



# IJRASET

International Journal For Research in  
Applied Science and Engineering Technology



---

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

---

**Volume:** 10    **Issue:** XII    **Month of publication:** December 2022

**DOI:** <https://doi.org/10.22214/ijraset.2022.48005>

[www.ijraset.com](http://www.ijraset.com)

Call:  08813907089

E-mail ID: [ijraset@gmail.com](mailto:ijraset@gmail.com)

# Estimation of Range of Appearance in Heptadecagonal Numbers

S. Saranya<sup>1</sup>, V. Pandichelvi<sup>2</sup>

<sup>1</sup> Research Scholar, PG & Research Department of Mathematics, Urumu Dhanalakshmi College, Trichy., (Affiliated to Bharathidasan University).

<sup>2</sup> Assistant Professor, PG & Research Department of Mathematics, Urumu Dhanalakshmi College, Trichy, (Affiliated to Bharathidasan University).

**Abstract:** In this document, the range of appearance  $I(r)$  of a positive integer  $r$  in heptagonal numbers and some results by comparing  $I(r)$  with Legendre symbol,  $p$  – adic range and prime conjunction function of  $r$  are accessible. Since  $I(r)$  does not possess a meticulous prototype, all the results are confirmed by Java program for all-natural numbers  $r \in \mathbb{N}$ .

**Keywords:** Heptadecagonal number, range of appearance,  $p$  – adic range

## I. INTRODUCTION

“In 1796, an arithmeticians Gauss showed that a regular polygon of 17 sides can be created using a scale and compass by screening that a primitive 17th root of unity can be found by solving a succession of quadratic equations over the rationals. A heptadecagram is a star polygon with 17-sides.

The sequence of Heptadecagonal number is provoked by  $H(m) = \frac{m(15m-13)}{2}$ ,  $m \in \mathbb{N}$  [1-4]. Many prominent arithmeticians have apportioned with concepts of divisibility of Fibonacci numbers. In [5], the author pondered the equation  $Z(n) = (2 - \frac{1}{K})n$  involving order of appearance in Fibonacci numbers. In this framework, one may grasp [6-10]. In this document, the range of appearance  $I(r)$  of  $r \in \mathbb{N}$  in the pattern of heptagonal numbers and few results by measure up to  $I(r)$  with Legendre symbol,  $p$  – adic range and prime conjunction functions are offered. Since  $I(r)$  does not have a particular model, each and every result are entrenched by Java program for all values of  $r \in \mathbb{N}$ .

### 1) Definition

“A quadratic residue modulo  $n$  is an integer  $t$  such that it is congruent to square of a number. That is there can exists an integer  $Y$  such that  $Y^2 \equiv t \pmod{n}$ ” [1].

### 2) Definition

In [2], The Legendre symbol is a number hypothetical function  $\left(\frac{r}{p}\right)$  which is defined by

$$\left(\frac{r}{p}\right) = \begin{cases} 0, & \text{if } \frac{p}{r} \\ 1, & \text{if } r \text{ is a quadratic residue modulo } p \\ -1, & \text{if } r \text{ is a quadratic nonresidue modulo } p \end{cases}$$

### 3) Definition

“The greater power of a prime number  $p$  which divides a positive integer  $r$  is called  $p$  – adic range of  $r$  and it is symbolized by  $d_p(r)$ ”.

### 4) Definition

“The prime conjunction function of a positive integer  $r$  is the separate prime factors of  $r$  and it is designated by  $\Lambda(r)$ ”.

5) Definition

“The range of appearance of a positive integer  $r$  in the sequence of Heptadecagonal numbers signified by  $I(r)$  is defined as the lowest natural number  $m$  such that  $r|H(m)$ ” [5].

The values of  $I(r)$  for discrete values of  $r$  such that  $1 \leq r \leq 20$  are listed in the table below.

