Report for Long Project 1

Integer arithmetic with arbitrarily large numbers

Implementation of Data Structures and Algorithms

By G16

Team Members:

Vasudev Karthik Ravindran

Ian Laurain

# Introduction:

The aim of this project is to develop a program that implements arithmetic with large integers, of arbitrary size. This document gives a detailed description of modules used in the project.

# Methodology:

The arithmetic operations on given large integers (in base 10) can be made effective if those integers are to be stored as polynomials of higher base and perform the same operations. Also if we are implementing improved versions of arithmetic operation, their performance can be enhanced further.

Here are the improved methodologies of multiple and divide operations:

1. Multiply – Here we used Divide and Conquer Approach to multiply two polynomials which improved the performance of multiplication from O(n^2) to O(n^log3).

DAC Multiplication Approach:

MultiplyDAC(p1, p2):

Given: two polynomials p1 and p2 in same base b

If any of the polynomials is zero, return zero.

Find the middle term of smaller degree polynomial of p1 and p2.

Split the polynomials p1 and p2 into two based on the middle term.

Let them be p1m, p1l and p2m, p2l.

The multiplication can now be represented as:

p1\*p2 = MultiplyDAC(p1m, p2m)\*base^(2\*k) +

(MultiplyDAC(p1m+p1l, p2m+p2l) - MultiplyDAC(p1m, p2m) - MultiplyDAC(p1l, p2l)) \*base^k +

MultiplyDAC(p1l, p2l)

Its time complexity is O(n^log3).

In term of performance, we could see a significant increase in performance as the number of digits of the input numbers get larger.

For e.g. here is the statistics:

|  |  |  |
| --- | --- | --- |
| No Of Digits of Input | DAC Execution Time in Seconds | Iterative Approach Execution Time in Seconds |
| 3000 | 0.424717045 | 0.483590947 |
| 10000 | 2.376695159 | 4.208171286 |
| 15000 | 4.491293127 | 9.144973603 |
| 20000 | 7.500100686 | 16.66279581 |
| 25000 | 10.99711814 | 25.92646536 |
| 30000 | 14.74797613 | 36.66056574 |
| 50000 | 38.03228729 | 102.8871312 |

1. Division Approach:

Conventional Polynomial Long division Method get quite cumbersome to implement. But the following Simpler DAC Approach is implemented for division operation which makes its time complexity in order of logn where n is the quotient;

DAC Approach:

Divide (p1, p2):

Polynomial start = 0;

Polynomial middle;

Polynomial end = p2;

While (start + 1 < end) {

Middle = p1 / 2;

If(p2 \* middle < 0)

Start = middle;

Else if (p2 \* middle > 0)

End = middle;

Else

Return middle;

}

Return start;

Similar DAC approach has been followed for square root, factorial and power functions as well.

Also as part of this Project, we have implemented Shunting Yard algorithm to parse the input instruction. The instructions are divided into four types: assignment, conditional loop, display and printlist. In order to execute the instruction, all of the instructions are stored in Arraylist as well as in HashMap for branching in case of conditional loop. In case of Data, all of the data are stored as Hashmap which maps the variable of the data with that of its actual value (BigNumber).

Before the instructions are executed, they are parsed using Tokenizer class which implements Shunting Yard Algorithm to convert the instruction into a postfix Expression queue.

Here is the high level flow:

1. Store all the instruction in memory.
2. Let current instruction be instruction at index 0. (Note: lineno is different from index).
3. While (current instruction != null) {

If current instruction is an assignment,

Get the postfix expression using tokenized class.

Execute the postfix expression (Processor.java)

Go to instruction at current instruction. Index + 1;

If current function is a conditional loop,

Check if the number is not zero, if the number is not zero, use the hashmap which maps the lineno to the actual instruction to retrieve the instruction at non zero line.

Go to instruction at lineno.

If current instruction is a display or printlist,

Execute it.

