

FEEL THE MEANING OF THE TRIP

青 / 春 / 不 / 老 / 梦 / 想 / 永 / 在

DREAM

MY DREAM WILL NEVER STOP

高级语言程序设计

实验四 程序测试&模块化程序设计

GO!
TAKE YOU ON A TRIP



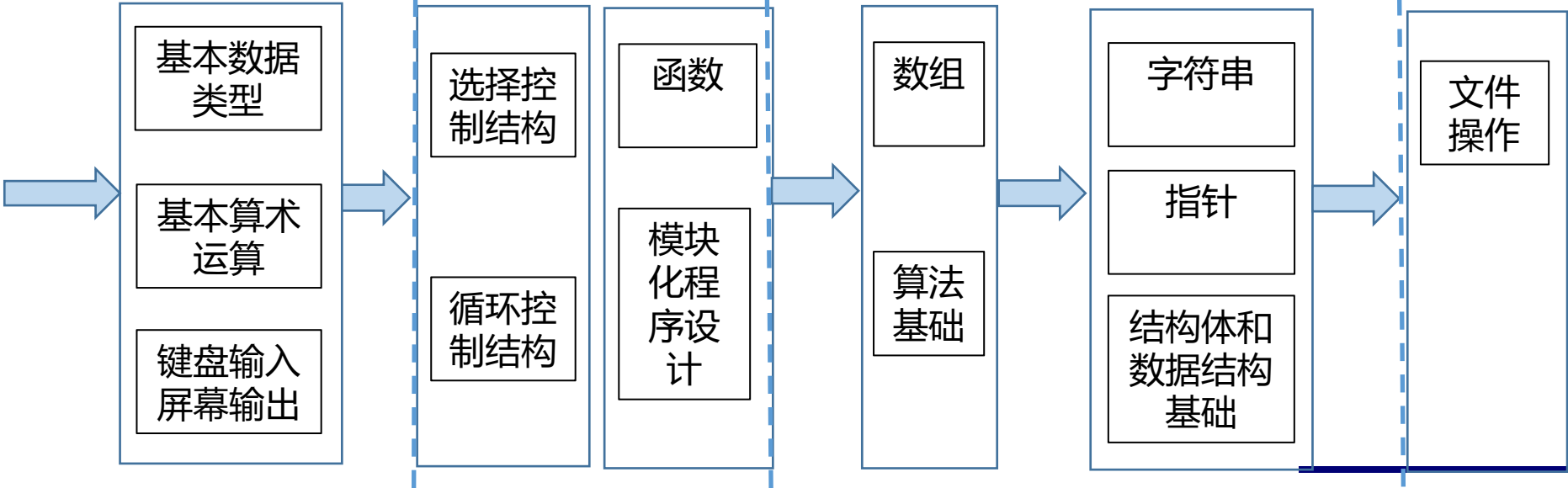
哈爾濱工業大學(深圳)
HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

探索 从未停止



实验课程安排与考核标准

编号	1	2	3	4	5	6	7	8	9	10
学时数	2	2	2	2	2	2	2	2	2	2
实验项目	初级编程	算数运算	循环控制	模块化程序设计	数组算法	字符串二维数组	指针	结构体	文件管理	
分数	1	2	2	2	3	3	3	3	3	
授课内容	集成开发环境	程序调试	编码规范	程序测试				软件文档	期末检查1	期末检查2



目录

CONTENT

01

实验回顾

02

实验目的

03

程序测试

04

实验内容

捉虫时刻

【改错题】 使用程序调试方法，改正下面程序中的错误，并简单描述调试的过程。

找出200以内的所有完全数，并输出其因子。

```
#include <stdio.h>
int main()
{
    /*****
    代码功能：找出200以内的所有完全数，并输出其真因子
    *****/
    int i,j,s=1;
    for(i=2;i<=200;i++)
    {
        for(j=2;j<=i/2;j++)
        {
            if(i%j==0){
                s = s + i;
            }
        }
        if(s==i)
        {
            printf("%d=1",i);
            for(j=2;j<=i/2;j++)
            {
                if(i%j==0)
                {
                    printf("+%d",j);
                }
            }
            printf("\n");
        }
    }
    return 0;
}
```

