

ASSIGNMENT 1

CS674: POST QUANTUM SECURITY

Name:- VIKRANT CHAUHAN

Roll No. :- 231110407

Batch:- Y23

Programe:- MS(R) CYBERSECURITY

1.Question

Write a software code for performing negative wrapped convolution using Number Theoretic Transformation (NTT) for the parameters $n = 512$, $q = 12289$, $g = 10968$ for $Rq = \mathbb{Z}_q[X]/Xn + 1$. Your code must have three distinct functions:

- A function that will take the input in Rq and will convert the input into its corresponding NTT transformation using radix-2 Cooley-Tukey method.
- A function that will take the two inputs which are already transformed by NTT and will produce their point wise multiplication
- A function that will take the inputs in NTT transformed domain and will perform inverse NTT using the radix-2 Gentleman-Sande method.

2.Important Formulas

->We have to write code for negative wrapped convolution using Number Theoretic Transformation (NTT).The important formulas are:-

$$\tilde{A}_j = \sum_{i=0}^{n-1} \gamma^i \cdot \omega_n^{ij} \cdot A_i$$

->The above formula is used to calculate Negative Wrapped NTT.

->For PointWise Multiplication we just have to do component-wise multiplication this means we Suppose have $a=[a_0,a_1,a_2,\dots]$ and $b=[b_0,b_1,b_2,\dots]$.Then the pointwise multiplication of a and b is $c=[a_0*b_0,a_1*b_1,a_2*b_2,\dots]$

$$A_j = \frac{1}{n} \cdot \gamma^{-j} \cdot \sum_{i=0}^{n-1} \omega_n^{-ij} \cdot \tilde{A}_j$$

->The above formula is used to calculate Negative Wrapped Inverse NTT.

->Actually if we directly calculate the NTT and Inverse NTT with the help of the above formula then the Time Complexity would be $O(n^2)$.But we have better algorithms using Cooley-Tukey method and Gentleman-Sande Method with the help of which we can calculate the results in $O(n \log(n))$ Time.

->Some Important Variables and Formulas used:-

γ is $2n^{th}$ root of unity

$$\gamma = \sqrt{\omega}.$$

$$\gamma^{2n} \equiv 1 \pmod{q}$$

ω is the n^{th} root of unity

3.Algorithm Of My Code

->We firstly take 2 inputs of size 512 namely P_N1 and P_N2 .I have made them randomly generated inputs of size 512.These inputs cannot have a value greater than q .

```
P_N1=np.random.randint(0, q, n)
```

```
P_N2=np.random.randint(0, q, n)
```

->We firstly need parameters for my calculation.The parameters are:-

```
n=512
```

```
q=12289
```

```
gamma=10968
```

```
omega_HH=(gamma**2)%q
```

->Now before doing NTT operation I have created a gamma array which will be required during the NTT calculation

```
make_gamma_array()
```

```
def make_gamma_array():
```

```
    for i in range(0,n):
```

```
        temp=pow(gamma,i,q)
```

```
        gamma_array[i]=temp
```

->Now actually we find out the NTT of PN_1 and P_N2

```
P_N1_Modified1=NTT(t1)
```

```
P_N2_Modified1=NTT(t2)
```

->Now I do negative wrapped NTT calculation using the below code.The Below is NTT transformation using radix-2 Cooley-Tukey method The Time complexity of the below code is $O(N \log n)$:-

```
def NTT(P_NN):

    P_temp_Modified=P_NN

    t = n

    m=1

    gamma_original_ntt=Bitreverse(gamma_array,math.log2(n))

    while(m<n):

        t = int(t/2)

        for i in range(0,m):

            j1 =2*i*t

            j2 = j1 + t - 1

            S = gamma_original_ntt[m + i]

            for j in range(j1,j2+1):

                U = P_temp_Modified[j]

                V =P_temp_Modified[j+t]*S

                P_temp_Modified[j]=(U+V)%q

                P_temp_Modified[j+t]=(U-V)%q

    print(P_temp_Modified)
```

```
m=m*2
```

```
return P_temp_Modified
```

