作业一report

Homework1

提示用户输入年月日信息，判断这一天是这一年中的第几天并打印

初始化信息

```
int[] monthArr = {31,28,31,30,31,30,31,31,30,31,30,31};
```

接收用户输入的信息，年 月 日

```
Scanner sc = new Scanner(System.in);
System.out.println("please input the year:");
int year = sc.nextInt();
if(year%4==0 &&year%100 !=0){
    monthArr[1] = 29;
}
System.out.println("please input the month:");
int month = sc.nextInt();
if(month<1||month>12){
    System.out.println("the information of the month is wrong ");
    return;
}
System.out.println("please input the day:");
int day = sc.nextInt();
if(day>monthArr[month-1]){
    System.out.println("the information of the day is wrong");
    return;
}
```

用条件语句if(year%4==0 &&year%100 !=0) 判断用户输入的年份是否为闰年
如果是闰年则用monthArr[1] = 29;表示，数组monthArr中第二个元素值为29

```
    int days = 0;
    for(int i=0;i<month-1;i++){
        days+= monthArr[i];
    }
    days += day;

    System.out.println(year+"-year "+month+"-month "+day+"-day is the "+days+" day in this year");
}
```

用for语句循环把这一个月前所有月份的总天数相加，最后再加本月所在的天数，得到总天数。打印结果。
程序运行如下，我输入2036年5月25日，得到这天是这一年的第146天。

Homework2

找出 1000 以内的所有完数并打印出来。

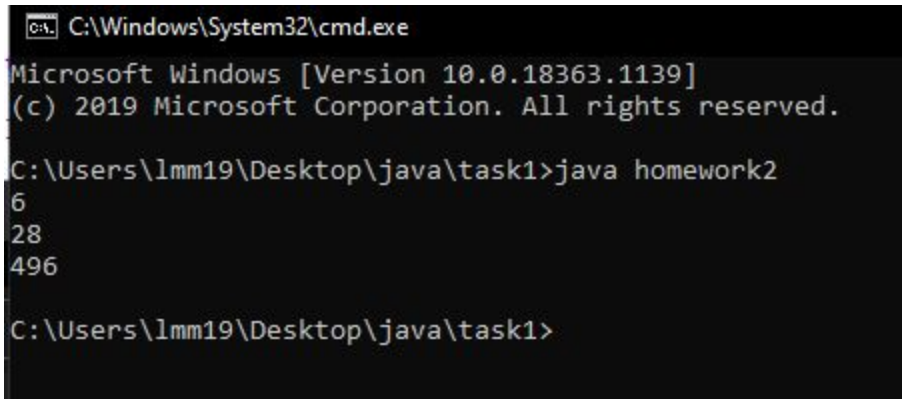
找到这个数的所有因子，其中1和其本身必定为因子

循环找到能被该数整除的因子并记录

判断除去该数本身的其他因子的和是否等于该数

```java
public class homework2 {
    public static void main(String[] args) {
        int sum;
            for (int i =1; i <= 1000; i++){
                sum=0;
                for(int j=1; j < i; j++){
                    if(i%j==0){
                        sum +=j;
                    }
                }
                if (sum == i){
                    System.out.println(i);
                }
            }
    }
}
```

运行程序，得到1000内的所有完数为 6，28，496



Homework3

随机生成6个范围在1~33的整数，每输入一个数时，判断之前是否生成过，若生成过，则重新生成，直至不跟之前重复为止，用 do while 结构实现

```
import java.util.Random;
import java.util.Arrays;

public class homework3 {

    public static void main(String[] args) {

        int[] numList = new int[7];
```

// 声明一个长度为6类型为int的一维数组来保存生成的号码
// 其中前六个元素为红球号码，第七个元素为蓝球号码
//默认初始值为0

随机生成6个数（1～33）
建立do while结构，如果数组numList[i]与numList[j]重复，则isOverlap为真，跳出j的for循环，重新取值，直到生成6个数值不重复的数组

```
int redLimit = 33;
Random ra = new Random();
boolean isOverlap = false;

for(int i=0; i<6; i++) {


    do {
        numList[i] = ra.nextInt(redLimit) + 1;

        for(int j=0; j<i; j++) {
            if(numList[i]==numList[j]) {
                isOverlap = true;
                break;
            } else {
                isOverlap = false;
            }
        }

    } while(isOverlap);

}

System.out.println("--------------------------------");
```

