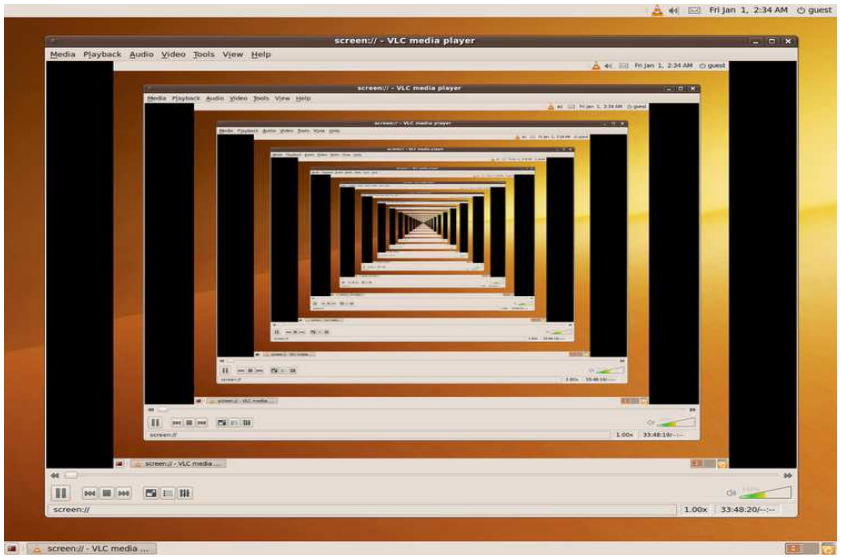
Рекурсия

Определение

- Начин да бъде описано нещо чрез обръщение към самото себе си
- Примери от живота
  - две огледала, насочени едно към друго
  - снимка на екрана на компютъра
  - името GNU
  - фракталите

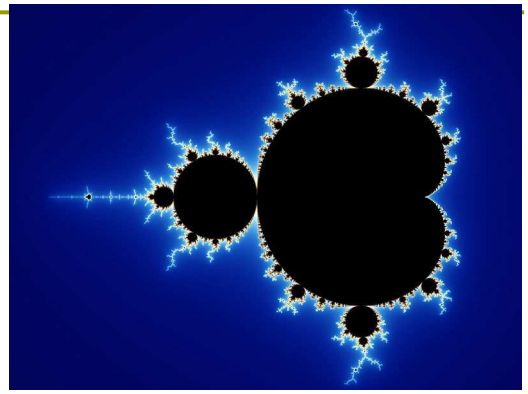


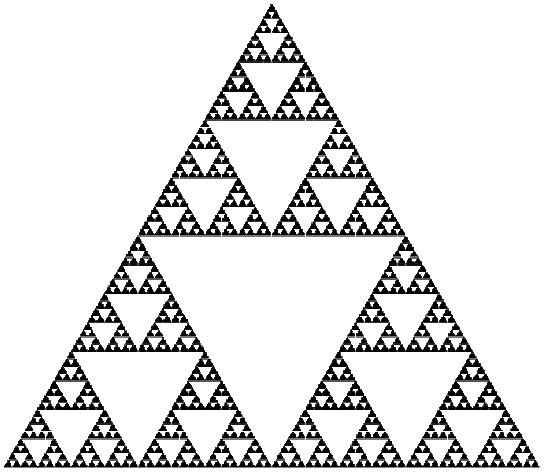
Още примери

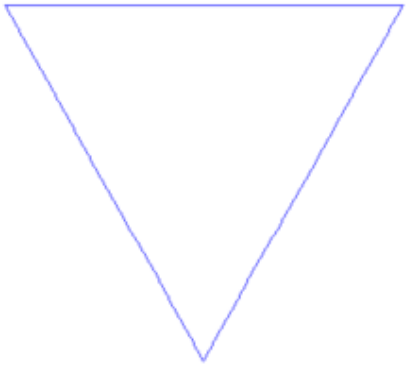
- За да отговорите на въпроса какво е рекурсия, трябва или да знаете какво е, или да питате някой друг
  - Ако не разбираш какво е рекурсия, виж: "Рекурсия"
- За да преместите 100 кашона, махнете най-горния, запомнете къде е и

