Concepts Informatiques

2019–2020

Matthieu Picantin

```
int res=1,cpt=2,arg=7;
while(cpt<=arg) res*=cpt++;
return res;
```

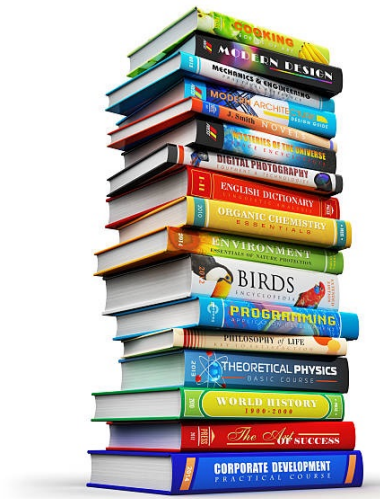
pensée

calcul
récursion
fonction
objet
⋮

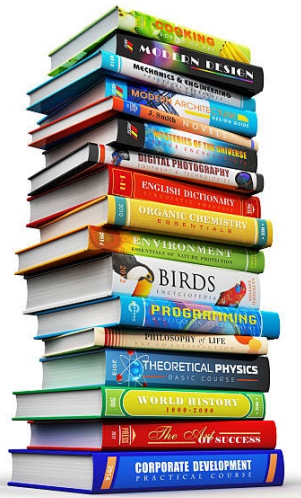
machine

circuit
pile
registre
mémoire
⋮

```
10111000 00000001 00000000
00000000 00000000 10111010
00000010 00000000 00000000
00000000 00111001 11011010
01111111 00000110 00001111
10101111 11000010 01000010
11101011 11110110 11000011
```

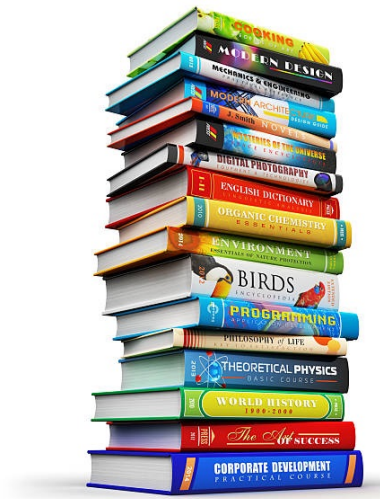

COOKING
MODERN DESIGN
DIGITAL PHOTOGRAPHY
ENGLISH DICTIONARY
ORGANIC CHEMISTRY
ESSENTIALS
ENVIRONMENT
BIRDS
PROGRAMMING
PHILOSOPHY of LIFE
THEORETICAL PHYSICS
BASIC COURSE
WORLD HISTORY
The Art of SUCCESS
CORPORATE DEVELOPMENT
PRACTICAL COURSE

pile
(stack)



pile
(stack)

tas
(heap)



pile
(stack)

tas
(heap)

http://www.pythontutor.com/java.html

```
int res=1,cpt=2,arg=7;
while(cpt<=arg) res*=cpt++;
return res;
```

