

# 实 验 报 告

课程名称： 软件技术基础

院系名称： 电气与信息工程学院

专业班级： 电子信息工程 22-1

学生姓名： 邱禹尧

学 号： 20221699

指导教师： 王 岩

**黑龙江工程学院教务处制**

# 实 验 报 告

实验项目	函数的应用				
同组人数	1 人	实验地点	实验楼 506	实验日期	2023.4.7
实验类型	<input type="checkbox"/> 验证型 <input type="checkbox"/> 综合型 <input checked="" type="checkbox"/> 设计型 <input type="checkbox"/> 其 它				
<p>一、实验目的</p> <ol style="list-style-type: none"> <li>1、熟悉定义函数的方法。</li> <li>2、熟悉声明函数的方法。</li> <li>3、熟悉调用函数时实参与形参的对应关系，以及“值传递”的方式。</li> <li>4、学习对多文件的程序的编译和运行。</li> <li>5、熟悉怎样利用函数实现指定的任务。</li> <li>6、熟悉函数的嵌套调用和递归调用的方法。</li> <li>7、熟悉全局变量和局部变量的概念和用法。</li> </ol>					
<p>二、实验器材</p> <p>MinGW VSCode 笔记本电脑</p>					
<p>三、实验内容（原理、方案、步骤、记录及分析等）</p>					

C 20221699邱禹尧素数.c × C 超大数.c

C 20221699邱禹尧素数.c > main()

```
4  int main()
5  {
6      int is_prime(int n);
7      int n;
8      do
9      {
10         printf("请输入整数: ");
11         scanf("%d", &n);
12         if (is_prime(n))
13             printf("%d是素数.\n", n);
14         else
15             printf("%d不是素数.\n", n);
16     } while (n != -1);
17     return 0;
18 }
19 int is_prime(int n)
20 {
21     if (n <= 1)
22         return 0;
23     int sqrt_n = sqrt(n);
24     for (int i = 2; i <= sqrt_n; i++)
25     {
26         if (n % i == 0)
27             return 0;
28     }
29     return 1;
30 }
```

1.

(1)

```

PS C:\Users\10670\Desktop\untitled\output> cd 'c:\Users\10670\Desktop\untitled\output'
PS C:\Users\10670\Desktop\untitled\output> & .\'20221699邱禹尧素数.exe'
请输入整数: 17
17是素数。
请输入整数: 34
34不是素数。
请输入整数: 2
2是素数。
请输入整数: 1
1不是素数。
请输入整数: 0
0不是素数。

```

结果正确

(2)

```

20221699邱禹尧素数.c:11:13: warning: implicit declaration of function 'is_prime' [-Wimplicit-function-declar
    if (is_prime(n))
        ^~~~~~

```

编译器给出警告，提示此函数隐式声明

(3)正常编译，提示编译成功

(4)

```

int main()
{
    int i;
    for (i = 100; i <= 200; i++)
    {
        if (is_prime(i))
            printf("%d ", i);
    }
    return 0;
}

```

```

PS C:\Users\10670\Desktop\untitled\output> & .\'20221699邱禹尧素数.exe'
101 103 107 109 113 127 131 137 139 149 151 157 163 167 173 179 181 191 193 197 199

```

2.

```

#include <stdio.h>
#include <string.h>
void copy_vowels(char s[], char c[])
{
    unsigned int j = 0;
    for (unsigned int i = 0; i < strlen(s); i++)
    {
        if (s[i] == 'a'
            || s[i] == 'e'
            || s[i] == 'i'
            || s[i] == 'o'
            || s[i] == 'u')
        {
            c[j] = s[i];
            j++;
        }
    }
    c[j] = '\0';
    printf("复制后的字符串: %s\n", c);
}
int main()
{
    char s[100], c[100];
    printf("请输入字符串\n");
    scanf("%s", s);
    copy_vowels(s, c);
    return 0;
}

```

```

PS C:\Users\10670\Desktop\untitled\output> & .\'20221699邱禹尧元音.exe'
请输入字符串
test
复制后的字符串: e

