

به نام خدا

امتحان میان برنامه نویسی ترم دو سال تحصیلی وقت: 120 دقیقه دانشگاه ایلام
ترم پیشرفته 91-92

لطفاً، در صورت وجود ابهام در سوال به صورت زیر عمل کنید:

- ابهام مورد را نظر را به طور روشن توضیح دهید.
- فرضهای خود را برای رفع ابهام ذکر کنید.
- مساله را مطابق فرضهای خود حل کنید.

1- خروجی کد: تعداد دفعات تکرار حلقه ی موجود در کد زیر را تعیین کنید: (۴ نمره)

```
int balance = 10;
while (balance >= 1) {
    if (balance < 9) {
        continue;
    }
    balance = balance - 9;
}
```

جواب: این حلقه بینهایت بار تکرار میشود.

2- خروجی کد: خروجی کد زیر را تعیین کنید. فرض کنید کاربر اعداد ۲، ۳، ۴، ۵ و ۰ را به ترتیب وارد کرده است. (۶ نمره)

```
import java.util.Scanner;
public class Test {
    public static void main(String[] args){
        Scanner input = new Scanner(System.in);
        int number, max;
        number = input.nextInt();
        max = number;
        while (number != 0) {
            number = input.nextInt();
            if (number > max)
                max = number;
        }
        System.out.println("max is " + max);
        System.out.println("number " + number);
    }
}
```

جواب: max is 5
number 0

3- خروجی: مقدار a، b و c را بعد از فراخوانی متد foo مشخص کنید. (۵ نمره)

```
public static void main(String[] args) {
    int a = 10;
    int b = 20;
    int[] c = { 5, 10, 15, 20 };
    foo(a,b,c);
}
public static void foo(int x, int y, int[] z) {
    x += 5;
    y *= 2;
    for (int i = 0; i < z.length; i++){
```

```

        z[i] -= 3;
    }
}

```

جواب: چون ما یک کپی از مقادیر a و b را به تابع foo فرستاده ایم پس تغییر x و y تاثیری روی a و b ندارد و a و b همان مقادیر قبلی خودشان را حفظ میکنند ولی چون پارامتر سوم تابع foo یک آرایه است و وقتی یک آرایه را به تابع میفرستیم تغییر روی آرایه در تابع آرایه اصلی را تغییر میدهد در این جا اعضای آرایه سه واحد کم میشوند.

```

a=10
b=20
{2,7,12,17}

```

4- آرایه: برنامه‌ای بنویسید که تعدادی عدد را از کاربر بگیرد و مشخص کند چه تعداد از آنها بالای میانگین، چه تعداد زیر میانگین و چه تعدادی برابر میانگین هستند. اگر کاربر یک عدد منفی را وارد کند به معنای این است که عدد قبلی آخرین عدد وارد شده بود. به هر حال، کاربر حداکثر می‌تواند ۱۰ عدد وارد کند. دقت کنید عدد منفی در محاسبات نقشی ندارد و فقط پایان اعداد را نشان می‌دهد. (۲۰ نمره)

```
import java.util.Scanner;
```

```

public class myClass{
    public static void main(String[] args){
        double[] a=new double[10];
        double b,sum=0,ave=0;
        Scanner s= new Scanner(System.in);
        int countave=0,countupave=0,countdownave=0,count=0;
        for(int i=0;i<10;i++){
            b=s.nextDouble();
            if(b<0)break;
            a[i]=b;count++;sum+=a[i];
        }//end of for
        if(count==0){System.exit(0);}
        ave=sum/count;
        for(int i=0;i<count;i++){
            if(a[i]==ave)countave++;
            if(a[i]>ave)countupave++;
            if(a[i]<ave)countdownave++;
        }//end of for
    }
}

```

```

System.out.print("upave="+countupave+"\tdownave="+countdownave+"\tcounta
ve="+countave);

```

```

} //end of main method

```

```

} //end of class

```

5- رشته: یک متد بنویسید که یک رشته را بگیرد و تعداد حروف کوچک، تعداد حروف بزرگ و تعداد ارقام رشته را چاپ کند. (۱۰ نمره)

```
import java.util.Scanner;
```

```

public class myClass{
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        String s = in.nextLine();
        int up=0, down=0, number=0;
        for (int i = 0; i < s.length(); i++) {
            if(s.charAt(i)>='0'&& s.charAt(i)<='9')
                number++;
            else if(s.charAt(i)>='A'&& s.charAt(i)<='Z')
                up++;
            else if(s.charAt(i)>='a'&& s.charAt(i)<='z')
                down++;
        }
        System.out.println(up+" "+down+" "+number);
    }
}

