

# FIBONACCI series

## Machine code

```
00004033
00000093
00100113
00100193
00100213
00000293
00a00313
00000393
00418c63
00110133
404181b3
00229393
0023a023
00420a63
```

## C-code

```
#include <iostream>
using namespace std;
int main()
{ int x = 0;
  int y = 1;
  int sel = 1;
  for (int i = 0; i < 10; i++)
  {
    if (sel == 1)
    { x = x + y;
      sel = 2;
      cout << x << endl; }
    else
    { y = x + y;
      sel = 1;
      cout << y << endl;}
  }
}
```

## Assembly code

```
0: xor x0,x0,x0    # reference register, always = 0
4: addi x1,x0,0    # data register (containing one of the last two values in FIBONACCI series)
8: addi x2,x0,1    # data register (containing the other value)
12: addi x3,x0,1    # selector (select the oldest register from R1 and R2 that had been refreshed)
16: addi x4,x0,1    # always constant and = 1
20: addi x5,x0,0    # loop counter
24: addi x6,x0,10   # total number of loops
28: addi x7,x0,0    # address of the data memory that will receive the latest evaluated result at
loop:
32: beq x3,x4,eq    # you can understand the code well from the attached c-code
36: add x2,x2,x1
40: sub x3,x3,x4
44: slli x7,x5,2
48: sw x2,0(x7)
52: beq x4,x4,endloop
eq:
56: add x1,x1,x2
60: add x3,x3,x4
64: slli x7,x5,2
68: sw x1,0(x7)
endloop:
72: addi x5,x5,1
76: blt x5,x6,loop
80: halt           # you can make it to prevent PC from increasing after the code had been finished
                  # But it is optional
```