pensée

calcul
récursion
fonction
objet
⋮

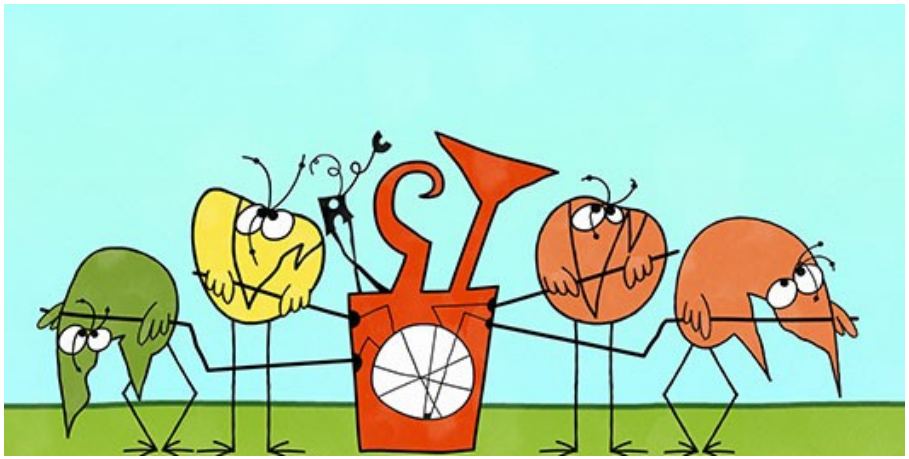
machine

circuit
pile
registre
mémoire
⋮

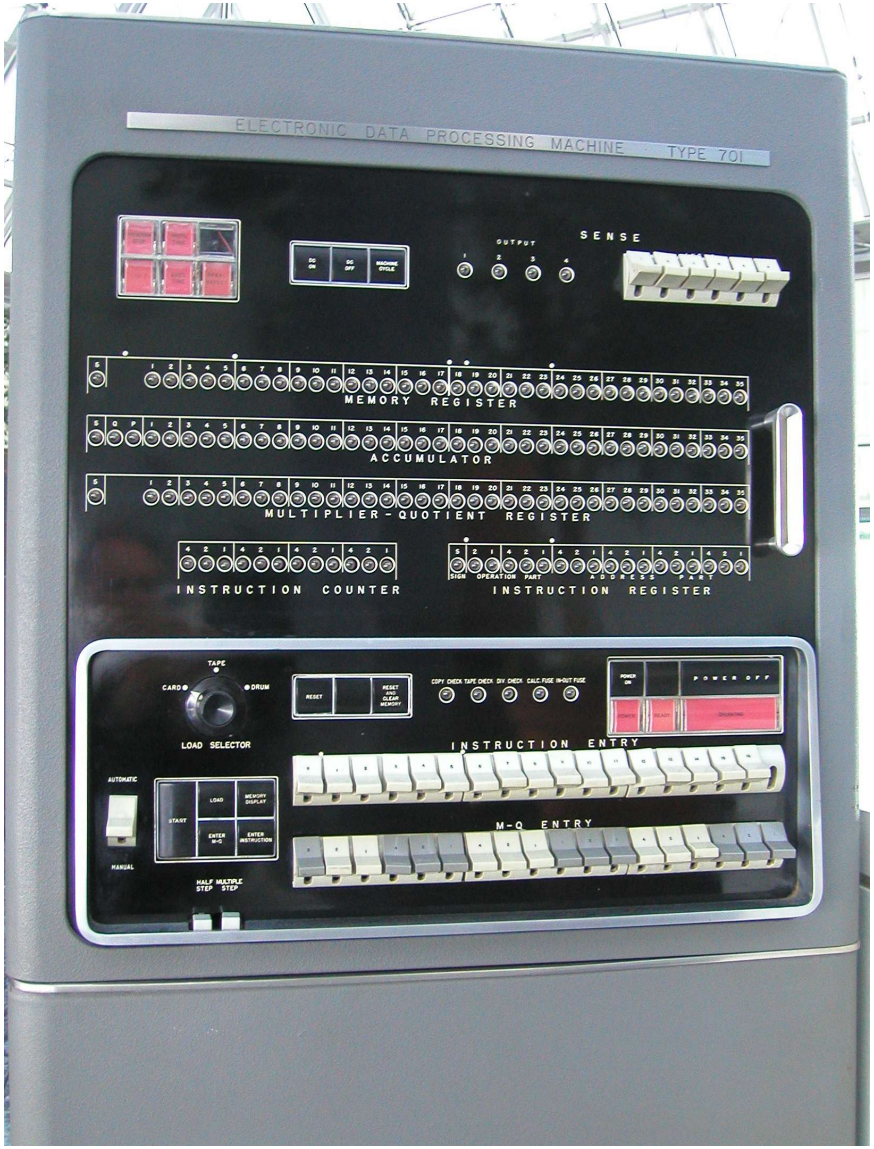
```
10111000 00000001 00000000
00000000 00000000 10111010
00000010 00000000 00000000
00000000 00111001 11011010
01111111 00000110 00001111
10101111 11000010 01000010
11101011 11110110 11000011
```