完全数：如果一个数恰好等于它的真因子（即除了自身以外的约数）之和，则称其为完全数。

例如：6=1+2+3

1、2、3为6的真因子，且6恰好也是其真因子之和。

输出格式：

6=1+2+3

28=1+2+4+7+14



实验回顾

➤ 数组的定义和初始化

- 固定长度: 用宏常量或者const常量定义, 程序编译时计算长度
 - `int a[5] = {1,2,3,4,5}; int a[5] = {1,2}; int a[] = {1,2,3,4,5}`
 - `int b[3][4] = {{1,2},{3,4,5},{6}}; int b[][3] = {1,2,3,4,5,6,7,8,9}`
- 可变长度 (C99): 程序执行时计算长度, 不能做初始化
 - `scanf("%d" ,&n); int score[n];`

➤ 数组的使用

- 数组下标从0开始, M为数组长度, 下标范围为[0, M-1]。

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

int main()
{
    int a[5];
    int b[5] = {1,2};
    int c[3][4] = {{1,2},{3,4,5},{6}};

    printf(" array a:\n");
    for(int i=0; i<5; i++)
    {
        printf("%d ",a[i]);
    }

    printf("\n\n array b:\n");
    for(int i=0; i<5; i++)
    {
        printf("%d ",b[i]);
    }

    printf("\n\n array c:\n");
    for(int i=0; i<3; i++)
    {
        for(int j=0; j<4; j++)
        {
            printf("%d ",c[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

