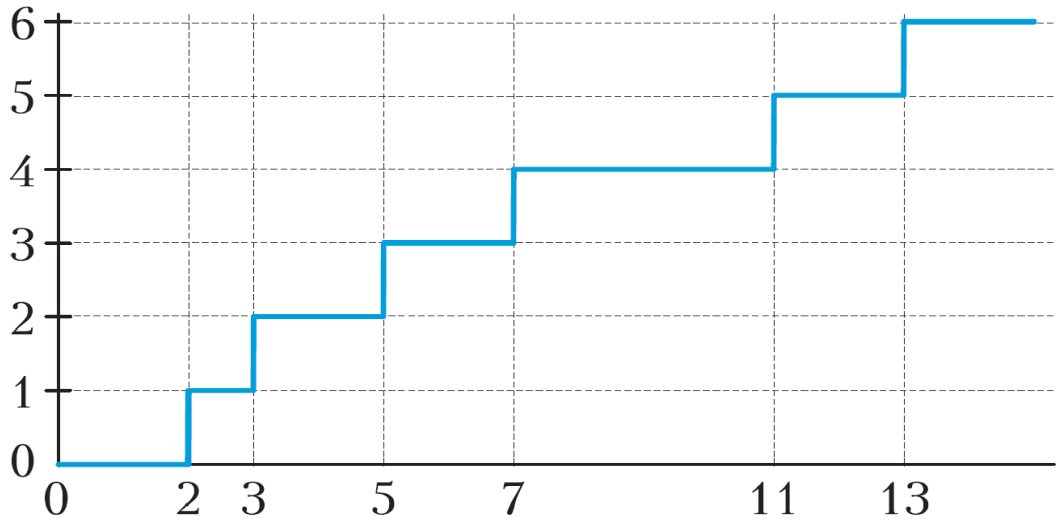

Figure №1. **Graph of the function $\pi(x)$ in the interval $0 \leq x \leq 15$.**

n	C_n^2	Prime	C_{n+1}^2
2	1	2	3
3	3	5	6
4	6	7	10
5	10	11	15
6	15	17	21
7	21	23	28
8	28	29	36
9	36	37	45
10	45	47	55

