实训六 数据类型构造与模块化程序设计——数组的构造与应用

第1题

题目：从键盘输入20 个整型数据，统计其中正数的个数，并计算它们的求和。

```c
#include <stdio.h>

int main()
{
    int a[20];
    int cnt, sum;

    /* 1. 输入 */
    for (int i = 0; i < 20; ++i)
    {
        scanf("%d", &a[i]);
    }

    /* 2. 依次判定是否是正数 */
    cnt = 0;
    sum = 0;
    for (int i = 0; i < 20; ++i)
    {
        if (a[i] > 0)
        {
            ++cnt;
            sum += a[i];
        }
    }

    /* 3. 输出结果 */
    printf("正数个数为%d, 这些正数的和为%d\n", cnt, sum);

    return 0;
}
```

第2题

题目：把 1000 以内的素数存放在数组中，并输出素数的个数和各个素数。

```c
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <math.h>

/* 判断是否为素数的函数 */
bool is_prime(int val)
```

```
{
    if (val <= 1)
    {
        return false;
    }
    int stop = sqrt(val);
    for (int i = 2; i <= stop; i++)
    {
        if (0 == val % i)
        {
            return false;
        }
    }
    return true;
}

int main(int argc, char const *argv[])
{
    int primes[1000];
    int cnt = 0;

    /* 1. 一次判断是否是素数，并加入到 primes 数组中 */
    for (int i = 2; i <= 1000; i++)
    {
        if (is_prime(i))
        {
            primes[cnt] = i;
            ++cnt;
        }
    }

    /* 2. 输出结果 */
    printf("素数个数：%d\n", cnt);
    for (int i = 0; i < cnt; ++i)
    {
        printf("%d ", primes[i]);
    }
    printf("\n");

    return 0;
}
```

第3题

在第一题的基础上找出最大数和最小数并输出对应的下标。

```
#include <stdio.h>

int main()
{
    int a[20];
    int max_id, min_id;

    /* 1. 输入数据 */
    for (int i = 0; i < 20; ++i)
    {
        scanf("%d", &a[i]);
```

```
    }

    /* 2. 寻找最大值和最小值的位置 */
    max_id = 0;
    min_id = 0;
    for (int i = 1; i < 20; ++i)
    {
        if (a[i] > a[max_id])
        {
            max_id = i;
        }
        if (a[i] < a[min_id])
        {
            min_id = i;
        }
    }

    /* 3. 输出结果 */
    printf("min_value: a[%d] = %d\n", min_id, a[min_id]);
    printf("max_value: a[%d] = %d\n", max_id, a[max_id]);

    return 0;
}
```

第4题

任意输入10 个数据，对其进行排序（用选择法小到大）。

解析： 选择排序，不想再说了，自己看代码吧。

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

#define N 10

int main()
{
    int a[10];

    /* 1. 输入 */
    for (int i = 0; i < N; ++i)
    {
        scanf("%d", &a[i]);
    }

    /* 2. 排序 */
    for (int left = 0; left <= N - 2; left++)
    {
        int min_id = left;
        for (int i = left + 1; i < N; i++)
        {
            if (a[i] < a[min_id])
            {
                min_id = i;
            }
        }
        int t     = a[left];
```

```
        a[left]   = a[min_id];
        a[min_id] = t;
    }

    /* 3. 输出 */
    for (int i = 0; i < N; ++i)
    {
        printf("%d ", a[i]);
    }
    printf("\n");

    return 0;
}
```

第5题

在第4 题的基础上，从键盘上接收一个数据，如果该数不存在，把该数
按照顺序放在数组中，若存在则把和该数相等的元素删除。

解析： 首先是选择排序，排序完了之后输入一个数字 `val` ，然后从前往后找，找到 `>= val` 的。然后判断如果是相等则删除，如果不等则插入：

1. **删除操作**：把后面的整体往前挪。
2. **插入操作**：把后面的整体往后挪，然后插入数据。

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

#define N 10


void print_arr(int *a, int n)
{
    for (int i = 0; i < n; i++)
    {
        printf("%d ", a[i]);
    }
    printf("\n");
}


int main()
{
    int a[20];
    int val;

    /* 1. 输入 */
    for (int i = 0; i < N; ++i)
    {
        scanf("%d", &a[i]);
    }

    /* 2. 排序 */
    for (int left = 0; left <= N - 2; left++)
    {
        int min_id = left;
        for (int i = left + 1; i < N; i++)
```

```
        {
            if (a[i] < a[min_id])
            {
                min_id = i;
            }
        }
        int t      = a[left];
        a[left]    = a[min_id];
        a[min_id] = t;
    }

    /* 3. 输出排序后 */
    printf("排序后的数列： ");
    print_arr(a, N);
    printf("\n");

    /* 4. 再输入一个数字 */
    scanf("%d", &val);
    printf("输入的数字为：%d\n", val);

    for (int i = 0; i < N; i++)
    {
        if (a[i] == val)
        {
            /* 找到相同的数字，执行删除操作。 */
            printf("输入的数字在数列中找到相同的数字，执行删除操作。\n");
            for (int j = i + 1; j < N; j++)
            {
                a[j-1] = a[j];
            }
            print_arr(a, N - 1);
            return 0;
        }

        if (a[i] > val)
        {
            /* 找到比 val 大的数字，执行插入操作。 */
            printf("输入的数字在数列中并不存在，执行插入操作。\n");
            for (int j = N - 1; j >= i; j--)
            {
                a[j+1] = a[j];
            }
            a[i] = val;
            print_arr(a, N + 1);
            return 0;
        }
    }

    /* 如果能执行到此处，说明整个数组里面都没有找到一个比 val 大的，则在末尾增加 val */
    a[N] = val;
    printf("输入的数字是全局最大的。");
    printf("输入的数字在数列中并不存在，执行插入操作。\n");
    print_arr(a, N + 1);

    return 0;
}
```

第6题

某学习小组有4 名同学，学习了5 门课程，编程求出最高分和最低分及
其对应的行号和列号。

解析： 很容易看出这题需要用到一个二维数组来存储这4名同学的成绩。首先需要声明一个二维数组 `int a[4][5];` 接下来找最大值和最小值的位置的方法就照着一维数组的方法来。先假设 `a[0][0]` 是最大/最小的。然后依次比较，如果发现更大/更小的，则记录位置即可。

```c
#include <stdio.h>

int main()
{
    int a[4][5];

    /* 1. 输入数据 */
    for (int i = 0; i < 4; i++)
    {
        for (int j = 0; j < 5; j++)
        {
            scanf("%d", &a[i][j]);
        }
    }

    int max_i, max_j;
    int min_i, min_j;

    /* 2. 寻找最大值和最小值的位置 */
    max_i = 0;
    max_j = 0;
    min_i = 0;
    min_j = 0;
    for (int i = 0; i < 4; i++)
    {
        for (int j = 0; j < 5; j++)
        {
            if (a[i][j] > a[max_i][max_j])
            {
                max_i = i;
                max_j = j;
            }
            if (a[i][j] < a[min_i][min_j])
            {
                min_i = i;
                min_j = j;
            }
        }
    }

    /* 3. 输出最小值和最大值的位置以及值 */
    printf("min_value: a[%d][%d] = %d\n", min_i, min_j, a[min_i][min_j]);
    printf("max_value: a[%d][%d] = %d\n", max_i, max_j, a[max_i][max_j]);

    return 0;
}
```