

# acm.mipt.ru

олимпиады по программированию на Физтехе

## Раздел «Парадигмы» . FibonacciNumbers :

### Содержание

- [Язык Mathematica](#)
- [Tcl](#)
- [Ruby](#)
- [Язык Perl](#)
- [Язык C](#)
- [Язык FreePascal](#)
- [Язык Python](#)
- [Язык Ocaml](#)
- [Язык Lisp](#)

### Язык Mathematica

- Вариант 1 (медленно работает)

```
F[n_] := F[n-1] + F[n-2];
F[0] = F[1] = 1;
F[1000]
```

- Вариант 2 (быстро работает)

```
F[n_] := (F[n] = F[n-1] + F[n-2]);
F[0] = F[1] = 1;
F[1000]
```

### Tcl

```
#!/usr/local/bin/tclsh
proc fib {n} {
    if {$n < 2} {
        return 1
    } else {
        return [expr {[fib [expr {$n-2}]] + [fib [expr {$n-1}]]}]
    }
}
set N [lindex $argv 0]
if {$N < 1} { set N 1 }
puts [fib $N]
```

### Ruby

```
#!/usr/local/bin/ruby
def fib(n)
    if n < 2 then
        1
    else
        fib(n-2) + fib(n-1)
    end
end

N = Integer(ARGV.shift || 1)
puts fib(N)
```

### Язык Perl

```
use strict;
use integer;
```

Поиск

Поиск

Раздел  
«Парадигмы»

Инструменты:

Поиск  
Изменения  
Index  
Статистика

Разделы

Информация  
Алгоритмы  
Язык Си  
Язык Ruby  
Язык  
Ассемблера  
E! Judge  
Парадигмы  
Образование  
Сети  
Objective C

Logon>>

```

sub fib {
    return $_[0] < 2 ? 1 : fib($_[0]-2) + fib($_[0]-1);
}

my $N = ($ARGV[0] < 1) ? 1 : $ARGV[0];
my $fib = fib($N);
print "$fib\n";

```

## Язык C

```

#include <stdio.h>
#include <stdlib.h>
unsigned long
fib(unsigned long n) {
    if (n < 2)
        return(1);
    else
        return(fib(n-2) + fib(n-1));
}
int
main(int argc, char *argv[]) {
    int N = ((argc == 2) ? atoi(argv[1]) : 1);
    printf("%ld\n", fib(N));
    return(0);
}

```

## Язык FreePascal

```

program fibo;
uses SysUtils;

function fib(N : integer) : longint;
begin
    if N < 2 then fib := 1
    else fib := fib(N-2) + fib(N-1);
End;

var
    NUM : integer;
    f : longint;

begin
    if ParamCount = 0 then
        NUM := 1
    else
        NUM := StrToInt(ParamStr(1));

    if NUM < 1 then NUM := 1;
    f := fib(NUM);
    WriteLn( IntToStr(f) );
end.

```

## Язык Python

```

#!/usr/local/bin/python
import sys
def fib(n):
    if (n < 2):
        return(1)
    return( fib(n-2) + fib(n-1) )

def main():
    N = int(sys.argv[1])
    #sys.setrecursionlimit(3000)

```

```
print fib(N)

main()
```

## Язык Ocaml

```
let rec fib n =
  if n < 2 then 1
  else fib (n - 2) + fib (n - 1)

let _ =
  let n =
    try int_of_string Sys.argv.(1)
    with Invalid_argument _ -> 1 in
  Printf.printf "%d\n" (fib n)
```

## Язык Lisp

BeautifierPlugin Error: Unable to handle "lisp" syntax.

Copyright © 2003-2022 by the contributing authors.