Table № 1. **First Conjecture's check for** $n \in [2; 10]$

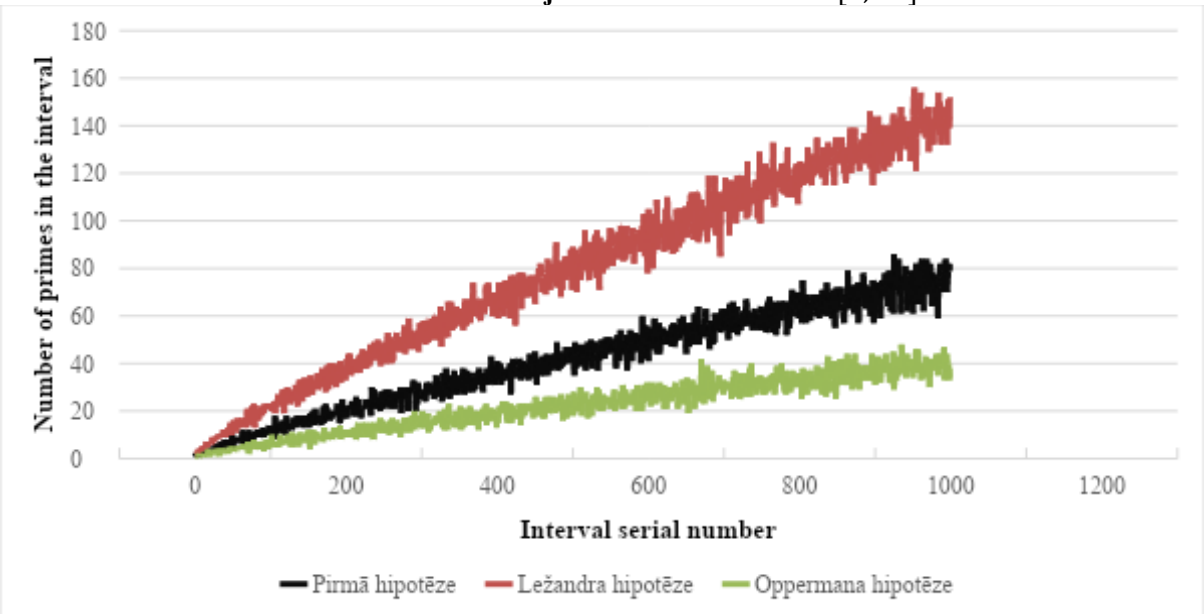

Figure №2. **Dependence of the number of primes in the interval on the index of the interval.**

n	x	C_{n+1}^2	Prime	C_{n+2}^2	C_{x+1}^2	Prime	$C_{x+\frac{1}{\alpha_1}+1}^2$
1	0,91	1	2	3	0,87	2	2,56
2	1,82	3	5	6	2,56	3	5,08
3	2,73	6	7	10	5,08	7	8,43
4	3,64	10	11	15	8,43	11	12,60
5	4,55	15	17	21	12,60	13	17,60
6	5,45	21	23	28	17,60	19	23,43
7	6,36	28	29	36	23,43	29	30,08
8	7,27	36	37	45	30,08	31	37,56
9	8,18	45	47	55	37,56	41	45,87
10	9,09	55	59	66	45,87	47	55,00

Table № 2. **Second Conjectures check and comparison with First Conjecture for** $n \in [1; 10]$

n	x	C_{n+1}^2	Prime	C_{n+2}^2	C_{x+1}^2	Prime	$C_{x+\frac{1}{\alpha_1}+1}^2$
1	0,781	1	2	3	0,695	2	2,000
2	1,562	3	5	6	2,000	3	3,914
3	2,342	6	7	10	3,914	5	6,438
4	3,123	10	11	15	6,438	7	9,572
5	3,904	15	17	21	9,572	11	13,315
6	4,685	21	23	28	13,315	15	17,668
7	5,465	28	29	36	17,668	19	22,631
...							
999	779,996	499500	498521	500500	304586,589	304597	305196,287
1000	780,776	500500	499507	501501	305196,287	305209	305806,594

Table № 3. **First Conjecture and Second Conjecture founded intervals comparation for** $n \in [1; 1000]$

i	α_i	i	α_i	i	α_i	i	α_i	i	α_i
1	1,28078	21	0,12744	41	0,08434	61	0,06652	81	0,05585
2	0,84126	22	0,12453	42	0,08182	62	0,06567	82	0,05577
3	0,56050	23	0,11926	43	0,08155	63	0,06512	83	0,05510
4	0,42063	24	0,11845	44	0,07926	64	0,06432	84	0,05485
5	0,35009	25	0,11247	45	0,07854	65	0,06406	85	0,05413
6	0,28042	26	0,11045	46	0,07715	66	0,06292	86	0,05344
7	0,27064	27	0,10792	47	0,07670	67	0,06139	87	0,05314
8	0,23339	28	0,10498	48	0,07605	68	0,06083	88	0,05292
9	0,22761	29	0,10442	49	0,07520	69	0,06050	89	0,05270
10	0,20392	30	0,10333	50	0,07397	70	0,06017	90	0,05207
11	0,19278	31	0,09976	51	0,07358	71	0,06007	91	0,05193

12	0,18040	32	0,09609	52	0,07188	72	0,05892	92	0,05105
13	0,17251	33	0,09566	53	0,07116	73	0,05862	93	0,05073
14	0,16557	34	0,09359	54	0,06929	74	0,05852	94	0,05034
15	0,15563	35	0,09320	55	0,06913	75	0,05794	95	0,05028
16	0,14884	36	0,09128	56	0,06833	76	0,05775	96	0,04997
17	0,14576	37	0,08913	57	0,06786	77	0,05766	97	0,04979
18	0,14429	38	0,08810	58	0,06726	78	0,05710	98	0,04972
19	0,14147	39	0,08616	59	0,06696	79	0,05655	99	0,04942
20	0,12949	40	0,08585	60	0,06681	80	0,05611	100	0,04901