->Now after finding the NTT we actually have to pass the two obtained NTT outputs to PointWise Multiplication Function

```
Product_Result=PointWise_Multiplication(P_N1_Modified1,P_N2_Modified1)
```

->Now I calculate the PointWise Multiplication with the help of the below code

```
def PointWise_Multiplication(NTT_Poly1,NTT_Poly2):
```

```
    result=[]
```

```
    for i in range(0,len(NTT_Poly1)):
```

```
        temp=(NTT_Poly1[i]*NTT_Poly2[i])%q
```

```
        result.append(temp)
```

```
    return result
```

->Now we have to calculate the inverse NTT but before doing that we form the gamma inverse array

```
make_gamma_inverse_array()
```

```
def make_gamma_inverse_array():
```

```
    gamma_inv=gamma_inverse_calculate(gamma)
```

```
    for i in range(0,n):
```

```
        temp=pow(gamma_inv,i,q)
```

```
        t=reverse_num_binary(i)
```

```
        gamma_inverse_reverse[t]=temp
```

->Now actually we find out the negative wrapped INTT

```
P_Final=Inverse_NTT(Product_Result)
```

->Now I do negative wrapped Inverse NTT calculation using the below code.The Below is NTT transformation using radix-2 Gentleman-Sande method The Time complexity of the below code is $O(N \log n)$:-

```
def Inverse_NTT(P_Modified):  
  
    P_temp_Modified2=P_Modified  
  
    t = 1  
  
    m=n  
  
    while(m>1):  
  
        j1 = 0  
  
        h = int(m/2)  
  
        for i in range(0,h):  
  
            j2 = j1 + t- 1  
  
            G = gamma_inverse_reverse[h + i]  
  
            j=j1  
  
            while(j<=j2):  
  
                u1 = P_temp_Modified2[j]  
  
                t1 =P_temp_Modified2[j+t]  
  
                P_temp_Modified2[j]=(u1+t1)%q
```

```

P_temp_Modified2[j + t] = ((u1-t1)*G)%q

j+=1

j1 = j1 + 2*t

t = 2*t

m=int(m/2)

for i in range(0,n):

    P_temp_Modified2[i]=(P_temp_Modified2[i]*n_inverse(n))%q

return P_temp_Modified2

```

->The Result Of the negative wrapped Inverse NTT is our final answer.

```

P_Final=Inverse_NTT(Product_Result)

print("After INTT ",P_Final)

```

4.Verification

->To check whether our output is coming correct or not we know that the result which we obtained after negative wrapped Inverse NTT should be equal to multiplication of polynomials P_{N1} and P_{N2} and then take mod x^N+1 . If these two results are equal then our answer is correct.

->For verification I have written the Below code which is used to find the multiplication of P_{N1} & P_{N2} and then take mod x^N+1 .

```

print("Verification :)")

f = np.zeros(n + 1)

```

```
f[0] = 1

f[n] = 1

ans=np.reminder(np.polydiv(np.polymul(P_N1[::-1],
P_N2[::-1]),f[1],q).astype(int))[:-1]

print(ans)
```

->Now we check equate ans and P_Final.If the result comes out as True then our result obtained is correct.In my code i am getting TRUE for all the random values of P_N1 and P_N2.

