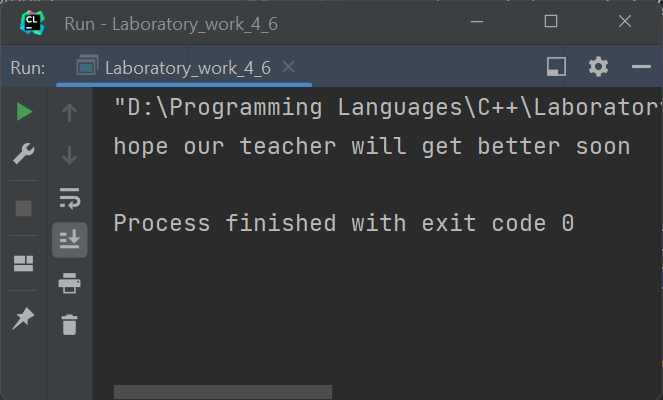
1. 实验步骤、算法与结果分析

1. 401indexof.cpp

**程序代码**

#include **<iostream>  
  
using** std::cout, std::endl;  
  
**constexpr int** NO\_MATCH = 1;  
**constexpr int** MATCH = 0;  
**constexpr char** NO\_MATCH\_STRING[] = {**"-1"**};  
  
**static** size\_t strLength(**const char** \*s) {  
 size\_t len = 0;  
 **while** (\*s != **'\0'**) {  
 s++;  
 len++;  
 }  
 **return** len;  
}  
  
*//checks if s1 and s2 are the same char by char***static bool** matchStr(**const char** \*s1, **const char** \*s2, size\_t length) {  
 **for** (**int** i = 0; i < length; i++) {  
 **char** c = \*s1++;  
 **char** c2 = \*s2++;  
 *//if chars don't exist for some reason* **if** (!c || !c2) {  
 **return** NO\_MATCH;  
 }  
 **if** (c != c2) **return** NO\_MATCH;  
 }  
 **return** MATCH;  
}  
  
**static const char** \*indexOf(**const char** \*s1, **const char** \*s2) {  
 **char** c;  
 size\_t len;  
  
 c = \*s1++;  
  
 *//if the first char doesn't exist, s1 is empty* **if** (!c)  
 **return** NO\_MATCH\_STRING;  
  
 len = strLength(s1);  
  
 **do** {  
 **char** sChar;  
  
 *//trying to find the first char of s1 in s2* **do** {  
 sChar = \*s2++;  
 *//if sChar doesn't exist, s2 is empty or at the end of iteration* **if** (!sChar)  
 **return** NO\_MATCH\_STRING;  
  
 } **while** (sChar != c);  
 *//checking if the rest of the s1 is in s2* } **while** (matchStr(s2, s1, len) != 0);  
  
 **return** (**char** \*) (s2 - 1);  
}  
  
  
**int** main() {  
  
 **constexpr char** s1[] = {**"hope"**};  
 **constexpr char** s2[] = {**"I hope our teacher will get better soon"**};  
  
 cout << indexOf(s1, s2) << endl;  
  
  
 **return** 0;  
}

**结果**

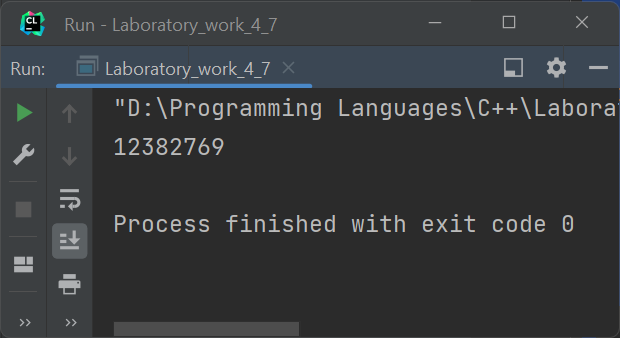


2. 402hextodec.cpp

**程序代码**

#include **<iostream>**#include **<valarray>  
  
  
constexpr int** BAD\_VALUE = -1;  
**constexpr int** hexDigits = 16;  
**constexpr char** hexDecimals[hexDigits] = {**'0'**,  
 **'1'**,  
 **'2'**,  
 **'3'**,  
 **'4'**,  
 **'5'**,  
 **'6'**,  
 **'7'**,  
 **'8'**,  
 **'9'**,  
 **'A'**,  
 **'B'**,  
 **'C'**,  
 **'D'**,  
 **'E'**,  
 **'F'**};  
  