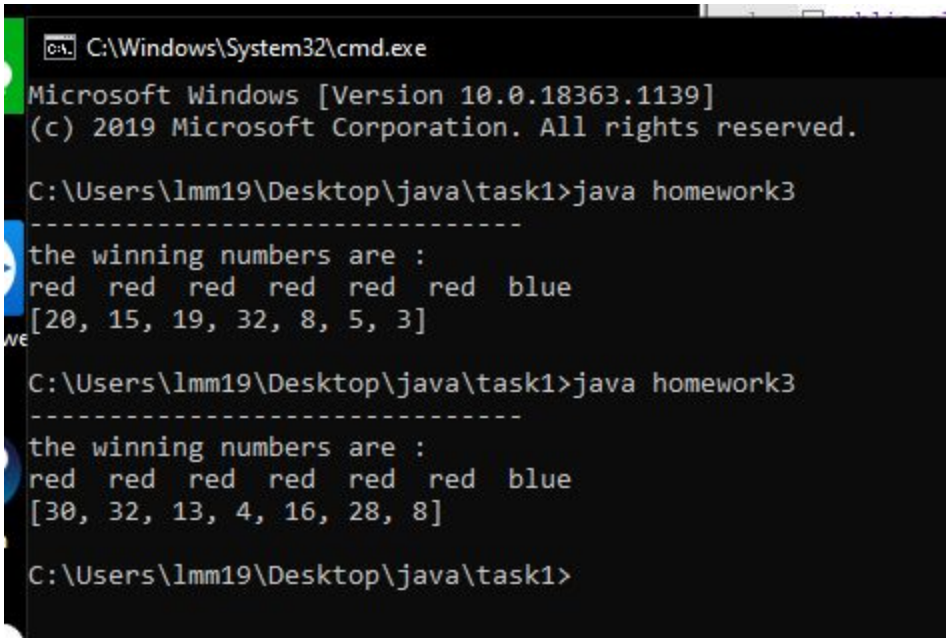
生成一个范围在1～16的随机整数记为蓝色号码

```
int blueLimit = 16;
numList[6] = ra.nextInt(blueLimit) + 1;


int[] RedNumList = new int[6];
for(int i=0; i<numList.length-1; i++) {
    RedNumList[i] = numList[i];
}
int blueNum = numList[6];

System.out.println("the winning numbers are :");
System.out.println("red  red  red  red  red  red  blue");
System.out.println(Arrays.toString(numList));
```

打印结果：前6个号码是红色球，最后一个是蓝色球

程序运行：

运行3次 分别得到随机的6个红色球和一个蓝色球



Homework4

声明一个数组，比如长度为用户输入的capacity，类型为int的一维数组

```
Scanner sc = new Scanner(System.in);

System.out.println("please enter the capacity of the array:");
int capacity = sc.nextInt();
int[] arr = new int[capacity];
System.out.println("the capacity of the array:" + arr.length);
```

提示用户在数组中存储元素，并同时检测数组容量
int stock = 1 定义已存储量
String instruct 保存是否继续输入的指令

```
System.out.println("please enter the elements in the array:");
System.out.println("----------------------------------------");
boolean stop = false;
int stock = 1;
String instruct;

while(!stop) {

    System.out.println("please the number " + stock + " element:");
    arr[stock-1] = sc.nextInt();
    System.out.println();
    System.out.println(Arrays.toString(arr));
```

