



計算機程序設計基礎（C）

實驗報告

學生姓名 許采婷

學 號 8208230832

專業班級 計通 2308 班

指導教師 楊迎澤

學 院 計算機學院

实验四 函数实验

1. 实验目的要求

- (1) 了解函数的声明定义和调用，函数形式参数和实际参数的关系和应用。
- (2) 了解程序的基本模块，掌握函数设计、嵌套调用与递归调用的方法。
- (3) 能将函数与数组等知识联系一起进行综合应用，解决较复杂的工程实际问题。

2. 实验内容

2.1 教材第9章编程题第1题：编写选择排序的函数，并递归调用。

说明：本题利用函数简化程序、使得结构更清晰、熟练掌握。

(1) 实验运行结果：

```
Enter integers: 1 6 5 9 3 4 8 2 7
Integers arranged in order from smallest to largest: 1 2 3 4 5 6 7 8 9
```

(2) 源代码：

```
// 编写选择排序的函数，并递归调用

#include <stdio.h>

void selection_sort(int arr[], int n) {
    int i, j, max_index, temp;

    if (n == 1) {
        return;
    }
    for (i=0; i<n-1; i++) {
        max_index = 0;
        for (j=1; j<n-i; j++) {
            if (arr[j] > arr[max_index])
                max_index = j;
        }

        temp = arr[max_index];
        arr[max_index] = arr[n - i - 1];
        arr[n - i - 1] = temp;
    }

    selection_sort(arr, n - 1);
}

int main() {
    int arr[100], n = 0, i;

    printf("Enter integers: ");
```

```

while (1) {

    int num;
    if (scanf("%d", &num) == EOF)
        break;

    arr[n++] = num;

    if (getchar() == '\n')
        break;
}

selection_sort(arr, n);
printf("Integers arranged in order from smallest to
largest: ");

for (i = 0; i < n; i++)
    printf("%d ", arr[i]);

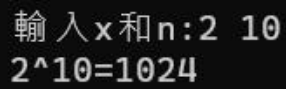
return 0;
}

```

2.2 教材第9章编程题第7题：编写计算 x^n 的函数。

说明：本题考查函数定义、调用，并初步考察递归运用。

(1) 实验运行结果：



```
輸入x和n:2 10
2^10=1024
```

(2) 源代码：

```
// 編寫計算  $x^n$  的函數

#include <stdio.h>

int power(int x, int n){
    if (n == 0)
        return 1;

    else if (n % 2 == 0){
        int result = power(x, n / 2);
        return result * result;
    }

    else
        return x * power(x, n - 1);
}

int main(){
    int x, n;

    printf("輸入 x 和 n:");
    scanf("%d %d", &x, &n);
    printf("%d^%d=%d", x, n, power(x, n));

    return 0;
}
```

2.3 教材第 10 章编程题第 6 题：编写程序实现逆波兰运算。

说明：本题考查对程序每部分的理解、掌握，外部、内部变量，主函数，小函数的编写与应用。初步了解栈的一些实现函数。

(1) 实验运行结果：

```
Enter an RPN expression: 1 2 3 * + =  
Value of expression: 7  
Enter an RPN expression: 5 8 * 4 9 - / =  
Value of expression: -8  
Enter an RPN expression: q  
Invalid expression
```

(2) 源代码：

```
#include <stdbool.h>  
#define STACK_SIZE 100  
  
int contents[STACK_SIZE];  
int top = 0;  
  
void make_empty(void){  
    top = 0;  
}  
  
bool is_empty(void){  
    return top == 0;  
}  
  
bool is_full(void){  
    return top == STACK_SIZE;  
}  
  
void push(int i){  
    if (is_full())  
        stack_overflow();  
    else  
        contents[top++] = i;  
}  
  
int pop(void){  
    if(is_empty()){  
        stack_underflow();  
        return 0;  
    }  
    else  
        return contents[--top];  
}
```

```

void stack_overflow(void){

    printf("Expression is too complex\n");
    exit(EXIT_FAILURE);
}

void stack_underflow(void){

    printf("Not enough operands in expression\n");
    exit(EXIT_FAILURE);
}

int main(){

    char ch;
    int op1, op2;

    for (;;) {
        printf("Enter an RPN expression: ");
        for (;;) {
            scanf(" %c", &ch);
            if (ch == '=') {
                printf("Value of expression: %d\n", pop());
                break;
            }
            else if (ch >= '0' && ch <= '9') {
                push(ch - '0');
            }
            else if (ch == '+' || ch == '-' || ch == '*' || ch == '/') {
                op2 = pop();
                op1 = pop();
                switch (ch) {
                    case '+': push(op1 + op2); break;
                    case '-': push(op1 - op2); break;
                    case '*': push(op1 * op2); break;
                    case '/':
                        if (op2 == 0) {
                            printf("Division by zero\n");
                            make_empty();
                            break;
                        } else {
                            push(op1 / op2);
                            break;
                        }
                }
            }
            else if (ch == ' ') {
                continue;
            }
            else if (ch == '\n') {
                break;
            }
            else if (ch == '=') {
                break;
            }
            else {
                printf("Invalid expression\n");
                make_empty();
            }
        }
    }
}

```

```
        break;
    }
}
if (ch != '=') {
    break;
}
}
return 0;
}
```

2.4 编程序找出 2~5000 中满足条件的数 x：x 是素数且 x 的每位数字是素数、每位数字之和是素数，如 23、223、2333 等。

说明：本题考查函数方法与循环选择结构的综合运用。

(1) 实验运行结果：

```
2
3
5
7
23
223
227
337
353
373
557
577
733
757
773
2333
2357
2377
2557
2753
2777
3253
3257
3323
3527
3727
```

(2) 源代码：

```
// 2~5000 中满足条件的数 x：x 是素数且每位数字是素数、每位数字之和是素数

#include <stdio.h>
#include <stdbool.h>

bool is_prime(int n){
    if(n <= 1)
        return false;

    for(int i=2; i*i<=n; i++) {
        if (n%i==0)
            return false;
    }

    return true;
}
```



```

}

bool is_prime_digit(int n){ // 每位數字是素數

    int digit;

    while(n>0){

        digit = n%10;

        if (!is_prime(digit))
            return false;

        n = n/10;
    }

    return true;
}

bool is_prime_digit_sum(int n){ // 每位數字之和是素數

    int sum = 0;

    while(n>0){

        sum = sum + n%10;
        n = n/10;
    }

    return is_prime(sum);
}

int main(){

    for (int i=2; i<=5000; i++){

        if (is_prime(i) && is_prime_digit(i) &&
is_prime_digit_sum(i)){

            printf("%d\n", i);
        }
    }

    return 0;
}

```