

图文作业1

1. 请简述**C/C++**程序开发的主要步骤及其内容。（略，参考教材或者课件）

图文作业2

4.用**switch**语句实现：从键盘输入一个整数，输出南京大学的各个书院名称（次序自定，并注意验证课件中所说的**switch**语句的注意事项）。

```
int main() {
    int n;
    scanf("%d", &n);
    switch (n) {
        case 1: printf("开甲书院\n"); break;
        case 2: printf("建雄书院\n"); break;
        case 3: printf("有训书院\n"); break;
        case 4: printf("安邦书院\n"); break;
        case 5: printf("秉文书院\n"); break;
        case 6: printf("毓秀书院\n"); break;
        case 7: printf("行知书院\n"); break;
        default: printf("false\n"); break;
    }
    return 0;
}
```

图文作业2

5. 分析下面两个程序片段功能是否等价。

```
int main() {  
    for (int i = 1; i < 5; ++i)  
        for (int j = 1; j < 5; ++j)  
            for (int k = 1; k < 5; ++k)  
                if (i == k && i == j && j == k)  
                    printf("%d, %d, %d \n", i,  
j, k);  
    return 0;  
}
```

运行结果

1, 1, 1
2, 2, 2
3, 3, 3
4, 4, 4

图文作业2

5. 分析下面两个程序片段功能是否等价。

```
int main() {  
    int i = 1, j = 1, k = 1;  
    while (i < 5) {  
        while (j < 5) {  
            while (k < 5) {  
                if (i == k && i == j && j == k)  
                    printf("%d, %d, %d \n", i,  
j, k);  
                ++k;  
            }  
            ++j;  
        }  
        ++i;  
    }  
    return 0;  
}
```

运行结果: 1, 1, 1

后续执行的时候，**k**和**j**的值变掉了，导致不再循环

图文作业2

5. 分析下面两个程序片段功能是否等价。

```
int main() {  
    int i = 1, j = 1, k = 1;  
    while (i < 5) {  
        j = 1;  
        while (j < 5) {  
            k = 1;  
            while (k < 5) {  
                if (i == k && i == j && j == k)  
                    printf("%d, %d, %d \n", i, j, k);  
                ++k;  
            }  
            ++j;  
        }  
        ++i;  
    }  
    return 0;  
}
```

运行结果

1, 1, 1

2, 2, 2

3, 3, 3

4, 4, 4

图文作业3

2. 不用 **goto** 或 **continue**，改写下面的程序片段，实现等价的功能。

```
int main() {
    int d, sum = 0, i = 1;
    while (i <= 10) {
        scanf("%d", &d);
        if (d <= 0)
            continue;
        sum += d;
        ++i;
    }
    printf("%d ", sum);
    return 0;
}
```

```
int main() {
    int d, sum = 0, i = 1;
    while (i <= 10) {
        scanf("%d", &d);
        if (d > 0)
            sum += d;
        ++i;
    }
    printf("%d ", sum);
    return 0;
}
```

图文作业3

2. 不用 **goto** 或 **continue** ， 改写下面的程序片段， 实现等价的功能。

```
int main() {  
    int d, sum = 0, i = 1;  
    while (i <= 10) {  
        scanf("%d", &d);  
        if (d <= 0)  
            continue;  
        sum += d;  
        ++i;  
    }  
    printf("%d ", sum);  
    return 0;  
}
```

```
int main() {  
    int d, sum = 0;  
    for (int i = 1; i <= 10; i++) {  
        cin >> d;  
        if (d > 0) {  
            sum += d;  
        }  
    }  
    cout << sum;  
}
```

图文作业3

4. (流程控制方法的应用) 设计C/C++程序，输出一个如下形式的日历，其中的汉字原样输出，年、月由用户输入。

2024 年 10 月

	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

提示：可按公式 $iWeek = ((c/4)-2c+y+(y/4)+(26(iMonth+1)/10)+iDay-1) \% 7$ 将某年（iYear）某月（iMonth）的第一日（iDay）换算成星期（iWeek）。

公式中：c=iYear/100，c>15；y=iYear%100；当iMonth为1时，iYear要改为iYear-1，iMonth要改为13；当iMonth为2时，iYear要改为iYear-1，iMonth要改为14；当iWeek<0时，iWeek要改为iWeek+7；iWeek为0表示星期日。

图文作业3

```
int iMonth, iYear, c, y, n, iDay;
int iWeek;
scanf("%d %d", &iYear, &iMonth);
printf("%d年%d月\n", iYear, iMonth);

if (iMonth == 1 || iMonth == 3 || iMonth == 5 || iMonth == 7 || iMonth == 8 || iMonth
== 10 || iMonth == 12) {
    n = 31;
} else if (iMonth == 4 || iMonth == 6 || iMonth == 9 || iMonth == 11) {
    n = 30;
} else {
    if ((iYear % 4 == 0 && iYear % 100 != 0) || (iYear % 400 == 0)) {
        n = 29;
    } else {
        n = 28;
    }
}
```

图文作业3

```
if (iMonth == 1) {
    iYear = iYear - 1;
    iMonth = 13;
}
if (iMonth == 2) {
    iYear = iYear - 1;
    iMonth = 14;
}

c = iYear / 100;
y = iYear % 100;

iDay = 1;
iWeek = ((c / 4) - 2 * c + y + (y / 4) + (13 * (iMonth + 1) / 5) + iDay - 1) % 7;

if (iWeek < 0) {
    iWeek = iWeek + 7;
}
```