扩容，检测已存储量占总量的百分比，达到80%则扩容至150%，通过声明新的数组，将原数组内容复制到新数组，并将新数组的地址赋值给原数组

if(stock>=arr.length*4/5) 判断已存储元素数量是否达到总容量的 80%
int[] arr2 = new int[arr.length*3/2];增加数组容量为原来的1.5倍
for(int i=0; i<arr.length; i++) {arr2[i] = arr[i];}将原来数组的元素复制到扩容后的数组中
arr = arr2; 将arr指向扩容后的数组

```
if(stock>=arr.length*4/5) {
    System.out.println("-----------------------------------------");
    System.out.println("The storage capacity has reached 80% of the array capacity, which has been expanded to 1.5 times");

    int[] arr2 = new int[arr.length*3/2];

    for(int i=0; i<arr.length; i++) {
        arr2[i] = arr[i];
    }
    arr = arr2;
    System.out.println("the array capacity:" + arr.length + " (hss already stored :" + stock + ")");
    System.out.println("-----------------------------------------");
}
```

提示用户输入数组元素，输入完后询问是否还继续输入

```
    System.out.println("-------------------------------------------------");
    System.out.println("whether to continue entering(y/n)");
    instruct = sc.next();
    stop = (instruct.equals("n")) ? true : false;
```

最后打印输出

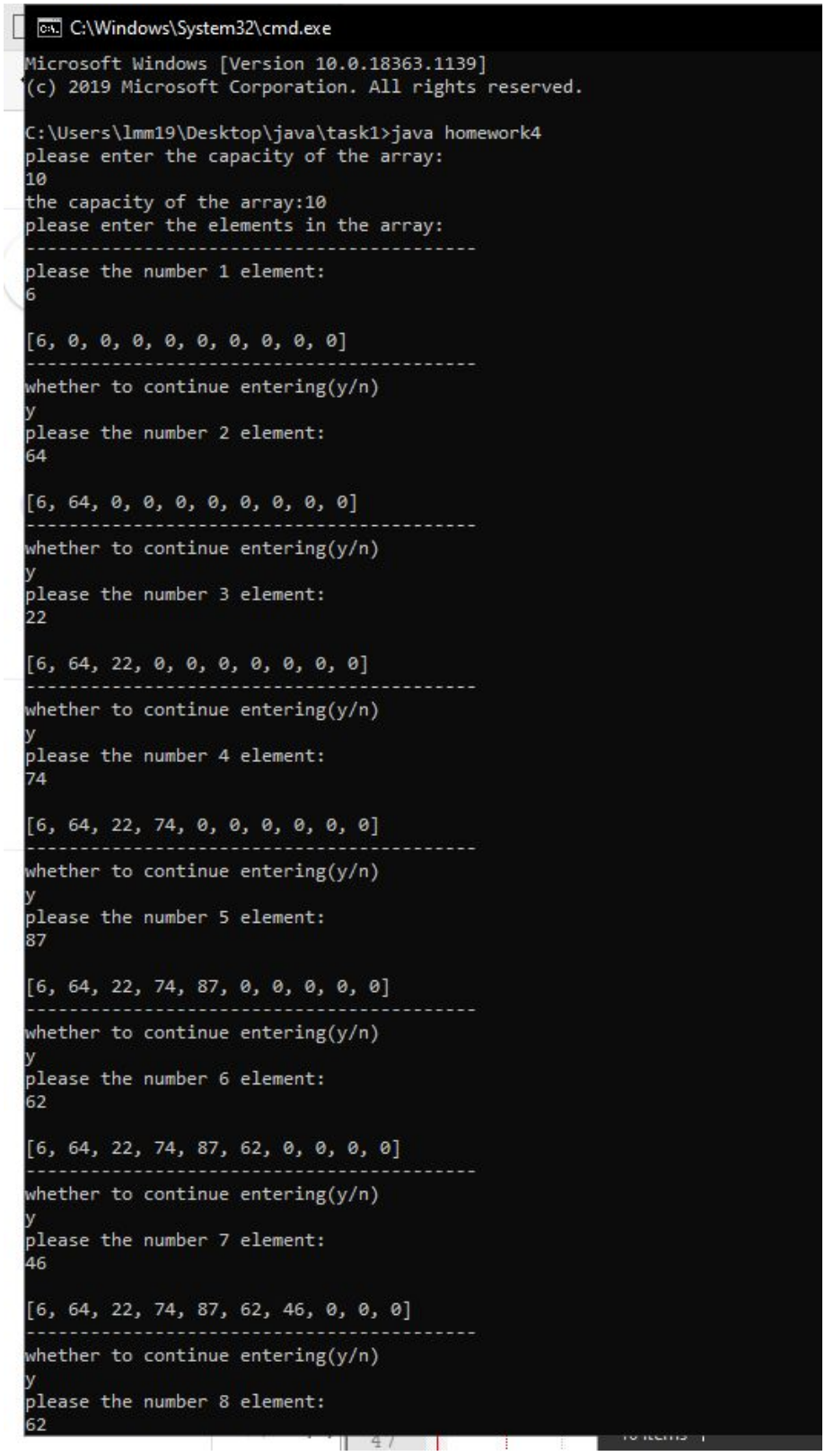
```
    if(stop){
        System.out.println("end of entering !" );
        System.out.println("the array now you get is :");
        System.out.println("-----------------------------------------");
        System.out.println(Arrays.toString(arr));
        System.out.println("-----------------------------------------");
        System.out.println("array capacity is :" + arr.length + "(has stored:" + stock + ")");
    }

    stock++;
}
```