Go to instruction at current instruction. Index + 1;

}

# Test Results: (Base is set to 100)

Sample Test Case:

Input:

3 a=100!

2 a

Output:

93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518286253697920827223758251185210916864000000000000000000000000

3)

Input:

1 a=2

2 a=a+3\*4^2^3!

3 a

Output:

1020847100762815390390123822295304634370

4)

Input:

1 x=10

2 p=1

3 n=1

4 p=p\*x

5 x=x-n

6 x?4

7 p

Output:

3628800

5)

Input:

1 x=999

2 y=8

3 z=x+y

4 z

5 a=x^y

6 a

7 z)

Output:

1007

992027944069944027992001

100: [ (7, 0) , (10, 1) ]

6)

Input:

1 a=90569784495866770974195656280275310090138980613960953881501965823101

2 b=75040970647524038461398929683905540248523961720824412136973299943953

3 c=a-b

4 c

5 c=b-a

6 c

Output:

15528813848342732512796726596369769841615018893136541744528665879148

-15528813848342732512796726596369769841615018893136541744528665879148

7)

Input:

1 a=1000!

2 a)

Output:

100: [ (20, 124) , (-53, 125) , (54, 126) , (-23, 127) , (3, 128) , (-30, 129) , (10, 130) , (-59, 131) , (40, 132) , (-21, 133) , (78, 134) , (-65, 135) , (39, 136) , (-2, 137) , (34, 138) , (-41, 139) , (11, 140) , (-10, 141) , (37, 142) , (-85, 143) , (5, 144) , (-10, 145) , (91, 146) , (-84, 147) , (35, 148) , (-76, 149) , (72, 150) , (-58, 151) , (63, 152) , (37, 154) , (-10, 155) , (56, 156) , (-92, 157) , (-43, 158) , (31, 159) , (-58, 160) , (53, 161) , (72, 162) , (-37, 163) , (-3, 165) , (1, 166) , (-16, 167) , (82, 168) , (-20, 169) , (36, 170) , (-65, 171) , (2, 172) , (-77, 173) , (30, 174) , (-92, 175) , (17, 176) , (-15, 177) , (18, 178) , (-26, 179) , (96, 180) , (-55, 181) , (41, 182) , (-23, 183) , (37, 184) , (-94, 185) , (2, 186) , (-76, 187) , (59, 188) , (-27, 189) , (22, 190) , (-26, 191) , (55, 192) , (-58, 193) , (10, 194) , (-2, 195) , (90, 196) , (-26, 197) , (73, 198) , (-82, 199) , (95, 200) , (-95, 201) , (69, 202) , (-85, 203) , (58, 204) , (-6, 205) , (40, 206) , (-46, 207) , (22, 208) , (-82, 209) , (41, 210) , (-94, 211) , (52, 212) , (-72, 213) , (21, 214) , (-6, 215) , (87, 216) , (-10, 217) , (89, 218) , (-80, 219) , (92, 220) , (-70, 221) , (30, 222) , (-10, 223) , (38, 224) , (-40, 225) , (13, 226) , (-54, 227) , (17, 228) , (-55, 229) , (25, 230) , (-99, 231) , (18, 232) , (-13, 233) , (95, 234) , (-91, 235) , (46, 236) , (-79, 237) , (62, 238) , (-5, 239) , (81, 240) , (-38, 241) , (83, 242) , (-41, 243) , (-9, 245) , (97, 246) , (-67, 247) , (87, 248) , (-52, 249) , (39, 250) , (-25, 251) , (63, 252) , (-19, 253) , (93, 254) , (-32, 255) , (80, 256) , (-92, 257) , (26, 258) , (-12, 259) , (50, 260) , (-31, 261) , (6, 