$r$	1	2	3	4	5	6	7	8	9	10	11	12
$I(r)$	1	3	3	3	5	3	6	3	9	15	6	3
$r$	13	14	15	16	17	18	19	20	21	22	23	24
$I(r)$	13	7	15	3	2	27	11	35	6	11	7	3
$r$	25	26	27	28	29	30	31	32	33	34	35	36
$I(r)$	25	39	27	27	26	15	5	35	6	19	20	27
$r$	37	38	39	40	41	42	43	44	45	46	47	48
$I(r)$	28	11	39	35	20	27	41	11	45	7	4	3
$r$	49	50	51	52	53	54	55	56	57	58	59	60
$I(r)$	27	75	36	91	15	27	50	35	30	55	52	75
$r$	61	62	63	64	65	66	67	68	69	70	71	72
$I(r)$	9	31	27	35	65	39	50	19	30	20	34	99
$r$	73	74	75	76	77	78	79	80	81	82	83	84
$I(r)$	69	28	75	11	3	39	43	35	81	20	23	27
$r$	85	86	87	88	89	90	91	92	93	94	95	96
$I(r)$	70	43	84	83	78	135	13	99	36	4	30	99
$r$	97	98	99	100								
$I(r)$	72	27	72	75								

The range of appearance of a positive integer  $r$  in the already mentioned sequence of Heptadecagonal numbers for all other choices of  $r \in N$  are found out by the following Java program.

```
import java.util.Scanner;
import java.util.ArrayList; // import the ArrayList class
import java.util.Collections; // import Collections
class Main {
public static void main(String[] args) {
Scanner myObj = new Scanner(System.in); // Create a Scanner object
System.out.println("Enter the value of m:");
String m = myObj.nextLine(); // Read user input
int x = Integer.parseInt(m);
ArrayList < Integer > result = findM(x);
System.out.println(result);
System.out.println("Enter the value of r:");
String r = myObj.nextLine(); // Read user input
int y = Integer.parseInt(r);
ArrayList < Integer > resultN = findN(y);
System.out.println(resultN); }
//set values to m,r and temp as arraylist
// array list will help to have elements of any size.
static ArrayList < Integer > tmNet = new ArrayList < Integer > ();
static ArrayList < Integer > tnNet = new ArrayList < Integer > ();
static ArrayList < Integer > tmp = new ArrayList < Integer > ();
static int m;
static int r;
```

```

static int tm;
static int result = 0;
static ArrayList < Integer > findM(int val){
m = val;
for(int i = 1; i <= m; i ++){
result = i * ((15 * i) - 13)/2;
tmNet.add(result);
} return tmNet; //return m
} static ArrayList < Integer > findN(int val){
r = val;
for(int j = 1; j <= r; j ++){
for(int i = 1; i <= r; i ++){
tm = i * ((15 * i) - 13)/2;
int val2 = (int) Math.round(tm/j);
if (val2 != 0 && tm%j == 0){ //take value divisible by iterator and the value is > 0
tmp.add(i);
}}
if(!tmp.isEmpty()){
tnNet.add(Collections.min(tmp)); //take min val from list
tmp.removeAll(tmp); //reset array to every iteration
}}
return tnNet; //return r
}}

```

## II. ESTIMATION OF RANGE OF APPEARANCE IN HEPTADECAGONAL NUMBERS:

Some results based on the range of appearance of a positive integer  $r$  in the above held sequence of Heptadecagonal numbers for numerous selections of a natural number  $r$  are explored and are verified by Java program.

### A. Result 1

Let  $n$  be a positive integer of the form  $r = p^x$ , where  $p$  is a prime number.

- i. If  $p = 2$ , then  $I(2^x) \leq 3 \cdot (2^{x-1})$ , for  $x \geq 1$ .
- ii. If  $p = 3$ , then  $I(3^x) \leq 3^x$ , for  $x \geq 1$ .
- iii. If  $p = 5$ , then  $I(5^x) \leq 5^x$ , for  $x \geq 1$ .
- iv. If  $p > 5$ , then  $I(p^x) \leq \left(p - \left(\frac{2}{p}\right)\right) \cdot p^{x-1}$ , for  $x \geq 1$  where  $\left(\frac{r}{p}\right)$  is a legendre symbol

Java program for monitoring this result is specified below.

```

import java.util.Scanner;
import java.util.ArrayList; // import the ArrayList class
import java.util.Collections; // import Collections
import java.lang.Math;
import java.util.*;
class Main {
public static void main(String[] args) {
Scanner myObj = new Scanner(System.in); // Create a Scanner object
System.out.println("Enter the value of r:");
String r = myObj.nextLine(); // Read user input
int y = Integer.parseInt(r);
ArrayList < Integer > resultN = findN(y);
System.out.println("r: " + getRepValues());
}
}