Traduire tout programme
dans une forme très proche
de celle acceptée par les machines

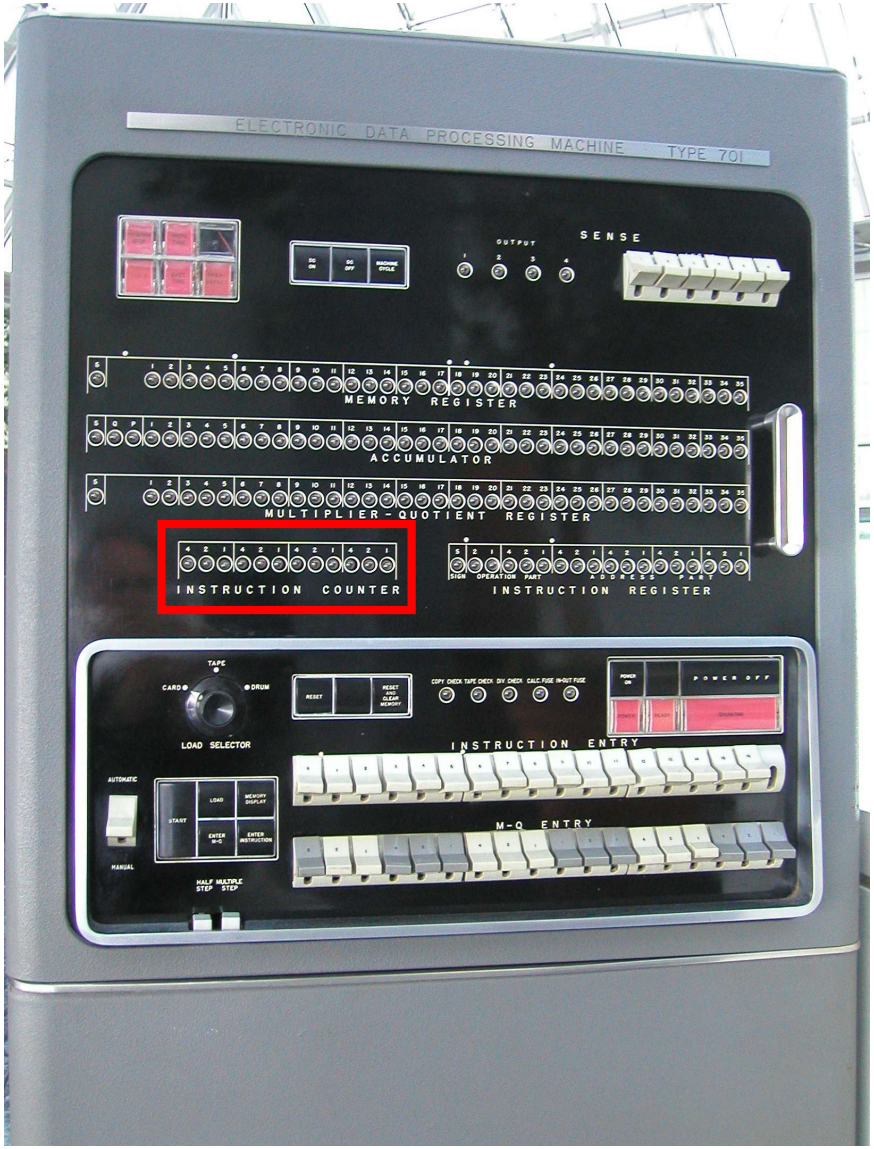
```
class Facto{
  static int f(int n){
    int i=1, r=1;
    while(i<=n){
      r = r*i;
      i = i+1;
    }
    return r;
  }
  public static void main(String[] args){
    int x=4;
    System.out.println("resultat : "+f(x));
  }
}
```