```

- (1) 结果成功输出了元音字母
- (2) 结果无任何区别

数组名本身就是数组在内存中的地址。在函数调用时，实参通过数组名传递给函数，函数可以使用这个地址来访问数组的元素，因此不需要知道数组的具体大小。

- (3)无所谓，数组大小在实参中就会被指定，与形参无关

```

#include <stdio.h>

#define NUM_STUDENTS 10
#define NUM_SUBJECTS 5

void get_scores(int scores[NUM_STUDENTS][NUM_SUBJECTS]) {
    int i, j;
    for (i = 0; i < NUM_STUDENTS; i++) {
        printf("输入%d号学生得分: ", i + 1);
        for (j = 0; j < NUM_SUBJECTS; j++) {
            scanf_s("%d", &scores[i][j]);
        }
    }
}

// 计算每个学生的平均分
void print_student_avg(int scores[NUM_STUDENTS][NUM_SUBJECTS]) {
    int i, j, sum;
    float avg;
    for (i = 0; i < NUM_STUDENTS; i++) {
        sum = 0;
        for (j = 0; j < NUM_SUBJECTS; j++) {
            sum += scores[i][j];
        }
        avg = (float)sum / NUM_SUBJECTS;
        printf("%d号学生平均分为: %.2f\n", i + 1, avg);
    }
}

// 计算每门课的平均分
void print_subject_avg(int scores[NUM_STUDENTS][NUM_SUBJECTS]) {
    int i, j, sum;
    float avg;
    for (j = 0; j < NUM_SUBJECTS; j++) {
        sum = 0;
        for (i = 0; i < NUM_STUDENTS; i++) {
            sum += scores[i][j];
        }
        avg = (float)sum / NUM_STUDENTS;
        printf("%d号学科平均分为: %.2f\n", j + 1, avg);
    }
}

//找出所有 50 个分数中最高的分数所对应的学生和课程。
void find_highest_score(int scores[NUM_STUDENTS][NUM_SUBJECTS]) {
    int i, j, max_score = -1, max_i, max_j;
    for (i = 0; i < NUM_STUDENTS; i++) {
        for (j = 0; j < NUM_SUBJECTS; j++) {
            if (scores[i][j] > max_score) {
                max_score = scores[i][j];
                max_i = i;
                max_j = j;
            }
        }
    }
    printf("最高分为%d (%d号学生,%d号课程)\n", max_score, max_i + 1, max_j + 1);
}

int main() {
    int scores[NUM_STUDENTS][NUM_SUBJECTS];
    get_scores(scores);
    print_student_avg(scores);
    print_subject_avg(scores);
    find_highest_score(scores);
    return 0;
}

```

3.

(1)(2)(3)如图所示

```

输入3号学生得分: 123 21354 34 2345 123
输入4号学生得分: 4123 234123 213 123132 421
输入5号学生得分: 2313 2134 3421 2321 32213
输入6号学生得分: 123123 342 123231 325 12334
输入7号学生得分: 12313 4332 12343 2312 21334
输入8号学生得分: 213235 21442 21331 32412 123
输入9号学生得分: 124124 21332 231 231 234
输入10号学生得分: 12421
23
23
124
1245
1号学生平均分为: 28644.60
2号学生平均分为: 185.20
3号学生平均分为: 4795.80
4号学生平均分为: 72402.40
5号学生平均分为: 8480.40
6号学生平均分为: 51871.00
7号学生平均分为: 10526.80
8号学生平均分为: 57708.60
9号学生平均分为: 29230.40
10号学生平均分为: 2767.20
1号学科平均分为: 49313.10
2号学科平均分为: 30743.90
3号学科平均分为: 17317.10
4号学科平均分为: 28673.70
5号学科平均分为: 7258.40
最高分为234123 (4号学生,2号课程)

```

4.(1)

20221699邱禹尧分数.c 单词函数.c

ConsoleApplication1

```

3 void print_longest_word(char* str)
4 {
5     char word[100];
6     char longest_word[100];
7     int word_length = 0;
8     int longest_word_length = 0;
9     int i = 0, j = 0;
10    while (str[i] != '\0')
11    {
12        if (isalpha(str[i]))
13        {
14            word[j++] = str[i];
15            word_length++;
16        }
17        else
18        {
19            if (word_length > longest_word_length)
20            {
21                longest_word_length = word_length;
22                strcpy_s(longest_word, sizeof(longest_word), word, word_length);
23            }
24            j = 0;
25            word_length = 0;
26        }
27        i++;
28    }
29
30    if (word_length > longest_word_length)
31    {
32        longest_word_length = word_length;
33        strcpy_s(longest_word, sizeof(longest_word), word, word_length);
34    }
35
36    longest_word[longest_word_length] = '\0';
37    printf("The longest word is: %s\n", longest_word);
38 }
39
40 void bubble_sort(char* arr[], int n) {
41     int i, j;
42     char* temp;
43
44     for (i = 0; i < n - 1; i++) {
45         for (j = 0; j < n - i - 1; j++) {
46             if (strcmp(arr[j], arr[j + 1]) > 0) {
47                 temp = arr[j];
48                 arr[j] = arr[j + 1];
49                 arr[j + 1] = temp;
50             }
51         }
52     }
53 }
54
55 int main() {
56     char str[10][100];
57     char* ptr[10];
58     int i;
59     printf("请输入10个字符串: \n");
60     for (i = 0; i < 10; i++) {
61         scanf_s("%s", str[i], 100);
62         ptr[i] = str[i];
63     }
64     bubble_sort(ptr, 10);
65     printf("排序后的结果为: \n");
66     for (i = 0; i < 10; i++) {
67         printf("%s ", ptr[i]);
68     }
69     return 0;
70 }