5.Output ScreenShots


```

vikrantchauhan@Vikrants-MacBook-Air Assignment % python3 Vikrant_Neg_Wrap.py
PN1 is [ 3815 3133 9194 2774 4112 2277 9465 11354 5130 5607 10764 5908
352 11134 5576 8973 1913 11462 9623 9348 11770 3679 6207 2478
630 8255 302 4825 3917 4495 11998 9296 10661 2214 4506 9050
5891 12248 1515 3376 9561 11590 4595 12058 10264 9901 1036 2186
4517 5570 1084 6989 4166 3487 2822 1245 1738 11170 9329 5661
1082 3243 2183 5463 5874 10902 6871 1299 6345 2728 10053 1350
1792 9758 1499 12276 8396 5442 11097 5096 3534 128 10117 1719
180 10845 7038 9371 6856 8175 2666 866 7757 1618 12136 6995
6539 7486 6779 615 522 5215 4199 5806 9129 6114 8494 8431
3844 5689 6034 12041 4433 2613 9251 2387 6940 2201 9917 673
6063 3170 4857 10972 11438 5167 5051 668 5449 628 7992 1591
6800 5196 2589 3277 10831 397 5373 3747 11965 10786 1443 7752
9560 6588 8023 6308 9344 7668 11268 5877 7973 413 11381 5635
3740 4201 1243 809 1754 5807 6750 11358 11478 7281 10394 10216
5032 7975 8832 8050 6624 314 8931 11008 7778 4428 9292 914
11443 3777 7853 8101 11900 1395 12244 11785 1624 293 10041 194
3964 2490 7595 3133 2908 6503 9106 5702 2675 9599 2078 9494
2523 7736 8176 12098 11431 11297 7517 5380 11369 1758 11464 1531
6305 9341 5571 3117 7863 416 3278 9507 1729 9471 6914 3306
2310 12148 8652 5227 1998 1276 799 10548 11964 733 2183 4005
12107 9971 11290 7266 4822 5771 10782 8938 744 4042 8617 12171
8730 1339 6353 343 4217 4560 3940 9900 3829 9241 6313 3564
6033 1387 2986 12282 10796 6341 9585 7546 6936 5341 3148 11766
11905 12098 2105 4946 1101 2340 9143 1793 11286 5583 6591 7636
6532 5990 10934 7535 10205 4224 8010 8328 2953 8435 7236 1777
511 3937 5544 8600 5732 11702 567 11363 8161 8270 7017 10673
2876 683 10313 332 6098 7898 7248 5064 966 12198 9283 7641
9479 4955 6681 5601 4929 422 11960 12261 7526 3827 3946 7905
5713 10528 5706 7022 3798 9287 9625 2184 798 1611 11895 11319
11383 1322 10582 2296 4539 4185 2193 10672 3677 3588 9393 7632
2346 398 2181 7187 6837 789 4320 1682 11503 5473 5266 7669
6634 9876 6924 4368 9812 2959 8837 3983 10064 11719 12009 4917
8589 1336 6905 2746 10832 9847 10316 6300 10572 5353 7501 4986
5004 10776 3057 1994 10896 7182 552 4685 8670 11099 5247 11468
9039 6644 8669 7963 3539 8359 11456 1051 2685 11827 5830 1490
11304 7865 2349 266 12130 3528 8434 4777 1343 6998 10137 10294
12089 8875 732 3989 1651 2563 9193 4037 10764 8518 9536 10212
2056 2402 7226 10144 558 5944 8616 2437 9476 6865 7130 8426
8704 1979 6520 4226 10430 5120 12242 2192 266 9964 6021 105
2124 5619 172 771 12062 93 177 2731 309 770 783 5027
2523 11341 8016 4186 11275 2596 11477 8611 2481 3398 4126 2241
2807 1311 9750 2718 9499 7354 3699 10622 10603 1318 432 4848
3782 7238 1593 10982 4634 7744 4536 4236]
PN2 is [ 1705 1009 2375 3749 11453 4744 5948 189 12287 386 10104 10078
5642 8814 4441 2999 11311 3069 5642 2567 2342 11852 2578 6837
3804 2075 5311 397 8587 4935 1378 3733 9180 8439 9058 7165
8812 10650 11472 6251 4064 4479 3308 3454 5234 1949 6451 10031
1338 819 1974 2071 7999 9308 2219 2322 8948 1996 4202 6236
8441 5233 5782 3975 3528 11913 9382 2842 1253 480 10681 1025
5950 9517 10910 7651 5384 7504 11882 10038 1893 776 1411 6004
6354 5110 6122 7501 7954 6311 3988 1533 4108 7831 3862 1034
5804 578 6085 6409 8106 2087 3026 10624 141 4340 2460 7784
11423 6239 3220 12136 8033 4432 11190 1994 2093 414 368 3612
11116 1159 6133 2272 9712 533 10830 6638 8269 9636 6046 8355