262) , (-37, 263) , (74, 264) , (-25, 265) , (21, 266) , (-80, 267) , (43, 268) , (-66, 269) , (53, 270) , (-43, 271) , (77, 272) , (-14, 273) , (76, 274) , (-25, 275) , (49, 276) , (-78, 277) , (83, 278) , (-58, 279) , (68, 280) , (-81, 281) , (78, 282) , (-58, 283) , (56, 284) , (-37, 285) , (99, 286) , (-62, 287) , (98, 288) , (-75, 289) , (1, 290) , (-5, 291) , (71, 292) , (-79, 293) , (7, 294) , (-65, 295) , (9, 296) , (-61, 297) , (-88, 298) , (33, 299) , (-36, 300) , (99, 301) , (-22, 302) , (3, 303) , (-54, 304) , (87, 305) , (69, 306) , (-18, 307) , (29, 308) , (-22, 309) , (33, 310) , (-57, 311) , (53, 312) , (-69, 313) , (8, 314) , (-1, 315) , (91, 316) , (-44, 317) , (45, 318) , (-24, 319) , (8, 320) , (-17, 321) , (35, 322) , (-85, 323) , (91, 324) , (-58, 325) , (70, 326) , (-1, 327) , (14, 328) , (-95, 329) , (16, 330) , (-20, 331) , (83, 332) , (-56, 333) , (34, 334) , (-88, 335) , (15, 336) , (-20, 337) , (16, 338) , (-80, 339) , (86, 340) , (-89, 341) , (35, 342) , (-86, 343) , (29, 344) , (-6, 345) , (7, 346) , (-6, 347) , (19, 348) , (-28, 349) , (34, 350) , (-25, 351) , (99, 352) , (-36, 353) , (27, 354) , (-81, 355) , (54, 356) , (-72, 357) , (75, 358) , (-87, 359) , (71, 360) , (-61, 361) , (15, 362) , (-47, 363) , (94, 364) , (-36, 365) , (87, 366) , (-59, 367) , (3, 368) , (-75, 369) , (-68, 370) , (13, 371) , (-9, 372) , (23, 373) , (-62, 374) , (58, 375) , (-4, 376) , (84, 377) , (-86, 378) , (27, 379) , (64, 380) , (-12, 381) , (24, 382) , (-63, 383) , (20, 384) , (-32, 385) , (30, 386) , (-81, 387) , (86, 388) , (-10, 389) , (23, 390) , (-13, 391) , (80, 392) , (-40, 393) , (50, 394) , (-88, 395) , (47, 396) , (-44, 397) , (86, 398) , (-79, 399) , (72, 400) , (-62, 401) , (85, 402) , (-12, 403) , (7, 404) , (-79, 405) , (20, 406) , (-36, 407) , (2, 408) , (-41, 409) , (85, 410) , (-71, 411) , (13, 412) , (-83, 413) , (22, 414) , (-90, 415) , (32, 416) , (-82, 417) , (-27, 419) , (90, 420) , (-82, 421) , (82, 422) , (40, 423) , (86, 425) , (-83, 426) , (71, 427) , (-11, 428) , (94, 429) , (-39, 430) , (36, 431) , (-98, 432) , (98, 433) , (-21, 434) , (62, 435) , (-50, 436) , (49, 437) , (-74, 438) , (66, 439) , (-44, 440) , (19, 441) , (-40, 442) , (89, 443) , (-99, 444) , (-66, 446) , (5, 447) , (-41, 448) , (77, 449) , (-79, 450) , (23, 451) , (-78, 452) , (91, 453) , (-6, 454) , (90, 455) , (-41, 456) , (3, 457) , (-36, 458) , (47, 459) , (-96, 460) , (32, 461) , (67, 462) , (-93, 463) , (15, 464) , (49, 466) , (-19, 467) , (84, 468) , (-27, 469) , (93, 470) , (-39, 471) , (65, 472) , (-3, 473) , (89, 474) , (-44, 475) , (87, 476) , (-17, 477) , (5, 478) , (-77, 