```

6-بازگشت: یک متد بازگشتی بنویسید که یک عدد طبیعی را بگیرد و معادل دودویی آنرا چاپ کند. متد شما دو آرگومان دارد. آرگومان اول عدد یک عدد طبیعی و آرگومان دوم یک رشته است که در ابتدا خالی است. در هر مرحله از بازگشت عدد را بر دو تقسیم و باقیمانده را به طور مناسب به رشته اضافه کنید. (۲۰ نمره)

```

public static String binary(int n, String s) {
    if(n==0)
        return s;
    return binary(n/2,(n%2)+s);
}

```

7-کلاس: فرض کنید که می‌خواهید یک کلاس بنویسید که عملکرد یک پنکه ی سقفی را پیاده کند. پنکه دارای سه سرعت کم، متوسط و زیاد است. سرعت را با یک متغیر طبیعی مدل کنید و مقدار ۰ را برابر کم، ۱ را برابر متوسط و ۲ را برابر زیاد در نظر بگیرید. بقیه ی مقادیر غیر مجاز هستند. پنکه دارای یک متغیر بولی است که نشان دهنده ی خاموش یا روشن بودن پنکه است. جز سازنده پیش فرض سرعت پنکه را برابر کم قرار می دهد و آنرا در وضعیت خاموش قرار می دهد. یک متد برای چک کردن وضعیت پنکه (خاموش یا روشن) و یک متد برای تغییر سرعت آن به یک مقدار جدید بنویسید. یک متد نیز بنویسید که اگر پنکه روشن باشد آنرا خاموش کند و اگر خاموش باشد آنرا روشن کند. متدهای toString و جزء سازنده ی کپی را نیز پیاده کنید. (۲۰ نمره)

```

public class Fan {
    private boolean situation;

```

```

private int speed;

public Fan() {
    situation=false;
    speed=0;
}

public boolean CheckSituation() {
    return situation;
}

public void ChangeSpeed(int newSpeed) {
    if (newSpeed < 0 || newSpeed > 2)System.exit(0);
    speed = newSpeed;
}

public void ChangeSituation() {
    if (situation)
        situation = false;
    else
        situation = true;
}

public String toString() {
    String s = "",z="";

    if (speed == 0)
        s += "Low";
    else if (speed == 1)
        s += "Medium";
    else if (speed == 2)
        s += "High";
    if(situation)z+="on";
    else z+="off";

    return "\tSituation : " + z+"\tspeed:"+s;
}

public Fan(Fan newFan) {
    this.situation=newFan.situation;
    this.speed=newFan.speed;
}
}

```

8-ترکیب: فرض کنید که یک کلاس به نام Repair داریم که تاریخ تعمیر وسیله خانگی (از نوع Date)، شرح تعمیرات انجام شده (از نوع String)، و نام تعمیرکار (از نوع String) در آن ذخیره شده است. وسیله ی خانگی را با کلاس Equipment مدل می‌کنیم. کلاس Equipment دارای تاریخ ساخت (از نوع Date)، آرایه ای از تعمیرات انجام شده (از نوع Repair [])، اسم وسیله (از نوع String)، و قیمت وسیله (از نوع int) است. جزء سازنده ی کلاس Equipment و متد equals آنرا پیاده کنید. (۲۰ نمره)

```
public class Equipment {

    private Date date;
    private Repair[] repair;
    private String name;
    private int cost;

    public Equipment(Date date, Repair[] r, String name, int cost) {
        if(date==null||r==null||name==null||name=="||cost<=0)System.exit(0);
        this.date = date;
        repair = helper(r);
        this.name = name;
        this.cost = cost;
    }
    private Repair[] helper(Repair[] r){
        Repair[] temp=new Repair[r.length];
        for(int i=0;i<r.length;i++){
            temp[i]=new Repair(i);
        }
        return temp;
    }

    public boolean equals(Equipment E){

        if(!(date.equals(E.date))||!(name.equals(E.name))||cost!=cost||(repair.length!=E.repair.length))return false;
        int n=0;
        for(int i=0;i<repair.length;i++){
            for(int j=0;j<E.repair.length;j++){
                if((repair[i].equals(E.repair[j])))n=1;break;
            }//end of for
            if(n==0)return false;
            n=0;
        }//end of for
        return true;
    }//end of equals method
}
```

```
}//end of main method  
}//end of class
```