D:\code\c\20

array a:
0 0 0 0 8

array b:
1 2 0 0 0

array c:
1 2 0 0
3 4 5 0
6 0 0 0

Process return
Press any key



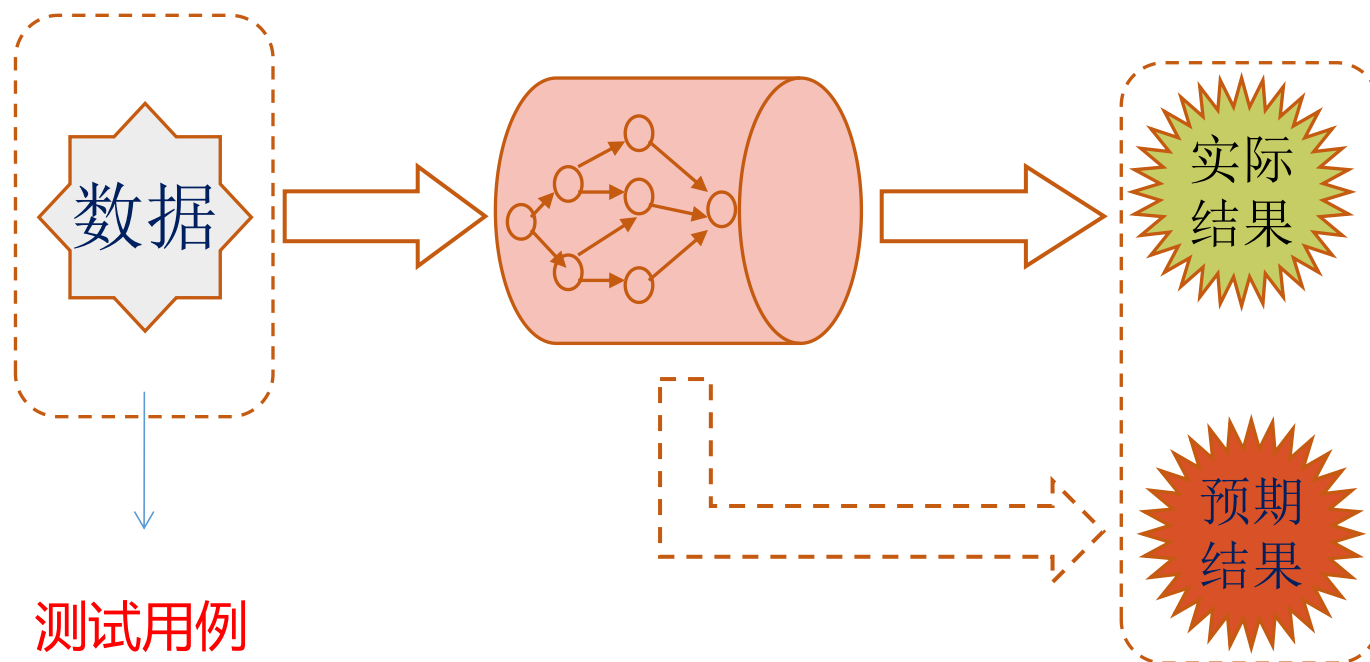
实验目的

- 熟悉函数设计、模块化程序设计的方法；
 - 掌握解决问题的两种方式：递归和迭代；
 - 熟悉常用的程序测试方法和测试用例的设计。
-



程序测试

- 确保程序质量的一种有效手段，测试的过程实质是发现错误的过程
- 目的：为了尽可能多地发现程序中的错误





程序测试

白盒测试

主要用于测试的早期

完全了解程序内部的逻辑结构和处理过程，按照程序内部的逻辑测试程序，检验程序中的每条逻辑路径是否都能按预定要求正确工作。

黑盒测试

主要用于测试的后期

不考虑程序内部的逻辑结构和处理过程，把系统看成一个黑盒子，只根据需求规格说明书的要求，设计测试用例，检查程序的功能是否符合它的功能说明。



黑盒测试：等价类划分

- 确定有效等价类和无效等价类
- 将无穷多的数据缩减到有限个等价区域中，通过测试等价区域完成穷尽测试。

身体质量指数 $t = \text{体重 } W / (\text{身高 } H)^2$

当 $t < 18$ 时，为低体重；
当 t 介于18（包含18）和25之间时，为标准体重；
当 t 介于25（包含25）和27之间时，为超重体重；
当 $t \geq 27$ 时，为肥胖。

从键盘输入你的身高 $H(m)$ 和体重 $W(kg)$ ，根据上述给定的公式计算 t ，然后判断你的体重属于何种类型。

编号	类别	数据集	输入	期望结果
1	有效等价类	$t < 18$	W: 46.24 H: 1.7	低体重
2	有效等价类	$t \geq 18 \& t < 25$	W: 66.47 H: 1.7	标准体重
3	有效等价类	$t \geq 25 \& t < 27$	W: 75.14 H: 1.7	超重体重
4	有效等价类	$t \geq 27$	W: 80.92 H: 1.7	肥胖
5	无效等价类	W超出范围	W=0, W=1000, W=-1	体重范围在[1,300kg]
6	无效等价类	H超出范围	H=0, H=15, H=-1	身高范围在(0,3m]
7	无效等价类	W不是数字	W: abc	提示需要输入数字
8	无效等价类	H不是数字	H: 12e	提示需要输入数字



黑盒测试 - 边界值分析

- 确定各种边界情况
- 选择系统边界或边界附近的数据来设计测试用例

基本原理

经过长期的测试工作经验表明，大量的缺陷经常发生在输入域或输出域的边界上。因此设计一些测试用例，使程序运行在边界情况附近，这样揭露错误的可能性比较大。

编号	类别	数据集	输入	期望结果
9	边界值	t=18	W: 52.02 H: 1.7	标准体重
10	边界值	t=25	W: 72.25 H:1.7	超重体重
11	边界值	t=27	W:78.03 H:1.7	肥胖

身体质量指数 $t =$
体重 $W/(\text{身高}H)^2$

当 $t < 18$ 时，为低体重；
当 t 介于18（包含18）和25之间时，为标准体重；
当 t 介于25（包含25）和27之间时，为超重体重；
当 $t \geq 27$ 时，为肥胖。

从键盘输入你的身高 $H(m)$ 和体重 $W(kg)$ ，根据上述给定的公式计算 t ，然后判断你的体重属于何种类型。



程序测试

核心代码

```
float t, w, h;  
scanf( "%f%f" , &h, &w);  
t = w / (h * h);
```

```
if ( t < 18)  
    printf( "低体重" );
```

```
else if ( t > 18 && t < 25)  
    printf( "标准体重" );
```

```
else if ( t > 25 && t < 27)  
    printf( "超重体重" );
```

```
else  
    printf( "肥胖" );
```

测试用例

身高H (m)	体重W (kg)	质量指数t	体型（期望结果）
1.7	46.24	16	低体重
1.7	66.47	23	标准体重
1.7	75.14	26	超重体重
1.7	80.92	28	肥胖
1.7	52.02	18	标准体重
1.7	72.25	25	超重体重
1.7	78.03	27	肥胖
-1	50		输入错误，身高在0-3m之间 （不包含0m）
1	1000		输入错误，体重在1-300kg 之间
12e	40		非法输入，请用数字表示 身高
1.6	abc		非法输入，请用数字表示 体重

