

Assignment Chapter 6

张理凯 202300101051

1.

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

#define MAX 100

void sieveOfEratosthenes(int n) {
    bool prime[n+1];
    for (int i = 0; i <= n; i++) {
        prime[i] = true;
    }

    for (int p = 2; p * p <= n; p++) {
        if (prime[p] == true) {
            for (int i = p * p; i <= n; i += p) {
                prime[i] = false;
            }
        }
    }

    for (int p = 2; p <= n; p++) {
        if (prime[p]) {
            printf("%d ", p);
        }
    }
    printf("\n");
}

int main() {
    int n = MAX;
    printf("Prime numbers from 1 to %d are:\n", n);
    sieveOfEratosthenes(n);
    return 0;
}
```

Prime numbers from 1 to 100 are:

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

4.

```
#include <stdio.h>

void insertSorted(int arr[], int n, int num) {
    int i;
    for (i = n - 1; (i >= 0 && arr[i] > num); i--) {
        arr[i + 1] = arr[i];
    }
    arr[i + 1] = num;
}

int main() {
    int arr[100] = {1, 3, 5, 7, 9}; // Example sorted array
    int n = 5; // Number of elements in the array
    int num;

    printf("Enter a number to insert: ");
    scanf("%d", &num);

    insertSorted(arr, n, num);
    n++;

    printf("Array after insertion: ");
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }

    printf("\n");
    return 0;
}
```

```
Enter a number to insert: 15
Array after insertion: 1 3 5 7 9 15
```

10.

为了简便，每行的读入改为不大于80个字符，统计出各统计量。

```
#include <stdio.h>
#include <ctype.h>

int main() {
    char text[3][81]; // 3 lines, each with 80 characters + 1 for null
    terminator
    int upper[3] = {0}, lower[3] = {0}, digits[3] = {0}, spaces[3] = {0},
    others[3] = {0};

    printf("Please enter 3 lines of text (each line up to 80
    characters):\n");
    for (int i = 0; i < 3; i++) {
```

```

        fgets(text[i], 81, stdin);
    }

    for (int i = 0; i < 3; i++) {
        for (int j = 0; text[i][j] != '\0'; j++) {
            if (isupper(text[i][j])) {
                upper[i]++;
            } else if (islower(text[i][j])) {
                lower[i]++;
            } else if (isdigit(text[i][j])) {
                digits[i]++;
            } else if (isspace(text[i][j])) {
                spaces[i]++;
            } else {
                others[i]++;
            }
        }
    }

    for (int i = 0; i < 3; i++) {
        printf("Line %d: %d upper case, %d lower case, %d digits, %d spaces, %d others\n", i + 1, upper[i], lower[i], digits[i], spaces[i], others[i]);
    }

    return 0;
}

```

Please enter 3 lines of text (each line up to 80 characters):
apple.
base..
Hello World!%%%

Line 1: 0 upper case, 5 lower case, 0 digits, 1 spaces, 1 others
Line 2: 0 upper case, 4 lower case, 0 digits, 1 spaces, 2 others
Line 3: 2 upper case, 8 lower case, 0 digits, 2 spaces, 4 others

14.

由于gets函数在读取字符串时没有边界检查，可能导致缓冲区溢出，所以在C11中被废除。于是在这段代码中改用fgets。

```

#include <stdio.h>
#include <string.h>

int compare_strings(const char *s1, const char *s2) {
    while (*s1 && (*s1 == *s2)) {
        s1++;
        s2++;
    }
}

```

```

        return *((unsigned char *)s1) - *((unsigned char *)s2);
    }

int main() {
    char s1[100], s2[100];

    printf("Enter first string: ");
    fgets(s1, sizeof(s1), stdin);

    printf("Enter second string: ");
    fgets(s2, sizeof(s2), stdin);

    s1[strcspn(s1, "\n")] = 0;
    s2[strcspn(s2, "\n")] = 0;

    int result = compare_strings(s1, s2);
    if (result == 0) {
        printf("The strings are equal.\n");
    } else {
        printf("%d\n", result);
    }

    return 0;
}

```

```

Enter first string: And
Enter second string: Aid
5

```

15.

```

#include <stdio.h>

void copyString(char *s1, const char *s2) {
    while ((*s1++ = *s2++) != '\0') {
        // Copy each character from s2 to s1, including the null terminator
    }
}

int main() {
    char s1[100];
    char s2[] = "Hello, World!";

    copyString(s1, s2);

    printf("s1: %s\n", s1);

    return 0;
}

```

```
s1: Hello, World!
```