The Fibonacci Sequence

1. `Fib1` is a recursive program for computing the N^{th} term of the Fibonacci sequence. This algorithm is $O(2^N)$.

```
#include <stdio.h>

main()
{
   int N;
   printf("Enter a positive integer N:\n");
   scanf("%d",&N);
   printf("The %d th Fibonacci Number is F%d = %d\n",N,N,Fib1(N));
}

int Fib1(int n)
{
   if (n < 2)
       return(n);
   else
       return(Fib1(n-1)+Fib1(n-2));
}
```

2. `Fib2` is a dynamic programming algorithm for computing the N^{th} term of the Fibonacci sequence. This algorithm is $O(N)$.

```
#include <stdio.h>

main()
{
   int N;
   printf("Enter a positive integer N:\n");
   scanf("%d",&N);
   printf("The %d th Fibonacci Number is F%d = %d\n",N,N,Fib2(N));
}

int Fib2(int n)
{
   int k,u=0,v=1,w=1;
   if (n < 2)
       return(n);
   else if (n == 2)
       return(1);
   else if (n > 2)
   {
       for (k = 3; k <= n; k++)
       {
            u = v; v = w;
            w = u+v;
       }
       return(w);
   }
}
```

3. `Fib3` is a divide–and–conquer program for computing the N^{th} term of the Fibonacci sequence. This algorithm is $O(\log N)$.

```
#include <stdio.h>

main()
{
   int N;
   printf("Enter a positive integer N:\n");
   scanf("%d",&N);
   printf("The %d th Fibonacci Number is F%d = %d\n",N,N,Fib3(N));
}

int Fib3(int N)
{
   int n = N;
   int i = 0, j = 1;
   int a1 = 1, a2 = 1;
   int b1 = 1, b2 = 0;
   int r,s,r1,r2,s1,s2;
   while (n > 0)
   {
        if (n%2 == 1)
        {
            r = i;
            s = j;
            i = a1*r + a2*s;
            j = b1*r + b2*s;
        }
        r1 = a1*a1 + a2*b1;
        r2 = a1*a2 + a2*b2;
        s1 = b1*a1 + b2*b1;
        s2 = b1*a2 + b2*b2;
        a1 = r1;
        a2 = r2;
        b1 = s1;
        b2 = s2;
        n = n/2;
   }
   return(i);
}
```