```

```

System.out.println("I(r) " + resultN); }
//set values to m,r and temp as arraylist
// array list will help to have elements of any size.
static ArrayList < Integer > tnNet = new ArrayList < Integer > ();
static ArrayList < Integer > tmp = new ArrayList < Integer > ();
static int m; static int r; static int tm;
static int result = 0;
static ArrayList < Integer > rep = new ArrayList < Integer > ();
static ArrayList < Integer > findN(int val){
r = val;
for(int j = 1; j ≤ r; j ++){
for(int i = 1; i ≤ r; i ++){
tm = i * ((15 * i) - 13)/2;
int val2 = (int) Math.round(tm/j);
if (val2 != 0 && tm%j == 0 && j%2 == 0 && (int)(Math.ceil((Math.log(j) / Math.log(2)))) ==
(int)(Math.floor((Math.log(j) / Math.log(2))))){ //take value divisible by iterator and the value is
> 0, is divisible by 2 and is also power of 2.
rep.add(j);
tmp.add(i); }}
if(!tmp.isEmpty()){
tnNet.add(Collections.min(tmp)); //take min val from list
tmp.removeAll(tmp); //reset array to every iteration
}} return tnNet; //return r
} static ArrayList < Integer > getRepValues(){
Set < Integer > unique_ = new HashSet <> (rep);
rep.clear();
rep.addAll(unique_);
return rep; }}

```

**B. Result 2**

Let  $r$  be an even integer and  $d_2(r)$ ,  $\wedge(r)$  are 2 – adic range of  $r$  and the prime conjunction function of  $r$  respectively.

i. If  $d_2(r) = 1$ , then

$$I(r) \leq \begin{cases} 2r, & \text{if } \wedge(r) = 2 \text{ and } r \nmid 5 \\ \frac{4r}{3}, & \text{if } \wedge(r) = 2 \text{ and } r \nmid 5 \\ 2 \left(\frac{5}{2}\right)^{\wedge(r) - \mathfrak{S}_r - 2} r, & \text{if } \wedge(r) > 2 \end{cases}$$

ii. If  $d_2(r) = 2$ , then

$$I(r) \leq \begin{cases} \frac{3r}{2}, & \text{if } \wedge(r) = 2 \text{ and } r \nmid 5 \\ \frac{7r}{6}, & \text{if } \wedge(r) = 2 \text{ and } r \nmid 5 \\ 3 \left(\frac{1}{2}\right)^{\wedge(r) - \mathfrak{S}_r - 2} r, & \text{if } \wedge(r) > 2 \end{cases}$$

iii. If  $d_2(r) = 3$ , then

$$I(r) \leq \begin{cases} \frac{5r}{2}, & \text{if } \wedge(r) = 2 \text{ and } r \nmid 5 \\ \frac{4r}{3}, & \text{if } \wedge(r) = 2 \text{ and } r \nmid 5 \\ 2 \left(\frac{1}{2}\right)^{\wedge(r) - \mathfrak{S}_r - 2} r, & \text{if } \wedge(r) > 2 \end{cases}$$

iv. If  $d_2(r) \geq 4, \wedge(r) = 2$ , then  $I(r) \leq \frac{7}{4} \left(\frac{2}{3}\right)^{\wedge(r) - \mathfrak{S}_r - 1} r$ , where  $\mathfrak{S}_r = \begin{cases} 0 & \text{if } r \nmid 5 \\ 1 & \text{if } r \mid 5 \end{cases}$