图文作业3

```
if (iWeek > 0) {
    for (int j = 0; j < iWeek; j++) {
        printf("    ");
    }
}

for (int k = 1; k <= n; k++) {
    if (iWeek % 6 == 0 && iWeek != 0) {
        printf("%-2d\n", k);
    } else {
        printf("%-2d  ", k);
    }

    iWeek++;
    if (iWeek > 6) {
        iWeek = 0;
    }
}
```

图文作业4

3. 继续完善日历程序：添加用按键增减月份，以便实现日历自动翻月的功能，用多个函数实现。

说明：开发环境一般会提供getche、_getche、getch、_getch等库函数用来捕获键盘功能区（不是符号区）的箭头键，getch、_getch读取字符不会回显在屏幕上，该类库函数一般需要调用两次：第一次调用返回值为224时说明捕获到一个功能区的键，进一步第二次调用判断其返回值为72/80时对应向上/下箭头键。调用前一般需加 #include <stdlib.h> 或 #include <conio.h> 微软官方文档的说明是：The _getche and _getwche functions read a single character from the console with echo, meaning that the character is displayed at the console. None of these functions can be used to read CTRL+C. When reading a function key or an arrow key, each function must be called twice; the first call returns 0 or 0xE0, and the second call returns the actual key code.

getch不带回显；getche回显输入的字符。需要调用两次！

图文作业4 (Windows系统)

```
void Calendar(int yrs, int mon);
```

```
int main() {  
    int year, month;  
    printf("依次输入年月: ");  
    scanf_s("%d%d", &year, &month);  
    Calendar(year, month);  
}
```

while一直循环即可，不要使用goto

```
while (1) {  
    char key = _getch();  
    key = _getch();  
    if (key == 72) {  
        month = month - 1;  
        if (month <= 0) {  
            month = month + 12;  
            year = year - 1;  
        }  
        Calendar(year, month);  
    }  
    else if (key == 80) {  
        month = month + 1;  
        if (month > 12) {  
            month = month - 12;  
            year = year + 1;  
        }  
        Calendar(year, month);  
    }  
}  
  
return 0;  
}
```

图文作业4 (Mac系统)

```
void Calendar(int yrs, int mon);
```

```
int main() {
```

```
    int year, month;
```

```
    printf("Please input the year and month: ");
```

```
    scanf("%d %d", &year, &month);
```

```
    initscr();
```

```
    cbreak();
```

```
    noecho();
```

```
    keypad(stdscr, TRUE);
```

```
    Calendar(year, month);
```

```
    getch();
```

```
    getch();
```

```
while (1) {
```

```
    int key = getch();
```

```
    if (key == 258) {
```

```
        month = month - 1;
```

```
        if (month <= 0) {
```

```
            month = month + 12;
```

```
            year = year - 1;
```

```
        }
```

```
        Calendar(year, month);
```

```
    }
```

```
    else if (key == 259) {
```

```
        month = month + 1;
```

```
        if (month > 12) {
```

```
            month = month - 12;
```

```
            year = year + 1;
```

```
        }
```

```
        Calendar(year, month);
```

```
    }
```

```
}
```

```
endwin();
```

```
return 0;
```

```
}
```

图文作业5

1. (递归) 编写递归函数，实现阿克曼 (Ackermann) 函数的计算，并在 main 函数中输入整数 m 、 n (取值范围 $[0, 4)$)，输出 Ackermann 函数值。Ackermann 函数的值随参数快速递增，用于可计算理论领域，其计算方法为：

$$A(m, n) = \begin{cases} n+1 & m=0 \\ A(m-1, 1) & m>0, n=0 \\ A(m-1, A(m, n-1)) & m>0, n>0 \end{cases}$$

```
int A(int m, int n) {
    if (m == 0)
        return n + 1;
    if (m > 0 && n == 0)
        return A(m - 1, 1);
    if (m > 0 && n > 0)
        return A(m - 1, A(m, n - 1));
}
```

```
int main() {
    int m, n;
    cin >> m >> n;
    cout << A(m, n);
    return 0;
}
```

图文作业5

1. (递归) 疫情期间W同学去某游戏公司实习，被安排参与开发一款**Naruto**同人系列益智类的游戏，来帮助疫情在家的人们舒缓心情。这款游戏中，一共有 **n** 个怪兽，而玩家每一次可以选择以下三种技能进行打怪：

千鸟：每次能打掉**1**只怪兽

螺旋手里剑：如果剩余的怪兽数 **n** 能被 **2** 整除，那么才能使用且可以打掉 **$n/2$** 只怪兽。

地爆天星：如果剩余怪兽数 **n** 能被 **3** 整除，那么才能使用且可以打掉 **$2*(n/3)$** 只怪兽。

每次玩家只能从以上 **3** 种技能中选择一种技能发动，技能发动次数越少，最后玩家得到的奖励越多。

现在为了得到最多的奖励，请你设计相关程序（可能需要使用递归函数来做）计算，打掉所有 **n** 个怪兽最少需要发动多少次技能。

提示1：你需要理解递归函数的执行过程。

提示2：既然每一次都最多有三种选择，那么可以选择其中递归结果最优的作为返回的值，注意，为了不漏解，每一层的三种选择都可能导致最优解，需要比较。

输入 **n** 表示怪兽的数量（ **$1 \leq n \leq 100$** ）输出**1**个整数，表示最少需要发动的技能次数。

图文作业5

```
int func(int n) {
    if (n == 1)
        return 1;
    int result = func(n - 1) + 1;
    if (n % 2 == 0)
        result = min(result, func(n / 2) + 1);
    if (n % 3 == 0)
        result = min(result, func(n / 3) + 1);
    return result;
}

int main() {
    int n, result;
    scanf("%d", &n);
    result = func(n);
    printf("%d", result);
}
```