479) , (39, 480) , (-25, 481) , (39, 482) , (-23, 483) , (24, 484) , (-14, 485) , (46, 486) , (-72, 487) , (43, 488) , (-85, 489) , (28, 490) , (-4, 491) , (47, 492) , (-67, 493) , (42, 494) , (-89, 495) , (41, 496) , (-23, 497) , (50, 498) , (-13, 499) , (23, 500) , (-46, 501) , (97, 502) , (-45, 503) , (21, 504) , (-13, 505) , (96, 506) , (-87, 507) , (56, 508) , (-44, 509) , (98, 510) , (-66, 511) , (28, 512) , (-7, 513) , (65, 514) , (-87, 515) , (24, 516) , (-10, 517) , (-7, 518) , (21, 519) , (-94, 520) , (20, 521) , (35, 522) , (-91, 523) , (48, 524) , (-48, 525) , (53, 526) , (88, 527) , (-79, 528) , (29, 529) , (-60, 530) , (15, 531) , (-50, 532) , (25, 533) , (-43, 534) , (57, 535) , (-54, 536) , (39, 537) , (-76, 538) , (44, 539) , (-76, 540) , (41, 541) , (-92, 542) , (91, 543) , (-63, 544) , (11, 545) , (-27, 546) , (26, 547) , (-32, 548) , (95, 549) , (-95, 550) , (47, 551) , (-3, 552) , (18, 553) , (-37, 554) , (45, 555) , (-37, 556) , (21, 557) , (-76, 558) , (73, 559) , (-48, 560) , (95, 561) , (-44, 562) , (44, 563) , (-99, 564) , (4, 565) , (60, 566) , (-42, 567) , (86, 568) , (-87, 569) , (13, 570) , (-94, 571) , (19, 572) , (-12, 573) , (40, 574) , (-91, 575) , (37, 576) , (-80, 577) , (56, 578) , (-44, 579) , (12, 580) , (-87, 581) , (-44, 582) , (60, 583) , (-54, 584) , (80, 585) , (-98, 586) , (16, 587) , (-14, 588) , (84, 589) , (-22, 590) , (61, 591) , (-82, 592) , (84, 593) , (-88, 594) , (97, 595) , (-1, 596) , (77, 597) , (-99, 598) , (67, 599) , (-33, 600) , (34, 601) , (-96, 602) , (67, 603) , (-68, 604) , (-74, 606) , (49, 607) , (-66, 608) , (49, 609) , (-57, 610) , (-83, 611) , (9, 612) , (-78, 613) , (66, 614) , (-11, 615) , (89, 616) , (-83, 617) , (59, 618) , (-82, 619) , (40, 620) , (-19, 621) , (58, 622) , (-93, 623) , (8, 624) , (-72, 625) , (28, 626) , (-18, 627) , (27, 628) , (-35, 629) , (59, 630) , (-1, 631) , (77, 632) , (-74, 633) , (66, 634) , (-94, 635) , (59, 636) , (-36, 637) , (23, 638) , (-32, 639) , (72, 640) , (-68, 641) , (59, 642) , (-46, 643) , (13, 644) , (-10, 645) , (26, 646) , (-52, 647) , (69, 648) , (-21, 649) , (-10, 650) , (91, 651) , (-84, 652) , (7, 653) , (33, 654) , (-31, 655) , (29, 656) , (-26, 657) , (40, 658) , (-86, 659) , (27, 660) , (-58, 661) , (-69, 662) , (84, 663) , (-96, 664) , (18, 665) , (-35, 666) , (3, 667) , (-46, 668) , (26, 669) , (-58, 670) , (29, 671) , (-84, 672) , (16, 673) , (-25, 674) , (21, 675) , (-65, 676) , (79, 677) , (-42, 678) , (2, 679) , (-74, 680) , (18, 681) , (-39, 682) , (91, 683) , (-14, 684) , (68, 685) , (-97, 686) , (33, 687) , (-42, 688) , (47, 689) , (-90, 690) , (73, 691) , (-39, 