捉虫时刻

【改错题】使用程序调试方法，改正下面程序中的错误。

根据下式求 π 值，直到某一项小于 10^{-6} 。

$$\frac{\pi}{2} = 1 + \frac{1!}{3} + \frac{2!}{3 \times 5} + \frac{3!}{3 \times 5 \times 7} + \frac{4!}{3 \times 5 \times 7 \times 9} + \dots + \frac{n!}{3 \times 5 \times \dots \times (2n+1)}$$

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

int main()
{
    int i;
    double sum=1,item=1,eps=1e-6;
    for(i=1;item>=eps;i++)
    {
        item=fact(i)/multi(2*i+1);
        sum=sum+item;
    }
    printf("PI=%.5f\n",sum*2);
    return 0;
}

int fact(int n)
{
    int i;
    int res;
    for(i=1;i<=n;i++)
        res=res*i;
    return res;
}

int multi(int n)
{
    int i;
    int res=1;
    for(i=3;i<=n;i=i+2)
        res=res*i;
    return res;
}
```



实验内容 – 编程题

【编程题1】 打印月历

已知1900年1月1日是星期一，给定年月，打印当月的月历表，**要求使用函数编程。**

【输入】 两个整数，第一个整数是**年份year**，第二个整数是**月份month**，中间用一个空格间隔。
(year在 [1900,2099] 之间，month在 [1,12] 之间)

【输出】 输出月历表，月历表第一行为**星期表头**，其余各行依次是当月**各天日期**，日期数字与星期表头左对齐。

【样例输入】

2006 5

【样例输出】

```
D:\code\c\2022\stucode\bin\Debug\stucode.exe
2006 5
Sun    Mon    Tue    Wed    Thu    Fri    Sat
  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
Process returned 0 (0x0)    execution time : 2.023 s
Press any key to continue.
```

```
2006 5 若无日期，直接显示\t
Sun \t Mon \t Tue \t Wed \t Thu \t Fri \t Sat \n
\t 1 \t 2 \t 3 \t 4 \t 5 \t 6 \t \n
7 \t 8 \t 9 \t 10 \t 11 \t 12 \t 13 \t \n
14 \t 15 \t 16 \t 17 \t 18 \t 19 \t 20 \t \n
21 \t 22 \t 23 \t 24 \t 25 \t 26 \t 27 \t \n
28 \t 29 \t 30 \t 31 \t 若无最后一天是周六，则显示\t\n
Process returned 0 (0x0)    execution time : 3.577 s
Press any key to continue.
```



实验内容 – 递归三要素

一、递归函数

函数是干什么用的，函数的输入参数应该是什么，返回值是什么。

二、递归表达式

确定单层递归的逻辑，重复调用自己来实现递归的过程。

三、递归终止条件

如果递归没有终止，操作系统的内存栈必然就会溢出。

斐波那契数列：1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ...

即这个数列从第三项开始，每一项都等于前两项之和。

```
1  #include <stdio.h>
2
3  int fib(int n)      递归函数
4  {
5      if (n <= 2)     终止条件
6          return 1;
7      else
8          return fib(n - 1) + fib(n - 2);  递归表达式: f(n)=f(n-1)+f(n-2)
9  }
10
11 int main()
12 {
13     int n = 0;
14     scanf("%d", &n);
15     int ret = fib(n);
16     printf("%d\n", ret);
17     return 0;
18 }
```




实验内容 – 编程题

【编程题2】 Pell数列

$a_1, a_2, a_3 \dots$ 的定义: $a_1 = 1, a_2 = 2, \dots, a_n = 2 * a_{n-1} + a_{n-2} (n > 2)$

求Pell数列的第k项模32767的值 ($1 \leq k \leq 30$) **要求使用递归编程。**

【输入】 第一行是测试数据的组数 n , 后面 n 行是测试数据 (k 值), 每组测试数据占一行

【输出】 n 行, 每行对应一个结果 ($a_k \% 32767$), 非负整数。

【样例输入】

2

1

8

【样例输出】

1

408

```
D:\code\c\2022\stucode\bin\Debug\stucode.exe
2
1
1
8
408
Process returned 0 (0x0)   execution time : 4.432 s
Press any key to continue.
```

输出结果

青 / 春 / 不 / 老 / 梦 / 想 / 永 / 在

FEEL THE MEANING OF THE TRIP

DREAM

MY DREAM WILL NEVER STOP

请同学们开始实验



哈爾濱工業大學(深圳)
HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

GO!
TAKE YOU ON A TRIP

探索 从未停止