给出以下代码

public class Switch1{

final static short x=2;

public static int y=0;

public static void main(String[] args) {

for(int z=0;z<4;z++){

switch(z){

case x:System.out.print(“0″);

default:System.out.print(“def”);

case x-1:System.out.print(“1″);break;

case x-2:System.out.print(“2″);

}

}

}

}

class ParentClass{

public int doStuff(int x){

return x\*2;

}

}

public class ChildClass extends ParentClass{

public static void main(String[] args){

ChildClass cc=new ChildClass();

long x=cc.doStuff(7);

System.out.println(“x=”+x);

}

public long doStuff(int x){

return x\*3;

}

}

Public static void main(String[] args){

X.x=new X();

X.x2=m1(x);

X.x4=new X();

X2=x4;

//comment

….

}

Static X m1(X mx){

Mx=new X();

Return mx;

}

}

以上代码执行到comment处时有多少个可被垃圾收集的对象

**给出以下代码**

**Public class TestObj{**

**Public static void main(String[] args){**

**Object o=new Object(){**

**Public boolean equals(Object obj){**

**Return true;**

**}**

**};**

**System.out.println(o.equals(“Fred”));**

**}**

**}**

**请给出结果：（）**

**A 运行时抛出异常**

***B True***

**C Fred**

**D 第三行编译错误**

1. **给出以下代码**

**Class Two{**

**Byte x;**

**}**

**Class PassO{**

**Public static void main(String[] args){**

**PassO p=new PassO();**

**P.start();**

**}**

**Void start(){**

**Two t=new Two();**

**System.out.print(t.x+””);**

**Two t2=fix(t);**

**System.out.print(t.x+” ” +t2.x);**

**}**

**Two fix(Two tt){**

**tt.x=42;**

**Return tt;**

**}**

**}**

**请给出结果：**

**A null null 42 C 0 42 42 E Compilation**

**B 0 0 42            D 0 0 0          F An exception is thrown at runtime**

5 给出以下代码：

Public class Switch2{

Public static void main(String[] args){

Int N=30;

Boolean[] flag=new boolean[N+1];

For(int i=2;i<=N;i++){

If(!flag[i]){

For(int j=i\*2;j<=N;j+=i)

Flag[j]=true;

}

}

}

for(int i=2;i<=N;i++){

if(!flag[i]){

System.out.print(i+” “);

}

}

}

}

运行结果：

1. 请写出以下程序的输出结果

public class A2{

public static void main(String[] args){

int[] a={2,4,6,8,3,6,9,12};

doSomething(a,0,a.length-1);

for(int i=0;i<=a.length-1;i++)

System.out.print(a[i]+” “);

}

}

private static void doSomething(int[] a,int start,int end){

if(start<end){

int p=core(a,start,end);

doSomething(a,start,p-1);

doSomething(a,p+1,end);

}

}

Private static int core(int[] a,int start,int end)

int x=a[end];

int i=start;

for(int j=start;j<=end-1;j++){

if(a[j]>=x){

swap(a,i,j);

i++;//交换了几次

}

}//把最大的放到最后

Swap(a,i,end);//把最大的放到i的位置

Return i;

}

private static void swap(int[] a,inti,int j){

int tmp=a[i];

a[i]=a[j];

a[j]=tmp;

}

}

答案：从大到小排序