692) , (82, 693) , (-69, 694) , (16, 695) , (-76, 696) , (44, 697) , (-98, 698) , (-97, 700) , (58, 701) , (55, 702) , (-71, 703) , (31, 704) , (-72, 705) , (67, 706) , (-26, 707) , (82, 708) , (-35, 709) , (96, 710) , (-58, 711) , (27, 712) , (-1, 713) , (46, 714) , (-83, 715) , (37, 716) , (-13, 717) , (22, 718) , (-99, 719) , (83, 720) , (-58, 721) , (42, 722) , (-56, 723) , (55, 724) , (-81, 725) , (6, 726) , (-20, 727) , (99, 728) , (-11, 729) , (23, 730) , (-38, 731) , (75, 732) , (-88, 733) , (4, 734) , (-46, 735) , (38, 736) , (-10, 737) , (12, 738) , (-1, 739) , (78, 740) , (-91, 741) , (90, 742) , (-84, 743) , (80, 744) , (-79, 745) , (77, 746) , (-84, 747) , (17, 748) , (-33, 749) , (44, 750) , (-68, 751) , (27, 752) , (-82, 753) , (46, 754) , (-4, 755) , (94, 756) , (98, 757) , (-12, 758) , (81, 759) , (-63, 760) , (16, 761) , (-40, 762) , (82, 763) , (-80, 764) , (40, 765) , (-76, 766) , (9, 767) , (-66, 768) , (13, 769) , (-9, 770) , (63, 771) , (-25, 772) , (70, 773) , (-26, 774) , (67, 775) , (-86, 776) , (68, 777) , (-71, 778) , (87, 779) , (-24, 780) , (63, 781) , (-50, 782) , (89, 783) , (-40, 784) , (48, 785) , (-29, 786) , (90, 787) , (-17, 788) , (-76, 789) , (3, 790) , (-4, 791) , (88, 792) , (-79, 793) , (90, 794) , (-38, 795) , (11, 796) , (-54, 797) , (42, 798) , (-45, 799) , (87, 800) , (-77, 801) , (73, 802) , (-75, 803) , (46, 804) , (-55, 805) , (20, 806) , (-86, 807) , (82, 808) , (-83, 809) , (94, 810) , (-7, 811) , (2, 812) , (-12, 813) , (78, 814) , (-10, 815) , (11, 816) , (-85, 817) , (93, 818) , (-25, 819) , (82, 820) , (-4, 821) , (4, 822) , (-79, 823) , (71, 824) , (-89, 825) , (17, 826) , (-67, 827) , (48, 828) , (51, 829) , (-47, 830) , (89, 831) , (-42, 832) , (2, 833) , (52, 835) , (-38, 836) , (53, 837) , (-57, 838) , (4, 839) , (-51, 840) , (69, 841) , (-8, 842) , (62, 843) , (-23, 844) , (-87, 845) , (8, 846) , (-47, 847) , (62, 848) , (-63, 849) , (79, 850) , (-87, 851) , (51, 852) , (-93, 853) , (29, 854) , (-64, 855) , (98, 856) , (-61, 857) , (51, 858) , (-6, 859) , (14, 860) , (-98, 861) , (-33, 862) , (56, 863) , (-45, 864) , (34, 865) , (-28, 866) , (2, 867) , (-9, 868) , (32, 869) , (-47, 870) , (96, 872) , (-16, 873) , (80, 874) , (-52, 875) , (43, 876) , (-46, 877) , (39, 878) , (-99, 879) , (90, 880) , (-81, 881) , (9, 882) , (-66, 883) , (36, 884) , (-35, 885) , (8, 886) , (-84, 887) , (46, 888) , (-51, 889) , (25, 890) , (-54, 891) , (97, 892) , (-33, 893) , (90, 894) , (-82, 895) , (43, 896) , (-72, 897) , (78, 898) , (-16, 899) , (76, 900) , (-36, 901) , (62, 902) , (-63, 903) , (6, 904) , (-5, 905) , (1, 906) , (-71, 907) , (61, 908) , (75, 