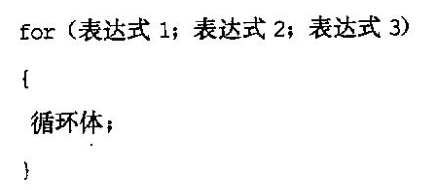
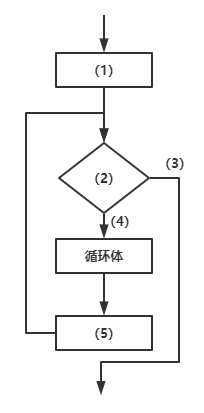
一、情景题

2.

**【说明】：**for语句比while语句和do-while语句都要灵活，是一种功能更强大、更常用的循环语句。一般语法格式如下图所示：



for语句具体的程序执行流程图如下图所示：



2.1[单选题]（1）：（ ）(2.0分)

|  |  |  |  |
| --- | --- | --- | --- |
| A、true | B、false | C、表达式1 | D、表达式2 |
| E、表达式3 |  |  |  |

［正确答案］：C

2.2[单选题]（2）：（ ）(2.0分)

|  |  |  |  |
| --- | --- | --- | --- |
| A、true | B、false | C、表达式1 | D、表达式2 |
| E、表达式3 |  |  |  |

［正确答案］：D

2.3[单选题]（3）：（ ）(2.0分)

|  |  |  |  |
| --- | --- | --- | --- |
| A、true | B、false | C、表达式1 | D、表达式2 |
| E、表达式3 |  |  |  |

［正确答案］：B

2.4[单选题]（4）：（ ）(2.0分)

|  |  |  |  |
| --- | --- | --- | --- |
| A、true | B、false | C、表达式1 | D、表达式2 |
| E、表达式3 |  |  |  |

［正确答案］：A

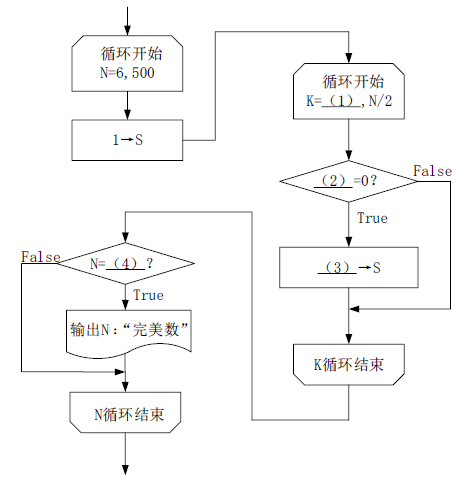
2.5[单选题]（5）：（ ）(2.0分)

|  |  |  |  |
| --- | --- | --- | --- |
| A、true | B、false | C、表达式1 | D、表达式2 |
| E、表达式3 |  |  |  |

［正确答案］：E

3.

【说明】：如果一个自然数N恰好等于它所有不同的真因子（即N的约数以及1，但不包括N）之和S，则称该数为“完美数”。例如6＝1＋2＋3，28＝1＋2＋4＋7＋14，所以6和28都是完美数。显然，6是第1个（即最小的）完美数。 下面流程图的功能是求500以内所有的完美数。



**循环开始框中要注明：循环变量＝初始值，终值［，步长］，步长为1时可以缺省。**

**其中备选项为：**

A、1

B、2

C、N%K

D、K%N

E、S+K

F、 S+N

G、S

H、K

**请选择正确的选项到如下对应的空中:**

3.1[单选题]（1）：（ ）(2.0分)

|  |  |  |  |
| --- | --- | --- | --- |
| A、1 | B、2 | C、N%K | D、K%N |
| E、S+K | F、S+N | G、S | H、K |

［正确答案］：B

3.2[单选题]（2）：（ ）(2.0分)

|  |  |  |  |
| --- | --- | --- | --- |
| A、1 | B、2 | C、N%K | D、K%N |
| E、S+K | F、S+N | G、S | H、K |

［正确答案］：C

3.3[单选题]（3）：（ ）(2.0分)