Java program for checking all the above four statements are laid down below.

```
import java.util.Scanner;
import java.util.ArrayList; // import the ArrayList class
import java.util.Collections; // import Collections
import java.lang.Math;
import java.util.*;
class Main {
public static void main(String[] args) {
Scanner myObj = new Scanner(System.in); // Create a Scanner object
System.out.println("Enter the value of p:");
String r = myObj.nextLine(); // Read user input
int y = Integer.parseInt(r);
boolean flag = false;
for (int i = 2; i <= y / 2; ++i) { // condition for nonprime number
if (y % i == 0) {
flag = true; break; }
}
if (!flag) {
int param = (int) Math.round(Math.pow(2, 4) * y);
System.out.println(y + " is a prime number.");
ArrayList < Integer > resultN = findN(param);
System.out.println("r: " + getRepValues());
System.out.println("I(r) " + resultN);
} else {
System.out.println(y + " is not a prime number. Please rerun the program and enter a prime number !!");
} // set values to m, n and temp as arraylist
// array list will help to have elements of any size.
static ArrayList < Integer > tnNet = new ArrayList < Integer > ();
static ArrayList < Integer > tmp = new ArrayList < Integer > ();
static ArrayList < Integer > finalresult = new ArrayList < Integer > ();
static int m;
static int r;
static double tm;
static int result = 0;
static ArrayList < Integer > rep = new ArrayList < Integer > ();
static ArrayList < Integer > findN(int val) {
r = val;
for (int j = 1; j <= r; ++j) {
for (int i = 1; i <= r; ++i) {
tm = i * ((15 * i) - 13) / 2;
int val2 = (int) Math.round(tm / r);
if (val2 != 0 && tm % r == 0) { // take value divisible by iterator and the value is > 0 and is integer value
rep.add(r); // n is a constant here so we add the calculated
tmp.add(i); //
}
}
if (!tmp.isEmpty()) {
tnNet.add(Collections.min(tmp)); // take min val from list
tmp.removeAll(tmp); // reset array to every iteration
}
}
if (!tnNet.isEmpty()) {
finalresult.add(Collections.min(tnNet)); // tnNet; // return r
}
```

```
}return finalresult;  
}static ArrayList < Integer > getRepValues(){  
Set < Integer > unique_ = new HashSet <> (rep);  
rep.clear();rep.addAll(unique_);  
return rep;}}
```

### III. CONCLUSION

In this paper, the range of appearance  $I(r)$  of a positive integer  $r$  in heptagonal numbers is defined and limited number of results consisting  $I(r)$ , legendre symbol,  $p$  – adic range and prime conjunction function of  $r$  are evaluated. Entire results are inveterate by Java program for all positive values of  $r$ . In this approach, one can define the range of appearance of an integer in other polygonal numbers and may analyse their results by comparing the values of  $I(r)$  with Jacobi symbol, quadratic residue and non-residue modulo  $r$ .

### REFERENCES

- [1] Ivan Niven, Hibert S, Zuckerman and Huge L, Montgomery, “An Introduction to the Theory of Numbers”, Fifth Edn., John Wiley & Sons Inc., 2004.
- [2] Carmichael, R.D., “The Theory of Numbers and Diophantine Analysis”, Dover Publication, New York (1959).
- [3] Mordell, L.J., “Diophantine Equations”, Academic press, London (1969).
- [4] John.H., Conway and Richard K.Guy, “The Book of Numbers”, Springer Verlag, NewYork,1995.
- [5] Eva Trojovs, “On the Diophantine Equation  $z(n) = \left(2 - \frac{1}{k}\right)n$  involving the order of appearance in the Fibonacci Sequence”, Mathematics 2020,1-8.
- [6] Trojovský, P. “The order of appearance of the sum and difference between two Fibonacci Numbers”,Asian-Eur.J.Math.2019,12,1950046.
- [7] Chung, C.L. “Some Polynomial Sequence Relations”. Mathematics 2019, 7,750.
- [8] Knott, R. “Fibonacci Numbers and the Golden Section”, The Mathematics Department of the University of Surrey, UK. Available online:[Http://www.maths.surrey.ac.uk/hostedsites/R.Knott/Fibonacci/fib.html](http://www.maths.surrey.ac.uk/hostedsites/R.Knott/Fibonacci/fib.html) . (accessed on 2 January 2020).
- [9] The Fibonacci Association, Official Website. Available online: <https://www.mathstat.dal.ca/fibonacci/>(accessed on 2 January 2020).
- [10] Kim, S. “The density of the terms in an elliptic divisibility sequence having a fixed G.C.D. with their indices”. J. Number Theory2020,207, 22–41.



10.22214/IJRASET



45.98



IMPACT FACTOR:  
7.129



IMPACT FACTOR:  
7.429



# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24\*7 Support on Whatsapp)