909) , (-11, 910) , (71, 911) , (-54, 912) , (41, 913) , (-2, 914) , (47, 915) , (-35, 916) , (84, 917) , (-85, 918) , (37, 919) , (55, 920) , (-89, 921) , (69, 922) , (11, 923) , (-14, 924) , (32, 925) , (-67, 926) , (22, 927) , (-92, 928) , (9, 929) , (-28, 930) , (26, 931) , (-17, 932) , (67, 933) , (-85, 934) , (23, 935) , (-58, 936) , (94, 937) , (90, 938) , (-81, 939) , (55, 940) , (-3, 941) , (65, 942) , (-51, 943) , (12, 944) , (-95, 945) , (88, 946) , (-8, 947) , (60, 948) , (-37, 949) , (90, 950) , (-56, 951) , (13, 952) , (-24, 953) , (93, 954) , (-70, 955) , (22, 956) , (10, 957) , (-52, 958) , (27, 959) , (-19, 960) , (83, 961) , (-61, 962) , (92, 963) , (-64, 964) , (16, 965) , (-76, 966) , (51, 967) , (-64, 968) , (65, 969) , (-99, 970) , (-82, 971) , (69, 972) , (-42, 973) , (64, 974) , (-16, 975) , (48, 976) , (-30, 977) , (78, 978) , (-21, 979) , (83, 980) , (-59, 981) , (74, 982) , (-98, 983) , (52, 984) , (-69, 985) , (69, 986) , (-39, 987) , (3, 988) , (8, 990) , (-98, 991) , (79, 992) , (-92, 993) , (26, 994) , (-59, 995) , (32, 996) , (-39, 997) , (24, 998) , (-74, 999) , (85, 1000) , (-56, 1001) , (33, 1002) , (91, 1003) , (-84, 1004) , (68, 1005) , (-93, 1006) , (33, 1007) , (-11, 1008) , (86, 1009) , (-53, 1010) , (26, 1011) , (-80, 1012) , (32, 1013) , (-52, 1014) , (14, 1015) , (-13, 1016) , (76, 1017) , (96, 1018) , (-40, 1019) , (48, 1020) , (87, 1021) , (-84, 1022) , (17, 1023) , (-58, 1024) , (31, 1025) , (-43, 1026) , (14, 1027) , (80, 1028) , (-24, 1029) , (70, 1030) , (-97, 1031) , (12, 1032) , (-14, 1033) , (56, 1034) , (-69, 1035) , (13, 1036) , (-72, 1037) , (4, 1038) , (-56, 1039) , (85, 1040) , (-8, 1041) , (14, 1042) , (43, 1043) , (-92, 1044) , (74, 1045) , (-71, 1046) , (79, 1047) , (-92, 1048) , (10, 1049) , (-72, 1050) , (20, 1051) , (-80, 1052) , (94, 1053) , (-35, 1054) , (59, 1055) , (-79, 1056) , (-57, 1057) , (53, 1058) , (-66, 1059) , (96, 1060) , (-63, 1061) , (86, 1062) , (-40, 1063) , (14, 1064) , (-54, 1065) , (18, 1066) , (-27, 1067) , (60, 1068) , (-55, 1069) , (38, 1070) , (-62, 1071) , (-21, 1072) , (87, 1073) , (-74, 1074) , (82, 1075) , (-20, 1076) , (50, 1077) , (-18, 1078) , (59, 1079) , (-81, 1080) , (38, 1081) , (-15, 1082) , (44, 1083) , (-9, 1084) , (38, 1085) , (-93, 1086) , (13, 1087) , (-71, 1088) , (-79, 1089) , (89, 1090) , (-34, 1091) , (97, 1092) , (-27, 1093) , (80, 1094) , (-33, 1095) , (17, 1096) , (-39, 1097) , (77, 1098) , (-35, 1099) , (65, 1100) , (-72, 1101) , (35, 1102) , (-45, 1103) , (48, 1104) , (-7, 1105) , (35, 1106) , (-26, 1107) , (-25, 1108) , (42, 1109) , (-38, 1110) , (43, 1111) , (-89, 1112) , (10, 