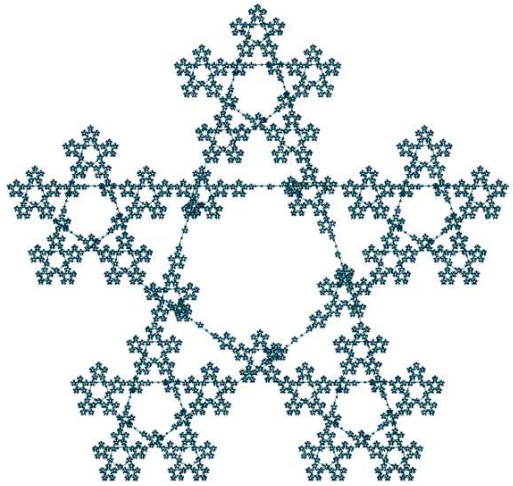
... преместете останалите :-)

Още примери









Видове рекурсия

- Пряка – когато една функция извиква самата себе си

```
int fact(int n) {
    if (n < 2) return 1;
    return n*fact(n-1);
}
```

- Непряка – когато една функция извиква друга, а тя от своя страна извиква първата

Части на рекурсията

- Разгъване – когато се извикват вложените функции
  - 3!=3.2!
  - 2!=2.1!
- Дъно – когато е достигнато условието за край
  - 1!=1
- Свиване – когато вложените функции една по една връщат резултатите си
  - 2!=2.1!=2.1=2
  - 3!=3.2!=3.2=6
- Дълбочина – броят на рекурсивните извиквания на функцията

Решаване на задачи чрез рекурсия

- Задачата се привежда до по-проста задача от същия вид
- Трябва задължително да има условие за край (осигуряващо дъното на рекурсията), и трябва разгъването на рекурсията постепенно да ни приближава до дъното
- Ако условието за дъно не е избрано коректно, рекурсията продължава докато се запълни стека, и програмата (или системата) блокира
- Кодът който трябва да се изпълни по време на разгъването се поставя преди рекурсивното извикване, а кодът, който ще се изпълнява при свиването на рекурсията - след него

Рекурсия или итерация

решение чрез рекурсия:

```
int fact(int n)
{
   if (n < 2) return 1;
   return n*fact(n-1);
}
```

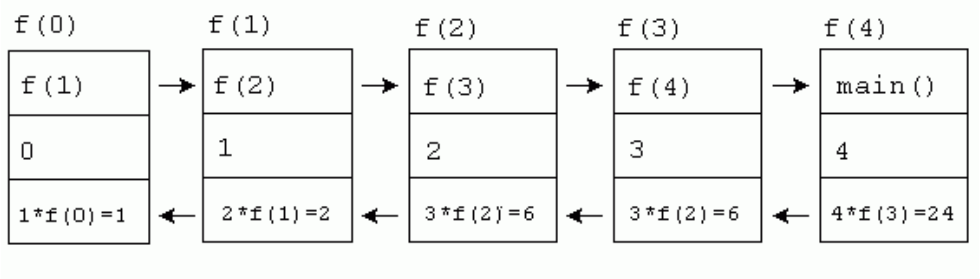
решение чрез итерации:

```
int fact(int n)
{
   int result = 1;
   for (int i=1; i<=n; i++)
         result = result * i;
   return result;
}
```

Някои задачи се решават по-удачно чрез итерации, други - чрез рекурсия

Предимства и недостатъци

- Предимства
  - По-прост код - обяснението на много задачи чрез рекурсия е по-просто от еквивалентен итерактивен алгоритъм
  - Незаменимо за някои задачи - някои задачи се решават най-елегантно чрез рекурсия
- Недостатъци
  - По-трудна за разбиране и осмисляне - рекурсията не е толкова очевидно разбираема колкото итерацията например
  - Коства повече памет - всяко рекурсивно извикване заделя допълнителна памет от стека



Ресурси в Интернет

- http://up-prakt.hit.bg/recursion_new.html
- http://www.l17cpprecurrent.hit.bg/
- http://www.mgu.bg/drugi/ebooks/hristov/chapter9.htm
- http://pitata.org/index.php?id=50
- http://www.introprogramming.info/intro-java-book/read-online/glava10-rekursia/

Край