Table № 4. **First 100 α_i values**

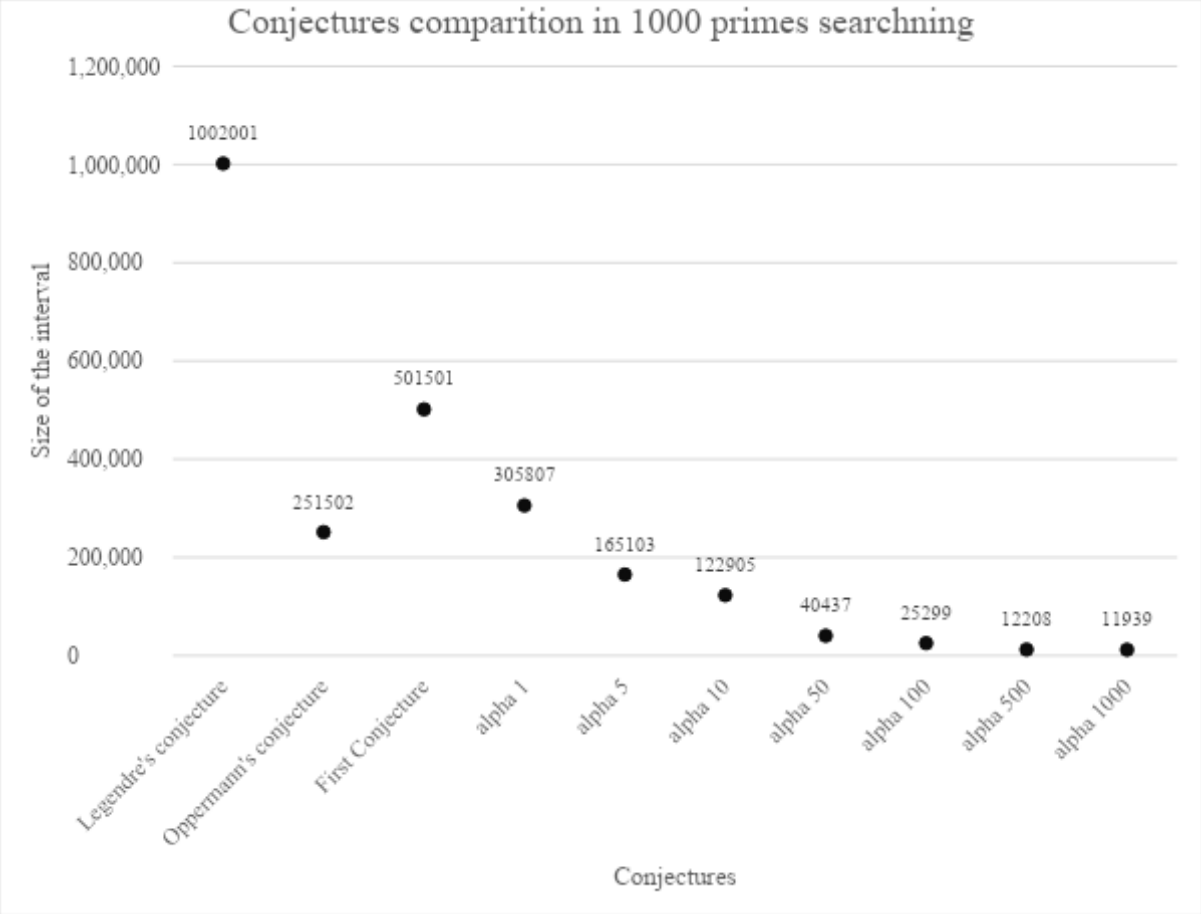

Figure №3. **Comparison of dimensions of the intervals in which 1000 primes are guaranteed for each conjecture.**