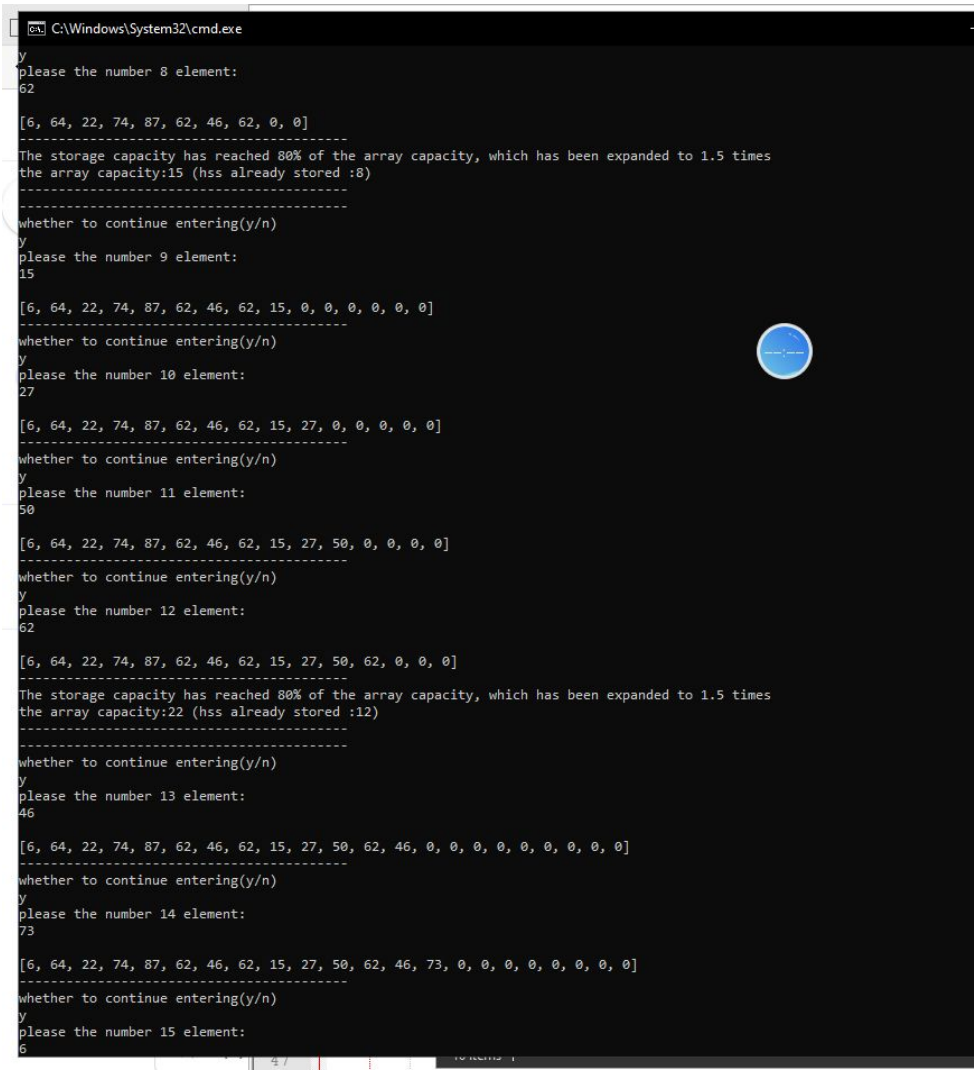
下图为程序运行

输入一维数组容量为10，并输入数组中的内容

当已存储量到达数组容量80%，即8时，增加数组容量为原来的1.5倍，即15

继续输入数组内容，输入到已存储容量为15时，数组容量变为22


C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.18363.1139]
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\lmm19\Desktop\java\task1>java homework4
please enter the capacity of the array:
10
the capacity of the array:10
please enter the elements in the array:
please the number 1 element:
6
[6, 0, 0, 0, 0, 0, 0, 0, 0, 0]
whether to continue entering(y/n)
y
please the number 2 element:
64
[6, 64, 0, 0, 0, 0, 0, 0, 0, 0]
whether to continue entering(y/n)
y
please the number 3 element:
22
[6, 64, 22, 0, 0, 0, 0, 0, 0, 0]
whether to continue entering(y/n)
y
please the number 4 element:
74
[6, 64, 22, 74, 0, 0, 0, 0, 0, 0]
whether to continue entering(y/n)
y
please the number 5 element:
87
[6, 64, 22, 74, 87, 0, 0, 0, 0, 0]
whether to continue entering(y/n)
y
please the number 6 element:
62
[6, 64, 22, 74, 87, 62, 0, 0, 0, 0]
whether to continue entering(y/n)
y
please the number 7 element:
46
[6, 64, 22, 74, 87, 62, 46, 0, 0, 0]
whether to continue entering(y/n)
y
please the number 8 element:
62




```
[6, 64, 22, 74, 87, 62, 46, 62, 15, 27, 50, 62, 46, 73, 0, 0, 0, 0, 0, 0, 0, 0]
------------------------------------------
whether to continue entering(y/n)
y
please the number 15 element:
6

[6, 64, 22, 74, 87, 62, 46, 62, 15, 27, 50, 62, 46, 73, 6, 0, 0, 0, 0, 0, 0, 0]
------------------------------------------
whether to continue entering(y/n)
n
end of entering !
the array now you get is :
------------------------------------------
[6, 64, 22, 74, 87, 62, 46, 62, 15, 27, 50, 62, 46, 73, 6, 0, 0, 0, 0, 0, 0, 0]
------------------------------------------
array capacity is :22(has stored:15)
```

Homework5

声明一个int类型的数组arr

```
public static void main(String[] args) {

    int[] arr = {48, 49, 50,51,52,53,54,55,56,57,97,98,99,100,101,102};
```

