

Time: 7月2日

昨日作业

1、正五角星：

```
1 int main(int argc, char* argv[])
2 {
3     for (int i = 0; i < 20; i++)
4     {
5         for (int j = 0; j < 50; j++)
6         {
7             if ( j == -13 * i / 15 + 24
8             || j == 18 * i / 19 + 24
9             || i == 6 && j % 2 == 0
10            || j == 42 * i / 13 - 21
11            || j == -42 * i / 13 + 70)
12            {
13                printf("* ");
14            }
15            else
16            {
17                printf(" ");
18            }
19        }
20        printf("\n");
21    }
22    return 0;
23 }
```

上课内容

1、求位数的方法：

(1) 公式使用：

因为 $10^y = x$ ，所以 $y = \log_{10} x$ ，那么 y 为一个实数，记为

$$y = n + 0.d_1 d_2 d_3 \dots d_k \dots$$

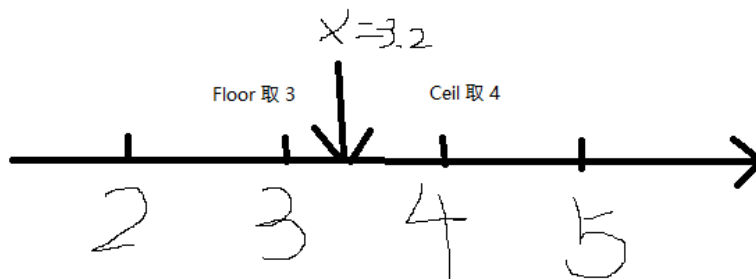
(2) 方法使用:求 (N x M) 的对数?

$$\log N + \log M$$

(3) C语言使用的函数

Ceil 上取整 (求不小于X的最小整数)

Floor 下取整 (求不大于X的最大整数)



方程:

```
1 #include <math.h>
2
3 int main(int argc, char* argv[])
4 {
5     //求100阶乘的位数
6     int n = 100;
7     double dbBitCount = 0.0;
8     for (int i = 1; i <= n; i++)
9     {
10        dbBitCount = dbBitCount + log10(i);
11    }
12    printf("%d\r\n", (int)ceil(dbBitCount));
13    return 0;
14 }
```

2. 如何传递计算结果:

(1)赋值 (包括加减乘除等运算的赋值) : 将计算结果赋值给变量

(2)传参: 传参也是有传递结果的功能

(3)传返回值: 这个也有传递结果的功能

3. goto Next \NEXT: 的使用(goto模拟三种循环)

goto 使用领域:

使用循环嵌套的时候, goto可以直接退出所有循环。break只能跳出一层。

4. 摘要法求素数

求100以内的质数, 整理10以内

(1)将每个数除以该数之前到1之间所有的数

(2)因为所有的数都是由0~9组成, 所以只要将要求素数的区间所有2~9的倍数的数去除, 剩余的就是素数

求10000以内的质数, 整理100以内

5. 递归与循环的优缺点

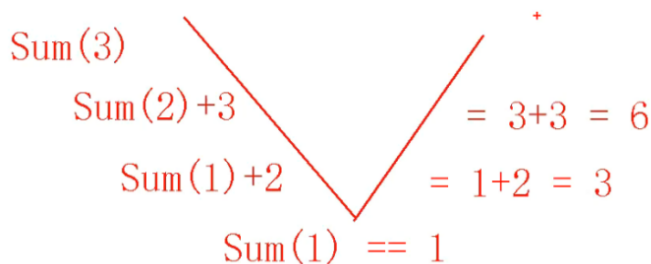
递归: 擅长解决非线性类问题

循环: 擅长解决线性类问题

注: 什么是线性?

数据的关系有唯一的前驱和唯一的后续, 前一个值和后一个值都确定。

6. 递归的传递过程



7. 菲不拉基数列 (黄金分割的算法之一)

//1 1 2 3 5 8 13

方法一 (数组+循环解决):

```
1 int main(int argc, char* argv[])
2 {
3     unsigned int aryFib[50] = { 1, 1 };
4     for (int i = 2; i < 42; i++)
5     {
6         aryFib[i] = aryFib[i - 1] + aryFib[i - 2];
    }
```

```

7  printf("%u\t%f", aryFib[i], (double)aryFib[i - 1] / aryFib[i]);
8  }
9  // \t -->tab键
10 system("pause");
11 return 0;
12 }

```

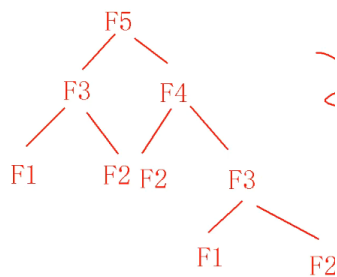
方法二（循环）：

```

1  int main(int argc, char* argv[])
2  {
3      unsigned int nFib1 = 1;
4      unsigned int nFib2 = 1;
5      unsigned int nFib3 = 1;
6      for (int i = 2; i < 42; i++)
7      {
8          nFib3 = nFib1 + nFib2;
9          printf("%02d:%15u\t%f\r\n", i-2, nFib3, (double)nFib2 / nFib3);
10         nFib1 = nFib2;
11         nFib2 = nFib3;
12     }
13     system("pause");
14     return 0;
15 }

```

方法三（递归）：



```

1  int GetFib(int n)
2  {
3      if (n == 1 || n == 2)
4      {
5          return 1;
6      }
7      return GetFib(n - 2) + GetFib(n - 1);

```

```

8  }
9
10 int main(int argc, char* argv[])
11 {
12     unsigned int nFib1 = 1;
13     unsigned int nFib2 = 1;
14     unsigned int nFib3 = 1;
15     for (int i = 2; i < 42; i++)
16     {
17         nFib1 = GetFib(i - 1);
18         nFib2 = GetFib(i);
19         printf("%02d:%15u\t%f\r\n", i-2, nFib2, (double)nFib1 / nFib2);
20     }
21     system("pause");
22     return 0;
23 }

```

课堂小知识:

- 1、 $1+2+3+\dots+n == (1+n)*n/2$
- 2、如何解决访问超过4G的内容?