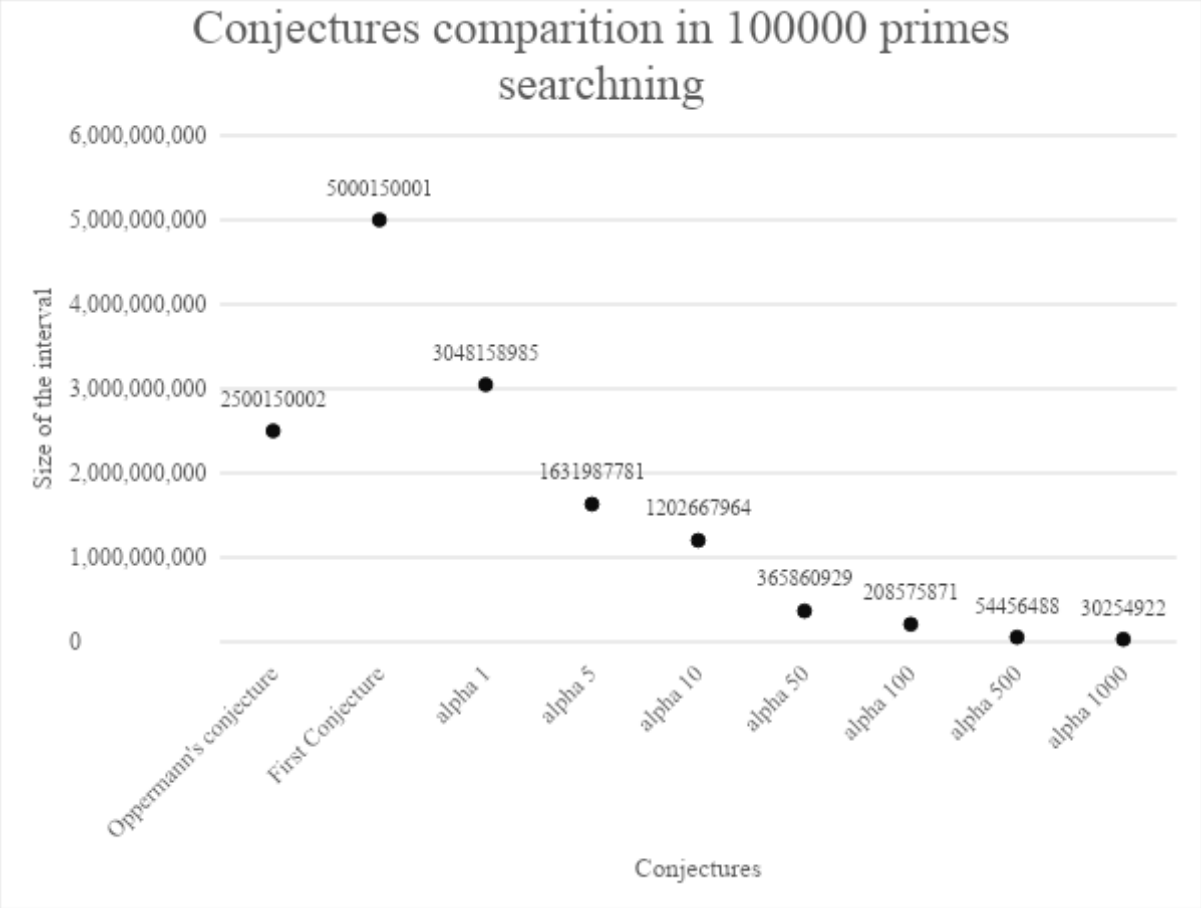

Figure №4. **Comparison of dimensions of the intervals in which 100000 primes are guaranteed for each conjecture.**

n	a_n	Primes	a_{n+1}
1	1	2	3
2	3	5	6
3	6	7	8
4	8	11	12
5	12	13	17
6	17	19	21
7	21	23	27
8	27	29	34
9	34	37	40
10	40	41	48
...			
999	333334	333337	334000
1000	334000	334021	334668