|  |  |  |  |
| --- | --- | --- | --- |
| A、1 | B、2 | C、N%K | D、K%N |
| E、S+K | F、S+N | G、S | H、K |

［正确答案］：E

3.4[单选题]（4）：（ ）(2.0分)

|  |  |  |  |
| --- | --- | --- | --- |
| A、1 | B、2 | C、N%K | D、K%N |
| E、S+K | F、S+N | G、S | H、K |

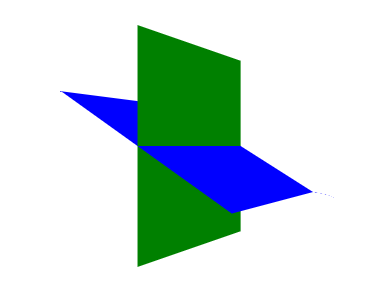
［正确答案］：G

3.5[单选题]如果某自然数小于其所有真因子之和（例如24＜1＋2＋3＋4＋6＋8＋12），则称该自然数为亏数；如果某自然数大于其所有真因子之和（例如8＞1＋2＋4），则称该自然数为贏数；如果某自然数等于从1开始的若干个连续自然数之和（例如10＝1＋2＋3＋4）则称该自然数为三角形数。据此定义，自然数496是（ 5 ）。(2.0分)

|  |  |
| --- | --- |
| A、亏数 | B、赢数 |
| C、完美数，非三角形数 | D、完美数和三角形数 |

［正确答案］：D

4.

**【说明】：**根据代码中的注释和效果图补全代码（1）-（5）：

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta http-equiv="X-UA-Compatible" content="IE=edge">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>3D呈现</title>      <style>          body{              perspective: 500px;          }          .box {              width: 200px;              height: 200px;              （1）; /\*设置box盒子为相对定位\*/              margin: 100px auto;              transition: all 2s;              （2）: preserve-3d;  /\* 让子盒子保持3D效果 \*/            }          .box:hover {              transform: （3）; /\*当鼠标悬停到box上时，沿Y轴旋转60度\*/          }          .box div{              position: （4）; /\*将box下的div元素设为绝对定位\*/              top:0;              left: 0;              width: 100%;              height: 100%;          }          .box div:（5）{ /\*表示选择box下的第一个div元素\*/              background-color: green;          }          .box div:last-child{              background-color: blue;              transform: rotateX(60deg);          }      </style>  </head>  <body>      <div class="box">          <div></div>          <div></div>      </div>  </body>  </html> |

4.1[单选题]（1）：（    ）(2.0分)

|  |  |
| --- | --- |
| A、fixed | B、rotateY(60deg) |
| C、nth-child(0) | D、transform |
| E、absolute | F、first-child |
| G、position: relative | H、transform-style |

［正确答案］：G

4.2[单选题]（2）：（    ）(2.0分)

|  |  |
| --- | --- |
| A、fixed | B、rotateY(60deg) |
| C、nth-child(0) | D、transform |
| E、absolute | F、first-child |
| G、position: relative | H、transform-style |

［正确答案］：H

4.3[单选题]（3）：（    ）(2.0分)

|  |  |
| --- | --- |
| A、fixed | B、rotateY(60deg) |
| C、nth-child(0) | D、transform |
| E、absolute | F、first-child |
| G、position: relative | H、transform-style |

［正确答案］：B

4.4[单选题]（4）：（    ）(2.0分)

|  |  |
| --- | --- |
| A、fixed | B、rotateY(60deg) |
| C、nth-child(0) | D、transform |
| E、absolute | F、first-child |
| G、position: relative | H、transform-style |

［正确答案］：E

4.5[单选题]（5）：（    ）(2.0分)

|  |  |
| --- | --- |
| A、fixed | B、rotateY(60deg) |
| C、nth-child(0) | D、transform |
| E、absolute | F、first-child |
| G、position: relative | H、transform-style |

［正确答案］：F

5.