```

Figure : Inputs PN1 And PN2

```

Gamma array [1, 10968, 3, 8326, 9, 400, 27, 1200, 81, 3600, 243, 10800, 729, 7822, 2187, 11177, 6561, 8953,
1, 9928, 9764, 5206, 4714, 3329, 1853, 9987, 5559, 5383, 4388, 3860, 875, 11580, 2625, 10162, 7875, 5908, 11
468, 8646, 7404, 1360, 9923, 4080, 5191, 12240, 3284, 12142, 9852, 11848, 4978, 10966, 2645, 8320, 7935, 38
5, 3704, 10327, 11112, 6403, 8758, 6920, 1696, 8471, 5088, 835, 2975, 2505, 8925, 7515, 2197, 10256, 6591, 6
734, 7969, 4624, 11618, 1583, 10276, 4749, 6250, 1958, 6461, 5874, 7094, 5333, 8993, 3710, 2401, 11130, 7203
8076, 10745, 11939, 7657, 11239, 10682, 9139, 7468, 2839, 10115, 8517, 5767, 973, 5012, 2919, 2747, 8757, 8
10806, 5092, 7840, 2987, 11231, 8961, 9115, 2305, 2767, 6915, 8301, 8456, 325, 790, 975, 2370, 2925, 7110, 8
099, 11287, 8719, 9283, 1579, 3271, 4737, 9813, 1922, 4861, 5766, 2294, 5009, 6882, 2738, 8357, 8214, 493, 6
8, 6234, 10805, 6413, 7837, 6950, 11222, 8561, 9088, 1105, 2686, 3315, 8058, 9945, 11885, 5257, 11077, 3482,
42, 453, 3748, 1359, 11244, 4077, 9154, 12231, 2884, 12115, 8652, 11767, 1378, 10723, 4134, 7591, 113, 10484
59, 11973, 11899, 11341, 11119, 9445, 8779, 3757, 1759, 11271, 5277, 9235, 3542, 3127, 10626, 9381, 7300, 35
2912, 11994, 8736, 11404, 1630, 9634, 4890, 4324, 2381, 683, 7143, 2049, 9140, 6147, 2842, 6152, 8526, 6167,
10963, 6608, 8311, 7535, 355, 10316, 1065, 6370, 3195, 6821, 9585, 8174, 4177, 12233, 242, 12121, 726, 11788
0040, 9280, 5542, 3262, 4337, 9786, 722, 4780, 2166, 2051, 6498, 6153, 7205, 6170, 9326, 6221, 3400, 6374, 1
, 1207, 3123, 3621, 9369, 10863, 3529, 8011, 10587, 11744, 7183, 10654, 9260, 7384, 3202, 9863, 9606, 5011,