Table № 5. **Fourth Conjectures check for** $n \in [1; 1000]$

Addendum

First addendum. α_i searching programm.

```
import math



def IsPrime(num): # Function which returns True if the given number is prime and False if not #
    flag = True

    if(num == 1):
        return False
    if(num == 2):
        return True

    for i in range(2 , math.floor(pow(num, 0.5)) + 2):
        if(num % i == 0):
            flag = False
            break
    return flag


f1 = 0.001
f2 = 200
f3 = 6

m = 0 # Variables which help to find alpha's values #
s = 0

alpha = f1

index = int(input())

current_digit = 1
digit_amount = 0


while (digit_amount < f3):

    for n in range(1,f2 + 1):

        print(n,": alpha =", round(alpha, current_digit))
        x = n / alpha
        p_amount = 0
```

```
41.
42.         for i in range(math.ceil(x*(x+1)/2 + 0.0000000001), math.ceil((x+1/alpha)*(x+1+1/alpha)/2 +
    0.0000000001)):
43.             if IsPrime(i):
44.                 p_amount += 1
45.
46.         print(n, ": ", round(x*(x+1)/2, 2), p_amount, round((x+1/alpha)*(x+1+1/alpha)/2, 2))
47.     if(p_amount < index):
48.         alpha -= (10 ** (-current_digit))
49.         current_digit += 1
50.         digit_amount += 1
51.         break
52.     if(n == f2 or p_amount > 10 * index):
53.         alpha += (10 ** (-current_digit))
54.         break
55.
56. alpha += 1.5 * (10 ** (-f3))
57.
58. for n in range(1,f2 + 1):
59.
60.     x = n / alpha
61.     p_amount = 0
62.
63.     for i in range(math.ceil(x*(x+1)/2 + 0.0000000001), math.ceil((x+1/alpha)*(x+1+1/alpha)/2 +
    0.0000000001)):
64.         if IsPrime(i):
65.             p_amount += 1
66.
67.     print(n, ": ", round(x*(x+1)/2, 2), p_amount, round((x+1/alpha)*(x+1+1/alpha)/2, 2))
68.
69.     if(p_amount < index):
70.         m = n + 1
71.         s = round((x+1/alpha)*(x+1+1/alpha)/2)
72.         break
73.     if(n == f2):
74.         print("Error", alpha)
75.         break
76.
77. alpha -= 1.5 * (10 ** (-f3))
78. print("\n\n\n")
79. print("alpha(",index,")=",round(alpha,f3), "\n")
80.
81. print(m,"+ √",(m*m + 8*m*m*s),)
```

```
print("_________")
print(" ",4*s," ")
```

Second Addendum. Fourth Conjecture's checking programm.

```
import math


def isPrime(num): # Function which returns True if the given number is prime and False if not #
flag = True

    for i in range(2 , math.floor(pow(num, 0.5)) + 2):
        if(num % i == 0):
            flag = False
            break
    return flag



Check_from = 6 # n value from which testing will be started #
Check_to = 3000001 # n value reaching which testing will be finished #


A = [] # Making an array filled with zeros #
for k in range(Check_to + 1):
        A.append(0)

A[1] = 1 # Setting first five A values #
A[2] = 3
A[3] = 6
A[4] = 8
A[5] = 12



for i in range(Check_from, Check_to + 1):
    A[i] = 2*A[i-1] - A[i-2] + A[i-3] - 2*A[i-4] + A[i-5]


    p = 0

    for j in range(A[i-1] + 1, A[i] + 1):
        if(isPrime(j) == True):
```

```
39.             p = j
40.             break
41.     if(p == 0):
42.         print(i-1, ": ", A[i-1], p, A[i])
43.         break
44. print(i-1, ": ", A[i-1], p, A[i])
```