In [1]:
```
# Problem 1
reset()
def problem1(f):
    show(f)
    print('Rational: ', len(f.roots(ring=QQ)))
    print('Irrational: ', len(f.roots(ring=RR)) - len(f.roots(ring=QQ)))
    print('Real: ', len(f.roots(ring=RR)))
    print('Complex: ', len(f.roots(ring=CC)))

f = x^10+x^9-7*x^8-8*x^7+6*x^6+14*x^5+22*x^4+8*x^3-7*x^2-15*x-15
problem1(f)
```

$$x^{10} + x^9 - 7\,x^8 - 8\,x^7 + 6\,x^6 + 14\,x^5 + 22\,x^4 + 8\,x^3 - 7\,x^2 - 15\,x - 15$$

```
Rational:  2
Irrational:  4
Real:  6
Complex:  10
```

In [2]:
```
# Problem 2
reset()
def problem2(k, a, b):
    counter1 = 0
    counter2 = 0
    for i in Primes():
        if(i%k == a):
            counter1+=1
        elif(i%k == b):
            counter2+=1

        if(counter1 > counter2):
            return i

# print(problem2(4,1,3))
print(problem2(5,1,2))
```

```
6791
```

In [6]:
```
# Problem 3
reset()

def random_poly(n):
    h=0
    for i in range(n):
        h = h + ZZ.random_element(-1,1)*x^i
    return h

def problem3(n,M):
    counter = 0
    for i in range(M):
        f = random_poly(n)
        R = QQ[x] # This is polynomials w coeffs in Q with variable x
        r = R(f)
        if(r.is_irreducible()):
            counter +=1
    return counter

problem3(11,1000)
```

Out[6]:
```
322
```

In [4]:
```
# Problem 4
reset()

# def _problem4(n, resultSoFar):
#     if(n == 0):
#         print(n)
#         return resultSoFar
#     else:
#         print(n)
#         resultSoFar = 1/(n**2) + resultSoFar
#         return _problem4(n-1, resultSoFar)


# def problem4(n):
#     return numerical_approx(_problem4(n, 0))

def problem4(n):
    result = 0
    for i in range(1,n):
        result = result + 1/(i**2)
    return numerical_approx(result)

t = cputime()
print(problem4(10000))
print('cputime: ', cputime(t))
```

```
1.64483406184806
cputime:  0.20400000000000063
```