Recursion

recursion methods

- a recursion method is a method that calls itself. see example,P290. code.....
- imagine the statements are stacked like plates.
- **Note**:each time stackWords() is called ,a new local variable word is created.
- EXE:P311,10

general from of simple recursion methods

every recursion method has two distinct parts:

- a base case or termination condition that causes the method to end .
- a non-base case whose actions move the algorithm towards the base case and termination.

```
public void recursiveMethod(){
  if(base case){

    <perform some action>
  }
  else{
    <perform some other method>
    recursiveMehtod();//recursive  method call
  }
}
```

EXE:P308 2

- tail recursion: a method has no pending statements following the recursive call .

```
public void catastrophe(int n){
  System.outprintln(n);
  catastrophe(n);
}
```

  - **notice ! infinite recursion!!!**
  - exe:P309:3\4

practice

1. n!
   - what is the terminate condition?
2. ouput the integer parameter with the digits reversed. method name revDigs()
   - for example:
     - revDigs(176) outputs 671
     - revDigs(78) outputs 87

analysis of recursive methods

fibonacci sequence:1,1,2,3,5,8,13

```
public static int fib(int n){
  if(n==1||n==2){
    return 1;
  }
  else{
    return fib(n-1)+fib(n-2);

  }
}
```

what is the number of 7th in fibonacci? fib(7)

- this is an exponential algorithm,very **inefficient**!!!
- so : !!what we should do ?

```
public static int fib(int n){
  int prev = 1;
  int next = 1;
  int sum = 1;
  for(int i=3;i<=n;i++){ //start from 3th
    sum=prev+next;
    prev=next;
    next=sum;
  }
  return sum;
}
```

- **use recursion when it significantly simplifies code**

  EXE:7/8/19/21S

---

recursive helper methods

```
public static int sum(int n){
  if(n==1){
    return 1;
  }
  else{
    return  n+sun(n-1);
  }
}
```

- can u see any problems????
- **a private recursive helper method:** a public nonrecursive driver method that calls a private recursive helper method to carry out the task.
  - the reason of doing this:
    - hide the implementation details of the recursion from the users

- the enhance the efficiency of the program.

```
public static int getSum(int n){
  if (n>0){
    return sum(n);
  }
  else{
    throw ......
  }

}
```

Recursion in two-dimensional grids

```
public void eraseBlob(int row,int col){
  if（row >= 0 && row < size && col >= 0 && col < size）
  {
    if(image[row][col]==BLACK){
      eraseBlob(row-1,col);
      eraseBlob(row+1,col);
      eraseBlob(row,col-1);
      eraseBlob(row,col+1);
      image[row][col]==WHITE;
    }
  }
}
```

- what is the problem?

surmerize

on the AP EXAM u will be expected to calculate the results of the recursive method calls.

you also should understand that the recursive can be very inefficient.