

lab6

PB21111618 侯超群

- 1.purpose
使用一种高级编程语言实现lab1~lab4编写的所有代码；同时保证使用的算法与之前一致；
- 2.principles
 - lab1
 - 利用test_bit从低位至高位测试a，根据b值进行循环；
 - 每次test_bit与a进行与运算后乘2，即左移；
 - lab2
 - 利用F，FP，FPP储存F(N)，F(N-1)，F(N-2)，根据n值进行循环；
 - 通过不断减p或q至负数，后再加上一次实现取模运算；
 - lab3
 - length记录查找时的相同字符长度，Max_length记录已查找的字符中最大长度，在查找过程中不断更新；
 - 指针从头开始向后移动，每次与相邻的字符比较；
 - lab4
 - 首先对其进行排序，两个指针构成两个循环的嵌套，逐个从后面选择最小的交换到前面；
 - 对排序后的数组进行统计A，B类学生；
- 3.procedure

```
#include <stdint>
#include <iostream>
#include <fstream>

#define MAXLEN 100
#ifndef LENGTH
#define LENGTH 3
#endif

int16_t lab1(int16_t a, int16_t b)
{
    // initialize
    int16_t output = 0, test_bit = 1;
    // calculation
    for (; b > 0; b--)
    {
        if ((a & test_bit) != 0)
        { // 测试位为1
            output++;
        }
        test_bit = test_bit + test_bit;
    }
    // return value
    return output;
}
```

```
}

int16_t lab2(int16_t p, int16_t q, int16_t n)
{
    // initialize
    int16_t F = 1, FP = 1, FPP = 1; // F(N), F(N-1), F(N-2)
    // calculation
    for (; n > 1; n--)
    {
        FPP = FP;
        FP = F;
        int16_t fp = FP, fpp = FPP;
        while (fp > 0)
            fp -= q;
        fp += q;
        while (fpp > 0)
            fpp -= p;
        fpp += p;
        F = fpp + fp;
    }
    // return value
    return F;
}

int16_t lab3(int16_t n, char s[])
{
    // initialize
    int16_t Max_length = 1, length = 1;
    char *p = s;
    // calculation
    for (; n > 1; n--)
    {
        if (*p == *(p + 1))
        {
            length++;
        }
        else
        { // 初始化length
            length = 1;
        }
        if (length > Max_length)
            Max_length = length;
        p++;
    }
    // return value
    return Max_length;
}

void lab4(int16_t score[], int16_t *a, int16_t *b)
{
    // initialize
    int16_t *p = score;
    *a = *b = 0;
    // calculation
```

```
for (int i = 0; i < 16; i++)
{
    int16_t min = 100, min_location = 0;
    for (int j = i + 1; j < 16; j++)
    {
        if (*(p + j) < min)
        {
            min = *(p + j);
            min_location = j;
        }
    }
    if (min < *(p + i))
    { // 交换最小值
        *(p + min_location) = *(p + i);
        *(p + i) = min;
    }
}

for (int i = 15; i > 11; i--)
{
    if (*(p + i) >= 85)
        (*a)++;
    else if (*(p + i) >= 75)
        (*b)++;
}

for (int i = 11; i > 7; i--)
{
    if (*(p + i) >= 75)
        (*b)++;
}
}

int main()
{
    std::fstream file;
    file.open("test.txt", std::ios::in);

    // lab1
    int16_t a = 0, b = 0;
    for (int i = 0; i < LENGTH; i++)
    {
        file >> a >> b;
        std::cout << lab1(a, b) << std::endl;
    }

    // lab2
    int16_t p = 0, q = 0, n = 0;
    for (int i = 0; i < LENGTH; i++)
    {
        file >> p >> q >> n;
        std::cout << lab2(p, q, n) << std::endl;
    }

    // lab3
    char s[MAXLEN];
```

```


for (int i = 0; i < LENGTH; i++)
{
    file >> n >> s;
    std::cout << lab3(n, s) << std::endl;
}

// lab4
int16_t score[16];
for (int i = 0; i < LENGTH; i++)
{
    for (int j = 0; j < 16; j++)
    {
        file >> score[j];
    }
    lab4(score, &a, &b);
    for (int j = 0; j < 16; j++)
    {
        std::cout << score[j] << " ";
    }
    std::cout << std::endl
                << a << " " << b << std::endl;
}

file.close();
return 0;
}

```

- 4.result of your test

 选择 D:\Codefiled\CODE_C++\C++_Single\exe\lab6.exe

```

2
4
15
146
818
1219
3
4
3
0 10 20 25 30 35 40 45 50 55 60 80 85 90 95 100
4 1
0 10 15 20 25 35 40 45 50 65 70 75 80 90 95 100
3 2
9 10 11 21 22 33 44 53 55 57 66 77 88 97 98 99
4 1
请按任意键继续. . .

```

- 5.answer to the question

- 高级编程语言相对于汇编语言简化了许多命令，同时对于数据的范围减少了限制，更便于编写者操作；
- 比如取模操作，以及对于两数之间的比较实现减法操作
- 学到了从更底层去了解编程语言，更清晰地实现高级语言的相关操作；