For循环

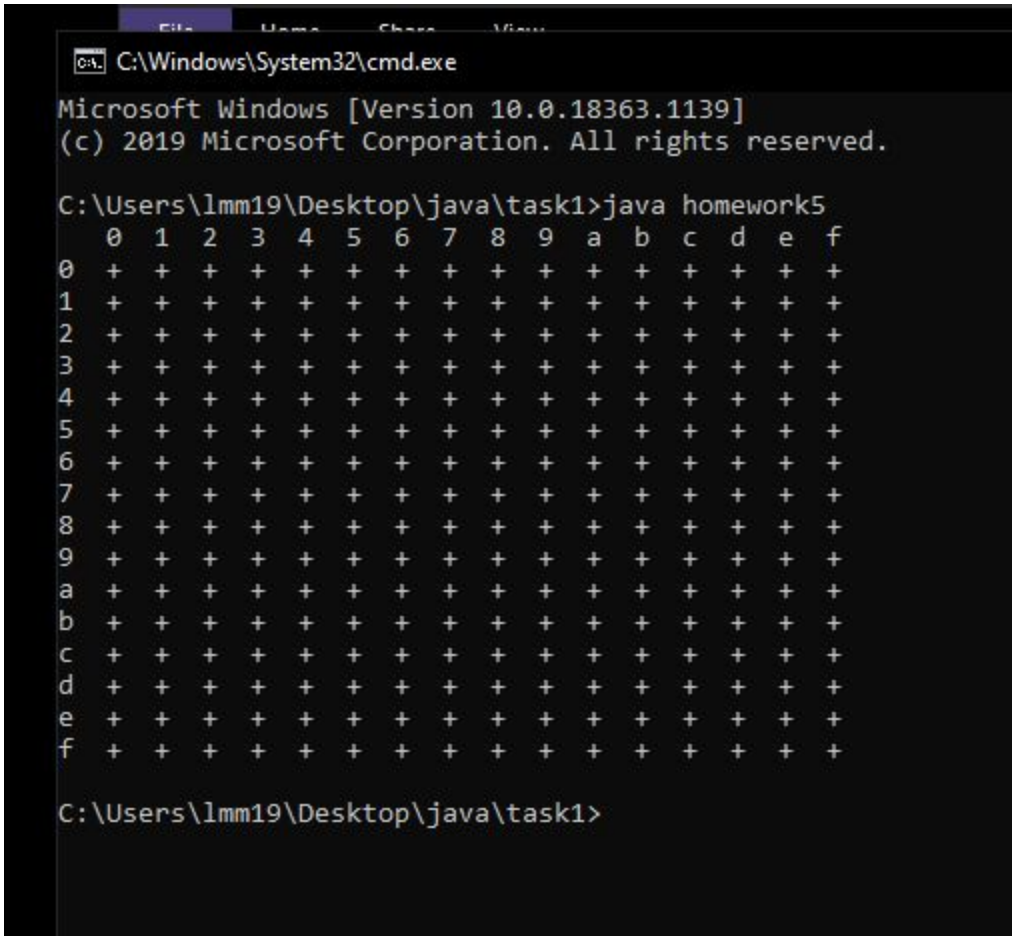
i为0时，在第一行数组打印0～9，a～f的char字符

```
for (int i = 0; i < arr.length; i++) {

    if (0 == i) {

        System.out.print(' ');

        System.out.print(' ');

        for (int j = 0; j < arr.length; j++) {

            System.out.print(' ');

            System.out.print((char)arr[j]);

            System.out.print(' ');

        }

        System.out.println();

    }
```

i不为0时，打印所在行数arr[i]的char字符，打印空格，后嵌套for循环，打印这一行后面为+，循环。

```
System.out.print((char) arr[i]);

System.out.print(' ');

for (int j = 0; j < arr.length; j++) {

    System.out.print(" + ");

}

System.out.println();

    }
```

改程序运行结果如下


C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.18363.1139]
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\lmm19\Desktop\java\task1>java homework5
0 1 2 3 4 5 6 7 8 9 a b c d e f
0 + + + + + + + + + + + + + + + +
1 + + + + + + + + + + + + + + + +
2 + + + + + + + + + + + + + + + +
3 + + + + + + + + + + + + + + + +
4 + + + + + + + + + + + + + + + +
5 + + + + + + + + + + + + + + + +
6 + + + + + + + + + + + + + + + +
7 + + + + + + + + + + + + + + + +
8 + + + + + + + + + + + + + + + +
9 + + + + + + + + + + + + + + + +
a + + + + + + + + + + + + + + + +
b + + + + + + + + + + + + + + + +
c + + + + + + + + + + + + + + + +
d + + + + + + + + + + + + + + + +
e + + + + + + + + + + + + + + + +
f + + + + + + + + + + + + + + + +
C:\Users\lmm19\Desktop\java\task1>