```

Figure : Gamma Array

```

Result Of NTT PN1 [221, 6881, 9044, 11452, 11531, 6756, 10293, 4700, 6798, 8568, 4873, 9257, 3964, 3144, 122, 5442, 600
16, 8905, 1265, 10200, 8126, 11944, 285, 4107, 7963, 11405, 3193, 5478, 9822, 1595, 6891, 9845, 8323, 4166, 6956, 6343,
093, 2106, 1284, 6862, 7969, 889, 6184, 1035, 5028, 2324, 1888, 5778, 11557, 7428, 5671, 8478, 6589, 8838, 1265, 12144,
7307, 5368, 6659, 10731, 10055, 11707, 8997, 3208, 2736, 8837, 3772, 2985, 6564, 7200, 4350, 682, 7927, 9850, 4012, 773
11284, 9792, 1730, 11955, 6041, 1562, 8712, 7935, 2019, 1494, 5032, 9563, 298, 3531, 11922, 9211, 537, 10467, 3324, 114
, 4252, 4878, 4472, 28, 3883, 2230, 6786, 3607, 3106, 6082, 4781, 3843, 2165, 2627, 3223, 2433, 9272, 4065, 7516, 3637,
498, 8725, 4441, 5844, 11613, 7618, 11775, 4712, 3246, 2374, 7611, 6556, 11442, 6622, 2954, 2227, 748, 4717, 2643, 6233,
73, 5051, 968, 8621, 11237, 9900, 8353, 181, 1918, 362, 11178, 6173, 3300, 2042, 7277, 6751, 12257, 5309, 7854, 3112, 59
2188, 7782, 7077, 9048, 8860, 6310, 12101, 4085, 1110, 4022, 8836, 3724, 11946, 4072, 1598, 5408, 4566, 4049, 8805, 289
04, 10152, 8874, 10251, 9937, 2784, 1539, 12096, 2775, 7735, 4824, 10217, 9823, 6844, 6273, 3280, 11789, 1279, 3573, 427
1, 7814, 11200, 9353, 9363, 11009, 4152, 12276, 8831, 9381, 9773, 6926, 6164, 12196, 6627, 7761, 5628, 3477, 4405, 2907,
51, 9901, 6171, 11742, 10266, 9609, 1156, 1467, 11442, 7343, 10286, 8677, 11132, 3868, 12284, 4858, 10804, 2452, 6508, 1
, 10510, 2474, 10064, 8169, 9079, 1188, 2304, 8124, 6872, 11846, 882, 4677, 5432, 9967, 8681, 2275, 9406, 7069, 10019, 7
10, 5389, 7401, 604, 10649, 4879, 8952, 7258, 8161, 4444, 1814, 7960, 8079, 5678, 11232, 55, 2678, 10693, 652, 3211, 983
855, 2556, 1608, 1155, 8836, 10475, 12116, 3307, 4405, 11230, 8174, 6678, 6009, 3227, 5595, 9076, 5194, 2976, 10295, 257
1705, 1304, 10417, 4431, 8294, 7543]
Result Of NTT PN2 [8727, 5963, 5008, 11545, 534, 5222, 5873, 8782, 4448, 11419, 8508, 2074, 1617, 904, 2660, 3857, 7021
, 1492, 4313, 3992, 676, 1810, 2802, 10625, 2710, 275, 4825, 1114, 895, 7954, 10513, 4740, 5025, 11522, 10202, 4927, 111
11923, 1112, 1146, 11339, 3395, 5650, 10764, 11104, 3528, 10293, 8359, 11186, 7832, 10973, 3862, 5835, 982, 5099, 6428,
149, 5309, 3054, 4567, 10348, 9310, 10752, 7399, 8283, 5959, 2171, 1641, 5925, 3611, 2370, 7058, 11778, 8720, 8012, 1331
87, 9871, 2804, 11326, 10513, 7893, 2052, 11105, 8900, 11331, 9708, 2599, 3622, 8466, 11399, 12129, 8689, 7166, 11895, 1
372, 2112, 8319, 5163, 5707, 6031, 6991, 12260, 1344, 4970, 4605, 4161, 10033, 3320, 1201, 4752, 7499, 5827, 2942, 1982,
507, 3296, 5957, 2365, 346, 10436, 10790, 11864, 8019, 10948, 2087, 6687, 5397, 9044, 10741, 6167, 11534, 3239, 8884, 10
2, 1421, 10246, 9804, 5198, 4306, 362, 8976, 7096, 11777, 6685, 11753, 7391, 11467, 1281, 11307, 11894, 8844, 4119, 9140
6105, 2550, 9867, 3782, 5107, 1568, 7517, 7298, 4988, 10429, 1659, 9221, 6884, 5253, 7251, 1176, 10404, 5209, 786, 7766
38, 5682, 2627, 11921, 4173, 9430, 5631, 2463, 11359, 961, 11539, 8662, 7647, 8534, 215, 3932, 10892, 7971, 11555, 4907,
84, 1172, 8336, 1614, 11247, 674, 10545, 9145, 4908, 5165, 11135, 12219, 4858, 9229, 968, 2546, 12078, 5969, 5136, 6350,
6118888, 5858, 8418, 8744, 7388, 8818, 5888, 8815, 8188, 5188, 11877, 8815, 1581, 1178, 1314, 11811, 8855, 11855, 12