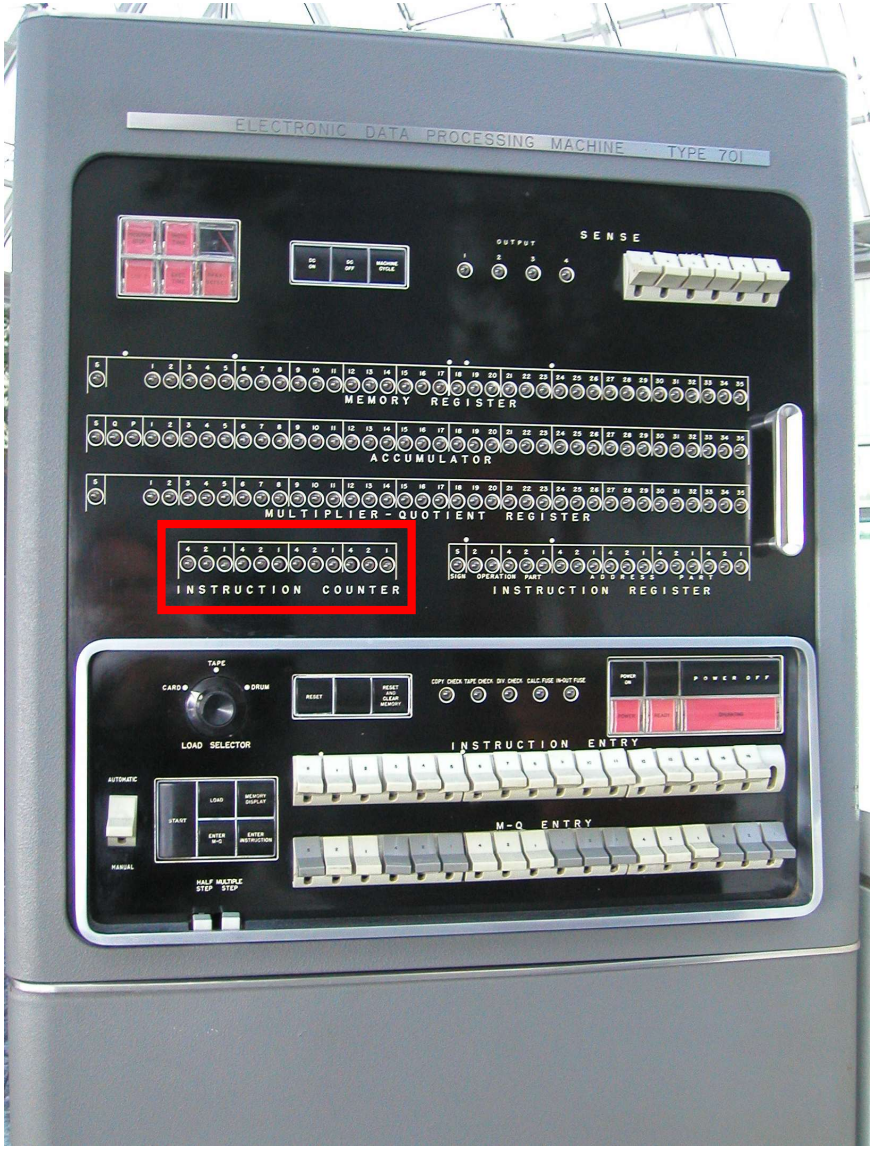
```
import java.util.*;
class FactoTraduit{
  public static void main(String[] args){
    int[] mem=new int[1000];
    int ic=0;
    Stack<BlocF> p=new Stack<BlocF>();
    while(true){
      switch(ic++){
        case 0: mem[0]=4;
        case 1: p.push(new BlocF(ic,mem[0]));
                ic=500; break;
        case 2: System.out.println("resultat : "
                +p.pop().getVal()); break;
        case 3: System.exit(0);
        case 50: p.peek().setVar1(1); break;
        case 51: p.peek().setVar2(1); break;
        case 52: if(p.peek().getVar1()
                >p.peek().getArg()) ic+=3; break;
        case 53: p.peek().setVar2(
                p.peek().getVar2()
                *p.peek().getVar1()); break;
        case 54: p.peek().setVar1(
                p.peek().getVar1()+1); break;
        case 55: ic-=4; break;
        case 56: p.peek().setVal(
                p.peek().getVar2()); break;
        case 57: ic=p.peek().getAdr(); break;
      }
    }
  }
}
```




```
import java.util.*;
class FactoTraduit{
  public static void main(String[] args){
    int[] mem=new int[1000];
    int ic=0;
    Stack<BlocF> p=new Stack<BlocF>();
    while(true){
      switch(ic++){
        case 0: mem[0]=4;
        case 1: p.push(new BlocF(ic,mem[0]));
                ic=500; break;
        case 2: System.out.println("resultat␣:␣"
                +p.pop().getVal()); break;
        case 3: System.exit(0);
        case 50: p.peek().setVar1(1); break;
        case 51: p.peek().setVar2(1); break;
        case 52: if(p.peek().getVar1()
                >p.peek().getArg()) ic+=3; break;
        case 53: p.peek().setVar2(
                p.peek().getVar2()
                *p.peek().getVar1()); break;
        case 54: p.peek().setVar1(
                p.peek().getVar1()+1); break;
        case 55: ic-=4; break;
        case 56: p.peek().setVal(
                p.peek().getVar2()); break;
        case 57: ic=p.peek().getAdr(); break;
      }
