|  |
| --- |
| #include<iostream>  using namespace std;  // 递归求斐波拉契数列  int fib(int n){  if (n <= 2){  return 1;  }else {  return fib(n-1) + fib(n-2);  }  }  int main(){  int fibValue = fib(5);  cout<<fibValue<<endl;  return 0;  } |
|  |

|  |
| --- |
| // （2）递推（循环）求斐波拉契数列  int fib2(int n){  int arr[n];  arr[0] = 1;  arr[1] = 1;  // 循环求解  for(int i = 2;i<n;i++){  arr[i] = arr[i-1] + arr[i-2];  }  return arr[n-1];  } |
|  |

|  |
| --- |
| #include<iostream>  using namespace std;  // (1)递归求斐波拉契数列  int fib1(int n){  // 程序出口  if (n <= 2){  return 1;  }else {  // 递归表达式  return fib1(n-1) + fib1(n-2);  }  }  // （2）非递归，递推（循环）求斐波拉契数列  int fib2(int n){  int arr[n];  arr[0] = 1;  arr[1] = 1;  // 循环求解  for(int i = 2;i<n;i++){  arr[i] = arr[i-1] + arr[i-2];  }  return arr[n-1];  }  // (3) 尾递归进行递归优化 v1收集上一次的返回结果，v2进行计算  int fib3(int n,int v1,int v2){  if (n == 1){  return v1;  }  if (n == 2){  return v2;  }  return fib3(n-1,v2,v1+v2);  }  int main(){  int v1 = fib1(5);  int v2 = fib2(6);  int v3 = fib3(7,1,1);  cout<<v1<<endl;  cout<<v2<<endl;  cout<<v3<<endl;  return 0;  } |
|  |