**【说明】：阅读以下说明和Java语言代码，并选出正确答案。**

题目描述：判断101-200之间有多少个素数，并输出所有素数。

**具体函数如下：**

public class Test {

    public static void main(String[] args) {

        int i, j, n, m, x;

        n = 0;

        m = 0;

        x = 0;

        for (（1）; （2）; i++) {

            for (（3）; （4）; j++) {

                n = i % j;

                if (n == 0) {

                    m = m + 1;

                }

            }

            if (（5）) {

                System.out.println(i + " ");

                x = x + 1;

            }

            m = 0;

        }

        System.out.println();

        System.out.println("在101~200之间一共有素数：" + x + "个");

    }

}

5.1[单选题]（1）：（    ）(2.0分)

|  |  |
| --- | --- |
| A、i = 101 | B、i = 100 |
| C、i < 200 | D、i <= 200 |
| E、j = 1 | F、j = 0 |
| G、j <= i | H、j < i |

［正确答案］：A

5.2[单选题]（2）：（    ）(2.0分)

|  |  |
| --- | --- |
| A、i = 101 | B、i = 100 |
| C、i < 200 | D、i <= 200 |
| E、j = 1 | F、j = 0 |
| G、j <= i | H、j < i |

［正确答案］：C

5.3[单选题]（3）：（    ）(2.0分)

|  |  |
| --- | --- |
| A、i = 101 | B、i = 100 |
| C、i < 200 | D、i <= 200 |
| E、j = 1 | F、j = 0 |
| G、j <= i | H、j < i |

［正确答案］：E

5.4[单选题]（2）：（    ）(2.0分)

|  |  |
| --- | --- |
| A、i = 101 | B、i = 100 |
| C、i < 200 | D、i <= 200 |
| E、j = 1 | F、j = 0 |
| G、j <= i | H、j < i |

［正确答案］：G

5.5[单选题]（5）：（    ）(2.0分)

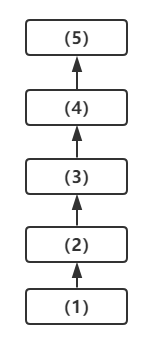
|  |  |
| --- | --- |
| A、i = 101 | B、i = 100 |
| C、i < 200 | D、i <= 200 |
| E、j = 1 | F、j = 0 |
| G、j <= i | H、m == 2 |

［正确答案］：H

二、案例题

6.

**【说明】：**CMM（能力成熟度模型）是组织进行软件过程改善和软件过程评估的一个有效的指导框架。它定义了当一个组织达到不同的过程时应该具有的软件工程能力，其主要目的是消除不同模型之间的不一致和重复，降低模型改善的成本。将CMM组织成如下图所示的五个等级，显示了该过程中五个成熟阶段的基本特性和应遵循的原则、采取的行动，以帮助软件组织改进其软件过程。其意义在于增加软件过程成熟的改进行动，按优先级排序。图中带有标记的箭头指示在成熟度框架的每一个步骤上，组织应给予规范化的过程能力的类型。



**其中（1）-（5）分别表示从最低级到最高级的顺序，请补充上述括号中缺失的部分到如下对应的空中:**

6.1[填空题]（1）：（ ）(2.0分)

［参考答案］：初始级

6.2[填空题]（2）：（ ）(2.0分)

［参考答案］：可重复级

6.3[填空题]（3）：（ ）(2.0分)

［参考答案］：已定义级

6.4[填空题]（4）：（ ）(2.0分)

［参考答案］：已管理级

6.5[填空题]（5）：（ ）(2.0分)

［参考答案］：优化级

8.

**【说明】：**通过阅读下列JavaScript代码补全（1）-（5）。

在计算机科学领域，二分查找（Binary Search）是一种查找算法，用来在一个数组中查找指定的元素。注意这个数组需要是一个有序数组才有效。二分查找优于标准的线性查找（Linear Search），因为它查找速度更快，效率更高。它的理念被开发者称之为“分而治之”。