    }
  }
}
```




```
import java.util.*;
class FactoTraduit{
  public static void main(String[] args){
    int[] mem=new int[1000];
    int ic=0;
    Stack<BlocF> p=new Stack<BlocF>();
    while(true){
      switch(ic++){
        case 0: mem[0]=4;
        case 1: p.push(new BlocF(ic,mem[0]));
                ic=500; break;
        case 2: System.out.println("resultat : "
                +p.pop().getVal()); break;
        case 3: System.exit(0);
        case 50: p.peek().setVar1(1); break;
        case 51: p.peek().setVar2(1); break;
        case 52: if(p.peek().getVar1()
                >p.peek().getArg()) ic+=3; break;
        case 53: p.peek().setVar2(
                p.peek().getVar2()
                *p.peek().getVar1()); break;
        case 54: p.peek().setVar1(
                p.peek().getVar1()+1); break;
        case 55: ic-=4; break;
        case 56: p.peek().setVal(
                p.peek().getVar2()); break;
        case 57: ic=p.peek().getAdr(); break;
      }
    }
  }
}
```




```
import java.util.*;
class FactoTraduit{
  public static void main(String[] args){
    int[] mem=new int[1000];
    int ic=0;
    Stack<BlocF> p=new Stack<BlocF>();
    while(true){
      switch(ic++){
        case 0: mem[0]=4;
        case 1: p.push(new BlocF(ic,mem[0]));
                ic=500; break;
        case 2: System.out.println("resultat : "
                +p.pop().getVal()); break;
        case 3: System.exit(0);
        case 50  p.peek().setVar1(1); break;
        case 51  p.peek().setVar2(1); break;
        case 52  if(p.peek().getVar1()
                >p.peek().getArg()) ic+=3; break;
        case 53  p.peek().setVar2(
                p.peek().getVar2()
                *p.peek().getVar1()); break;
        case 54  p.peek().setVar1(
                p.peek().getVar1()+1); break;
        case 55  ic-=4; break;
        case 56  p.peek().setVal(
                p.peek().getVar2()); break;
        case 57  ic=p.peek().getAdr(); break;
      }
    }
  }
}
```