**static** size\_t strLength(**const char** \*s) {  
 size\_t len = 0;  
 **while** (\*s != **'\0'**) {  
 s++;  
 len++;  
 }  
 **return** len;  
}  
  
*//returns a number raised to a power***static int** powerOf(**int** number, **int** power) {  
 **if** (power == 0) {  
 **return** 1;  
 }  
 **int** value = number;  
 **for** (**int** it = 1; it < power; it++, value \*= number);  
 **return** value;  
}  
  
*//if char value is hex, function returns a corresponding int value***static int** isHex(**char** value) {  
  
 **for** (**int** it = 0; it < hexDigits; it++) {  
 **if** (value == hexDecimals[it]) **return** it;  
 }  
 **return** BAD\_VALUE;  
}  
  
**static int** parseHex(**const char** \***const** hexString) {  
  
 **int** result = 0;  
 size\_t length = strLength(hexString);  
  
 *//checking every element of the hexString* **for** (**int** it = 0; it < length; it++) {  
 **int** value = isHex(hexString[it]);  
  
 *//if char isn't a hex digit return -1* **if** (value == BAD\_VALUE) **return** BAD\_VALUE;  
  
 result += (value \* powerOf(16, length - it - 1));  
 }  
  
 **return** result;  
}  
  
  
**int** main() {  
  
  
 std::cout << parseHex(**"BCF231"**) << std::endl;  
  
 **return** 0;  
}

**结果**

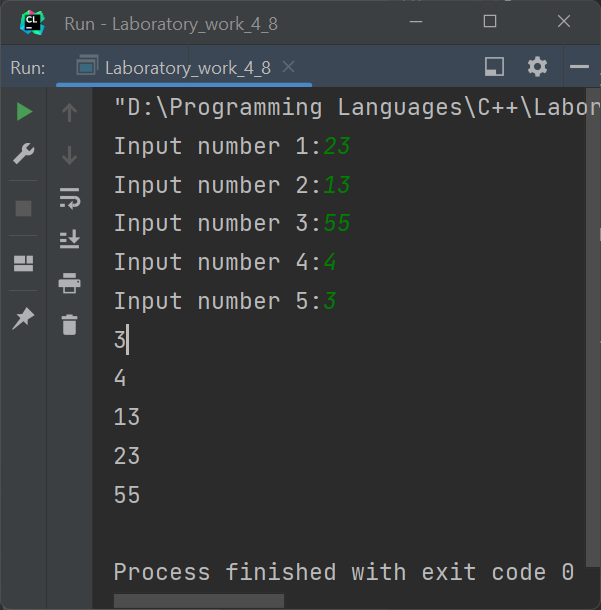
****

3. 403dynamicarray

**程序代码**

#include **<iostream>  
  
using** std::cout, std::cin, std::endl;  
  
  
**static void** sortArray(**int** \*array, **int** size) {  
  
 **bool** changed;  
 **do** {  
 changed = **false**;  
 **for** (**auto** j = 0; j < size - 1; j++)  
 **if** (array[j] > array[j + 1]) {  
 **auto** temp = array[j];  
 array[j] = array[j + 1];  
 array[j + 1] = temp;  
 changed = **true**;  
 }  
 } **while** (changed);  
}  
  
**static int** getNumber(**int** position) {  
  
 **int** number = -1;  
  
 printf(**"Input number %d:"**, position + 1);  
 cin >> number;  
  
 **while** (cin.fail()) {  
 cin.clear();  
 cin.ignore();  
 printf(**"Input number %d:"**, position + 1);  
 cin >> number;  
 };  
  
 **return** number;  
}  
  
**int** main() {  
  
  
 **constexpr int** arraySize = 5;  
  
 **auto** \*array = **new int**[arraySize];  
  
 **for** (**auto** it = 0; it < arraySize; it++) {  
  
 array[it] = getNumber(it);  
  
 }  
  
  
 sortArray(array, arraySize);  
  
 **int** \*pArray = array;  
  
 **for** (**auto** it = 0; it < arraySize; it++) {  
 cout << \*array++ << endl;  
 }  
  
 **delete**[] pArray;  
  
 **return** 0;  
}

**结果**

****

四，遇到的问题与解决办法

实验过程中没有遇到任何问题。

五，体会

During this laboratory work I had a chance to practice working with pointers and arrays in C++

Though pointers create an additional level of complicity, they are useful, make programs faster and more lightweight. There may be lots of data stored in arrays and since stack has it limitations, pointers are used to dynamically allocate data on the heap, that’s why it is a common practice to access data in arrays with pointers. Arrays are great to store and manage data in a comprehensive to a programmer way.