1113) , (-26, 1114) , (56, 1115) , (-25, 1116) , (31, 1117) , (-54, 1118) , (59, 1119) , (-61, 1120) , (-67, 1121) , (86, 1122) , (-69, 1123) , (52, 1124) , (-68, 1125) , (55, 1126) , (-27, 1127) , (56, 1128) , (-41, 1129) , (84, 1130) , (-11, 1131) , (29, 1132) , (-37, 1133) , (77, 1134) , (-76, 1135) , (2, 1136) , (-23, 1137) , (-15, 1139) , (48, 1140) , (-64, 1141) , (79, 1142) , (-75, 1143) , (-41, 1144) , (5, 1145) , (-19, 1146) , (92, 1147) , (-89, 1148) , (38, 1149) , (-27, 1150) , (49, 1151) , (-14, 1152) , (79, 1153) , (-57, 1154) , (-12, 1155) , (39, 1156) , (-61, 1157) , (57, 1158) , (-90, 1159) , (30, 1160) , (-4, 1161) , (65, 1162) , (-36, 1163) , (95, 1164) , (-20, 1165) , (28, 1166) , (-82, 1167) , (15, 1168) , (-40, 1169) , (-56, 1170) , (71, 1171) , (-51, 1172) , (63, 1173) , (-46, 1174) , (98, 1175) , (-26, 1176) , (9, 1177) , (-88, 1178) , (77, 1179) , (-48, 1180) , (-49, 1182) , (60, 1183) , (-50, 1184) , (80, 1185) , (-3, 1186) , (52, 1187) , (-68, 1188) , (-12, 1189) , (19, 1190) , (-73, 1191) , (78, 1192) , (-38, 1193) , (24, 1194) , (-12, 1195) , (92, 1196) , (-41, 1197) , (38, 1198) , (-93, 1199) , (-12, 1200) , (41, 1201) , (-50, 1202) , (25, 1203) , (-26, 1204) , (-9, 1206) , (46, 1207) , (-6, 1208) , (70, 1209) , (-4, 1210) , (-17, 1211) , (24, 1212) , (-87, 1213) , (91, 1214) , (-42, 1215) , (86, 1216) , (-20, 1217) , (95, 1218) , (-71, 1219) , (88, 1220) , (-34, 1221) , (17, 1222) , (-37, 1223) , (59, 1224) , (-40, 1225) , (20, 1226) , (-3, 1227) , (2, 1228) , (-39, 1229) , (89, 1230) , (-1, 1231) , (48, 1232) , (48, 1234) , (-60, 1235) , (70, 1236) , (-31, 1237) , (49, 1238) , (-92, 1239) , (43, 1240) , (-96, 1241) , (93, 1242) , (-95, 1243) , (3, 1244) , (-71, 1245) , (87, 1246) , (-61, 1247) , (13, 1248) , (-15, 1249) , (94, 1250) , (-71, 1251) , (5, 1252) , (-9, 1253) , (-62, 1255) , (55, 1256) , (-68, 1257) , (47, 1258) , (-29, 1259) , (11, 1260) , (-58, 1261) , (87, 1262) , (-26, 1263) , (94, 1264) , (-29, 1265) , (86, 1266) , (-61, 1267) , (1, 1268) , (-77, 1269) , (40, 1270) , (-57, 1271) , (3, 1272) , (-63, 1273) , (55, 1274) , (-27, 1275) , (38, 1276) , (-91, 1277) , (78, 1278) , (26, 1280) , (-13, 1281) , (24, 1282) , (-60, 1283) , (1, 1284) ]

Execution time: 8.750281862 Seconds (execution time won’t be displayed in final project source code. it was taken in test phase.).

# Conclusion:

Thus the arithmetic operations for large integers has been implemented efficiently. We can also see the difference in performance of operations when the bases are increased. This is due to the fact that more the base, fewer the terms when the polynomial is created.