```

Figure : Result Of NTT of PN1 and PN2

```

Result Of Point wise Multiplication [11583, 10721, 7387, 8278, 765, 10402, 1198, 8938, 6564, 5263, 8687,
7921, 3118, 546, 1851, 11918, 4943, 12282, 2289, 12074, 10925, 246, 2680, 8108, 7148, 817, 4382, 1428,
5816, 5838, 3823, 3411, 2284, 11181, 11763, 7350, 1973, 6906, 1985, 2309, 4275, 2532, 8611, 12259, 8676,
1189, 7520, 6217, 4538, 521, 10580, 12234, 10466, 1029, 9025, 5933, 1372, 1318, 4538, 7363, 9304, 7965,
9157, 404, 276, 10400, 3847, 9054, 2128, 11770, 2999, 8818, 6045, 2582, 6561, 1881, 5879, 10213, 6598,
59, 188, 3620, 9254, 1804, 10194, 39, 7828, 7478, 12119, 5942, 1836, 979, 10139, 6226, 11024, 9043, 3602
4411, 2057, 2326, 4644, 8159, 6628, 11439, 9388, 9537, 9142, 9709, 2071, 6008, 2701, 8068, 10459, 5020,
5615, 12038, 7592, 7623, 5463, 9246, 135, 641, 3041, 5640, 4403, 2592, 9940, 1151, 12173, 10152, 1211, 5
8393, 4393, 6433, 4519, 48, 12058, 6490, 7851, 2151, 5588, 5776, 10862, 331, 12244, 3791, 2169, 4905, 43
9376, 2007, 11438, 12209, 4445, 1656, 792, 12019, 6543, 11151, 7880, 942, 9839, 1423, 10526, 9987, 10965
, 2387, 912, 3282, 4150, 11954, 3009, 5817, 2854, 6589, 8896, 6936, 4827, 4965, 6587, 9002, 2649, 8168,
6, 11098, 7495, 7450, 11097, 5425, 8350, 12232, 10786, 1634, 2374, 3194, 2955, 8930, 5036, 7073, 3559, 7
, 11502, 3337, 8651, 7244, 11664, 1481, 2547, 6095, 4727, 9645, 8875, 11451, 6581, 11919, 10969, 3567, 6
7, 10532, 8095, 9941, 1398, 3396, 2237, 187, 8792, 3160, 10850, 117, 27, 2615, 6636, 139, 5276, 11090, 9
95, 2329, 3654, 5600, 2468, 9642, 11876, 8791, 7768, 5181, 4287, 4161, 10494, 6209, 2740, 9712, 2709, 75
4, 1300, 5569, 4464, 10275, 6674, 8321]