```

(2)

```
20221699邱禹尧分数.c 20221699邱禹尧单词.c 单词函数.c X
ConsoleApplication1
4 void print_longest_word(char* str)
5 {
6     char word[100];
7     char longest_word[100];
8     int word_length = 0;
9     int longest_word_length = 0;
10    int i = 0, j = 0;
11    while (str[i] != '\0')
12    {
13        if (isalpha(str[i]))
14        {
15            word[j++] = str[i];
16            word_length++;
17        }
18        else
19        {
20            if (word_length > longest_word_length)
21            {
22                longest_word_length = word_length;
23                strcpy_s(longest_word, sizeof(longest_word), word, word_length);
24            }
25            j = 0;
26            word_length = 0;
27        }
28        i++;
29    }
30    if (word_length > longest_word_length)
31    {
32        longest_word_length = word_length;
33        strcpy_s(longest_word, sizeof(longest_word), word, word_length);
34    }
35    longest_word[longest_word_length] = '\0';
36    printf("The longest word is: %s\n", longest_word);
37 }
38
39
40
```

```
20221699邱禹尧分数.c 20221699邱禹尧单词.c X
ConsoleApplication1
1 #include <stdio.h>
2 #include <string.h>
3
4
5 void bubble_sort(char* arr[], int n) {
6     int i, j;
7     char* temp;
8
9     for (i = 0; i < n - 1; i++) {
10        for (j = 0; j < n - i - 1; j++) {
11            if (strcmp(arr[j], arr[j + 1]) > 0) {
12                temp = arr[j];
13                arr[j] = arr[j + 1];
14                arr[j + 1] = temp;
15            }
16        }
17    }
18 }
19
20 int main() {
21     char str[10][100];
22     char* ptr[10];
23     int i;
24     printf("请输入10个字符串: \n");
25     for (i = 0; i < 10; i++) {
26         scanf("%s", str[i]);
27         ptr[i] = str[i];
28     }
29     bubble_sort(ptr, 10);
30     printf("排序后的结果为: \n");
31     for (i = 0; i < 10; i++) {
32         printf("%s ", ptr[i]);
33     }
34     return 0;
35 }
```

5.(1)

```
20221699邱禹尧素数.c C 20221699邱禹尧元音.c C 20221699起泡法.c
20221699起泡法.c > bubble_sort(char *[], int)
1 #include <stdio.h>
2 #include <string.h>
3 void bubble_sort(char *arr[], int n) {
4     int i, j; char *temp;
5     for (i = 0; i < n - 1; i++) {
6         for (j = 0; j < n - i - 1; j++) {
7             if (strcmp(arr[j], arr[j + 1]) > 0) {
8                 temp = arr[j];
9                 arr[j] = arr[j + 1];
10                arr[j + 1] = temp;
11            }
12        }
13    }
14 }
15
16 int main() {
17     char str[10][100]; char *ptr[10]; int i;
18     printf("请输入10个字符串: \n");
19     for (i = 0; i < 10; i++) {
20         scanf("%s", str[i]);
21         ptr[i] = str[i];
22     }
23     bubble_sort(ptr, 10);
24     printf("排序后的结果为: \n");
25     for (i = 0; i < 10; i++) {
26         printf("%s ", ptr[i]);
27     }
28     return 0;
29 }
```



请输入5个字符串：

asd  
adfaw  
adfscx  
asdfewa  
xcsad  
asdfcx  
asdwqad  
asfasdf  
xzcsad  
wqeres

排序后的结果为：

adfaw adfscx asd asdfcx asdfewa asdwqad asfasdf wqeres xcsad xzcsad

```
20221699邱禹尧素数.c 20221699邱禹尧元音.c 20221699起泡法.c
20221699起泡法.c > main()
1  #include <stdio.h>
2  #include <string.h>
3  void bubble_sort(char *arr[], int n) {
4      int i, j; char *temp;
5      for (i = 0; i < n - 1; i++) {
6          for (j = 0; j < n - i - 1; j++) {
7              if (strcmp(arr[j], arr[j + 1]) < 0) {
8                  temp = arr[j];
9                  arr[j] = arr[j + 1];
10                 arr[j + 1] = temp;
11             }
12         }
13     }
14 }
15 int main() {
16     char str[5][100]; char *ptr[5]; int i;
17     printf("请输入5个字符串:\n");
18     for (i = 0; i < 5; i++) {
19         scanf("%s", str[i]);
20         ptr[i] = str[i];
21     }
22     bubble_sort(ptr, 5);
23     printf("排序后的结果为: \n");
24     for (i = 0; i < 5; i++) {
25         printf("%s ", ptr[i]);
26     }
27     return 0;
28 }
```