 这种算法不是在一个for循环中依次按照索引0,1,2,3,4这样查找，它的具体做法是每次把查找范围缩小一半，直到找到目标值。

我们会在数组中间的元素作为middleIndex元素，然后把它的值和目标值进行比对。根据目标值是大于还是小于middleIndex元素，我们可以从数组中排除左侧或者右侧的元素。

|  |
| --- |
| <script>  function find(nums, target) {  let left = 0;  let rigth = (1); //  while(left <= rigth) {  let mid = parseInt((left + rigth) / 2) ;  if (2)  return mid ;  if (3) {  if (target >= nums[left] && target <= nums[mid]) {  rigth = mid - 1;  } else {  (4);  }  } else {  if (target > nums[mid] && target <= nums[rigth]) {  left = mid + 1;  } else {  (5);  }  }  }  return -1  }  console.log(find([6,7,8,9,10,2,3,4,5], 5))//输出8  </script> |

8.1[填空题]（1）：（）(2.0分)

［参考答案］：nums.length - 1

8.2[填空题]（2）：（）(2.0分)

［参考答案］：nums[mid] == target

8.3[填空题]（3）：（）(2.0分)

［参考答案］：nums[left] < nums[mid]

8.4[填空题]（4）：（）(2.0分)

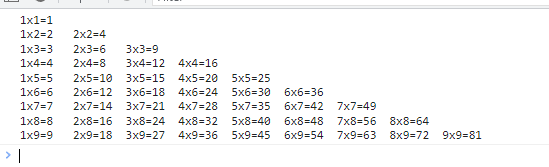
［参考答案］：left = mid + 1

8.5[填空题]（5）：（）(2.0分)

［参考答案］：rigth = mid - 1

9.

**【说明】：**通过效果图和阅读下列JavaScript代码补全（1）-（5）。



|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta http-equiv="X-UA-Compatible" content="IE=edge">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>九九乘法表</title>  </head>  <body>      <script>             var str="";                  for(var i=1;i<=（1）;i++){                      for(var j=（2）;j<=i;j++){                           str+= j+"x"+i+"="+（3）+"\t";                      }                     （4）; //换行显示                  }                  （5）; //将变量str输出到控制台      </script>  </body>  </html> |

9.1[填空题]（1）：（）(2.0分)

［参考答案］：9

9.2[填空题]（2）：（）(2.0分)

［参考答案］：1

9.3[填空题]（3）：（）(2.0分)

［参考答案］：i\*j 或 j\*i

9.4[填空题]（4）：（）(2.0分)

［参考答案］：str+='\n'

9.5[填空题]（5）：（）(2.0分)

［参考答案］：console.log(str)

三、简答题

10.评价模块划分好坏的标准主要是模块内的联系程度——内聚和模块之间的联系程度——耦合。耦合是模块之间依赖程度的度量。内聚和耦合是密切相关的，与其他模块存在强耦合的模块通常意味着弱内聚，而强内聚的模块通常意味着与其他模块之间存在弱耦合。如果系统中两个模块彼此之间完全独立，不需要另一个模块就能单独的工作，则这两个模块之间的耦合度最低。但事实上，一个系统中不可能所有的模块之间没有任何联系。文献“Pressman99”归纳了6种耦合类型，耦合程度由低到高大致划分为低端、中端和高端。模块设计应当争取“ 高内聚、低耦合”，避免“低内聚，高耦合”。请分别说出耦合的6种类型，并说明每种类型的特点。

［参考答案］：（1）非直接耦合：模块之间没有直接的信息传递，称为非直接耦合。<br>（2）数据耦合：模块之间通过接口传递参数(数据)称为数据耦合。这种耦合在两个模块之间交换的信息仅仅是数据，是低耦合。<br>（3）标记耦合：模块间通过接口传递内部数据结构的一部分(而不是简单的参数)，称为标记耦合。此数据结构的变化将使相关的模块发生变化。<br>（4）控制耦合：模块传递信号(如开关值标志量等)给另一个模块， 接收信号的模块根据信号值调整动作，称为控制耦合。<br>（5）公共耦合：两个或两个以上的模块共享信息，则这几个模块的耦合称为公共耦合。<br>（6）内容耦合：当一个模块直接修改或操作另一个模块的数据，或者直接转入另一个模块时，就发生了内容耦合。内容耦合是耦合程度最高的耦合。