**Nth term in the series**

**Consider the below series : 0,0,2,1,4,2,6,3,8,4,10,5,12,6,14,7,16,8 This series is a mixture of 2 series all the odd terms in this series form even numbers in ascending order and every even terms is derived from the previous term using the formula (x/2) .Write a program to find the nth term in this series. The value n in a positive integer that should be read from STDIN the nth term that is calculated by the program should be written to STDOUT. Other than the value of the nth term no other characters /strings or message should be written to STDOUT.**

**For example if n=10,the 10 th term in the series is to be derived from the 9th term in the series. The 9th term is 8 so the 10th term is (8/2)=4. Only the value 4 should be printed to STDOUT. You can assume that the n will not exceed 20,000.**

**Input (stdin)**

**10**

**Output (stdout)**

**4**

int main() {

int n;

scanf("%d", &n);

if(n % 2 == 1)

{

int a = 1;

int r = 2;

int term\_in\_series = (n+1)/2;

int res = 2 \* (term\_in\_series - 1);

printf("%d ", res);

}

else

{

int a = 1;

int r = 3;

int term\_in\_series = n/2;

int res = term\_in\_series - 1;

printf("%d ", res);

}

return 0;

}

**Nth term in the fibonacci series**

**Look at the series below:1,2,3,5,8,13,21,34,55,89,144,233,377,610,987,.....This series is formed as below: 1.term(1)=1 2.term(2)=2 3.term(N)=term(N-1)+term(N-2)for N>2 Write a program to find the Nth term in this series The value N is a positive integer that should be read from STDIN. The Nth term that is calculated by the program should be written to STDOUT, other than the value of nth term no other characters /strings and messages should be written to STDOUT.**

**Sample Input:**

**15**

**Sample Output:**

**987**

#include<stdio.h>

int main()

{

//Type your code here

int t1=1,t2=2,t3,i,n;

scanf("%d",&n);

if(n==1)

printf("1");

else if(n==2)

printf("2");

else{

for(i=3;i<=n;i++){

t3=t1+t2; //3

t1=t2; //t1=2

t2=t3; //t2=3

}

printf("%d",t3);

}

return 0;

}

**Nth term in the series**

**Consider the below series : 0,0,2,1,4,2,6,3,8,4,10,5,12,6,14,7,16,8 This series is a mixture of 2 series all the odd terms in this series form even numbers in ascending order and every even terms is derived from the previous term using the formula (x/2) .Write a program to find the nth term in this series. The value n in a positive integer that should be read from STDIN the nth term that is calculated by the program should be written to STDOUT. Other than the value of the nth term no other characters /strings or message should be written to STDOUT.**

**For example if n=10,the 10 th term in the series is to be derived from the 9th term in the series. The 9th term is 8 so the 10th term is (8/2)=4. Only the value 4 should be printed to STDOUT. You can assume that the n will not exceed 20,000.**

**Input (stdin)**

**10**

**Output (stdout)**

**4**

int main() {

int n;

scanf("%d", &n);

if(n % 2 == 1)

{

int a = 1;

int r = 2;

int term\_in\_series = (n+1)/2;

int res = 2 \* (term\_in\_series - 1);

printf("%d ", res);

}

else

{

int a = 1;

int r = 3;

int term\_in\_series = n/2;

int res = term\_in\_series - 1;

printf("%d ", res);

}

return 0;

}

**Q.**

**For Example, consider the given series: 1, 2, 1, 3, 2, 5, 3, 7, 5, 11, 8, 13, 13, 17, …**

**This series is a mixture of 2 series – all the odd terms in this series form a Fibonacci series and all the even terms are the prime numbers in ascending order. Now write a program to find the Nth term in this series.**

**Input: 14 Output: 17**

#include<stdio.h>

#define MAX 1000

void fibonacci(int n)

{

int i, t1 = 0, t2 = 1, nextTerm;

for (i = 1; i<=n; i++)

{

nextTerm = t1 + t2;

t1 = t2;

t2 = nextTerm;

}

printf("%d", t1);

}

void prime(int n)

{

int i, j, flag, count =0;

for (i=2; i<=MAX; i++)

{

flag = 0;

for (j=2; j<i; j++)

{

if(i%j == 0)

{

flag = 1;

break;

}

}

if (flag == 0)

if(++count == n)

{

printf("%d", i);

break;

}

}

}

int main()

{

int n;

scanf("%d", &n);

if(n%2 == 1)

fibonacci (n/2 + 1);

else

prime(n/2);

return 0;

}