(2)

PS C:\Users\10670\Desktop\untitled\output> & .\'20221699起泡法.exe'

请输入5个字符串：

asdfw  
asdwads  
csadcsa  
dsfaw  
asdcxs

排序后的结果为：

dsfaw csadcsa asdwads asdfw asdcxs

```
C 20221699邱禹尧递归.c × C 20221699邱禹尧素数.c
C 20221699邱禹尧递归.c > main()

3 void int_to_str(int n, char* str) {
4     if (n < 0) {
5         *str++ = '-';
6         n = -n;}
7     if (n < 10) {
8         *str++ = n + '0';
9         *str = '\0';
10        return;}
11    int_to_str(n / 10, str);
12    while (*str) {
13        str++;}
14    *str++ = (n % 10) + '0';
15    *str = '\0';
16    }
17    int main() {
18        int n;
19        printf("请输入数字");
20        scanf("%d", &n);
21        char str[20];
22        int_to_str(n, str);
23        printf("%s\n", str);
24        return 0;
25    }
26

问题 输出 调试控制台 终端
50468
PS C:\Users\10670\Desktop\untitled\output>
```

6.(1)

结果正确

(2)

递归调用是一种函数调用自身的方式，可以在某些情况下方便地解决复杂的问题。

特点：

1. 递归调用可以使问题分解成更小的子问题，这些子问题与原问题具有相同的结构，因此可以使用相同的算法解决。
2. 递归调用通常需要使用额外的空间来存储函数调用栈。每次函数调用都会在栈上创建一个新的栈帧，存储函数的参数和局部变量。在函数返回时，该栈帧将被销毁，释放所占用的空间。
3. 递归调用可能导致栈溢出，特别是当递归调用的深度非常大时。这是因为每个栈帧都需要占用一定的空间，而栈的大小是有限制的。当栈的空间用尽时，程序将崩溃。
4. 递归调用需要注意递归边界条件的正确性。如果递归边界条件不正确，函数可能会无限递归下去，导致程序崩溃。

(3) 能

C 20221699邱禹尧递归.c

C 20221699非递归.c X

C 20221699非递归.c > ...

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  void int_to_str(int n, char* str) {
4      if (n == 0) {
5          *str++ = '0';
6          *str = '\0';
7          return;}
8      int sign = n < 0 ? -1 : 1;
9      n = abs(n);
10     char buf[20];
11     int len = 0;
12     while (n > 0) {
13         buf[len++] = n % 10 + '0';
14         n /= 10;}
15     if (sign < 0) {
16         *str++ = '-';}
17     while (len > 0) {
18         *str++ = buf[--len];}
19     *str = '\0';
20 }
21
22 int main() {
23     int n = 483;
24     char str[20];
25     int_to_str(n, str);
26     printf("%s\n", str);
27     return 0;
28 }
```

```
PS C:\Users\10670\Desktop\untitled\output> & .\'20221699非递归.exe'
483
PS C:\Users\10670\Desktop\untitled\output> █
```

#### 四、实验结论

## 五、思考题（见实验指导书思考题）

已在前文中写过

## 六、实验心得

包括：

- (1) 试验中遇到的问题及解决方法；
- (2) 本次实验的收获，你的能力有那方面的提升？

七、实验情况及成绩评定		
	预    习:	<input type="checkbox"/> 优秀 <input type="checkbox"/> 一般 <input type="checkbox"/> 不及格
	出    席:	<input type="checkbox"/> 正常 <input type="checkbox"/> 迟到 <input type="checkbox"/> 缺席 <input type="checkbox"/> 早退 <input type="checkbox"/> 事假
	过程表现:	<input type="checkbox"/> 优秀 <input type="checkbox"/> 一般 <input type="checkbox"/> 不及格
	完成报告:	<input type="checkbox"/> 按时 <input type="checkbox"/> 迟交
	实验结论	<input type="checkbox"/> 正确 <input type="checkbox"/> 基本正确 <input type="checkbox"/> 错误
	思考题回答情况	<input type="checkbox"/> 正确 <input type="checkbox"/> 基本正确 <input type="checkbox"/> 错误
	心得体会	<input type="checkbox"/> 优秀 <input type="checkbox"/> 一般 <input type="checkbox"/> 不及格
	成绩评定:	
补充记录或评语:	教师签字:	