```

Figure : Result Of PointWise Multiplication

```

Gamma inverse array [1, 10810, 5146, 8246, 11567, 10984, 8155, 6553, 8668, 9744, 8747, 3
5728, 11227, 9995, 3553, 4805, 4846, 9542, 3135, 8577, 11334, 11499, 1170, 2319, 1326, 5
731, 3932, 7399, 6378, 6747, 12144, 3637, 3459, 3694, 5179, 10530, 8582, 11934, 8907, 423
, 7222, 10092, 11462, 6522, 8541, 953, 9452, 5374, 130, 4354, 8340, 3296, 4452, 2396, 427
7952, 11854, 10911, 10377, 11082, 3248, 7012, 1168, 11224, 2143, 404, 4645, 7278, 1002,
77, 1045, 2859, 3778, 3833, 390, 773, 442, 9888, 1067, 7188, 545, 5019, 2678, 8585, 12047
9424, 9919, 3510, 6957, 3978, 2969, 9603, 3247, 4905, 8304, 11813, 3531, 10111, 1544, 118
8236, 9644, 2780, 5195, 1484, 4895, 1426, 4654, 1663, 10512, 2704, 6998, 3636, 4938, 1012
, 5919, 7856, 7032, 8455, 8049, 3570, 6224, 11454, 1319, 3150, 4046, 709, 4079, 1058, 922
28, 4049, 5106, 5961, 1594, 1962, 168, 9597, 4298, 8960, 6119, 6992, 3956, 10929, 6122, 2
7, 4493, 2057, 5369, 1512, 350, 1815, 6906, 5915, 1483, 11026, 49, 5942, 10706, 2500, 148
8, 218, 4115, 9259, 1843, 2361, 10238, 10335, 1805, 9407, 6142, 9842, 11713, 3963, 5315,
, 1050, 5445, 8429, 5456, 4449, 8500, 147, 5537, 7540, 7500, 4467, 8410, 10367, 8291, 203
083, 6136, 6427, 5415, 3643, 6137, 4948, 10561, 11889, 1956, 7280, 885, 6008, 1003, 3532,
11868, 8209, 6077, 7665, 9026, 8689, 11858, 10710, 6383, 9784, 3957, 9450, 12138, 2127, 1
10240, 7377, 12097, 1321]

```

Figure : Gamma Inverse Array


```
Result Of INTT Final Answer [11875, 11205, 8771, 10171, 5756, 9266, 519, 6984, 12081, 7732, 9
7, 6055, 2005, 11218, 10781, 2031, 4712, 7814, 7474, 5601, 1610, 1851, 4373, 9016, 7720, 5770,
10823, 3715, 1465, 5782, 9008, 93, 3955, 5679, 9474, 7879, 6423, 6468, 818, 1835, 10974, 3889,
668, 4824, 7633, 2458, 9700, 385, 9824, 2841, 9164, 10684, 9573, 6927, 8597, 1761, 8015, 12259
600, 6368, 5863, 2750, 2219, 6590, 7494, 865, 618, 8244, 4749, 12250, 2470, 8618, 9708, 3537,
4, 9852, 3709, 2472, 8259, 11874, 4909, 4747, 7804, 7752, 8398, 2288, 3392, 1614, 651, 4525, 5
99, 5957, 5423, 9699, 5902, 6455, 4780, 3193, 439, 9717, 4046, 10765, 1585, 15, 11116, 587, 60
7, 5331, 10169, 9317, 9697, 2343, 7285, 11221, 12231, 1821, 480, 8024, 11733, 4862, 8174, 5723
49, 6828, 4572, 5432, 8931, 9583, 2848, 2065, 1191, 2949, 11938, 3391, 4245, 7503, 5845, 4579,
, 8866, 7713, 5461, 9596, 2420, 6121, 6422, 11003, 2381, 10219, 2338, 11754, 11973, 5469, 3270
768, 6861, 5126, 5613, 9615, 4428, 10996, 3114, 1494, 10893, 7535, 1955, 8408, 10471, 2444, 23
5, 1629, 4350, 3983, 12011, 10712, 268, 8418, 7662, 6073, 2742, 4681, 637, 10624, 924, 4165, 1
17, 2332, 11819, 1638, 10682, 3050, 5430, 4228, 1566, 9658, 670, 6349, 9580, 6204, 4766, 10374
, 7465, 4212, 10222, 1151, 2412, 1050, 6694, 8920, 9623, 7151, 329, 6118, 4789, 11375, 1932, 5
2055, 1989, 8648, 3958, 1782, 1523, 9032, 1183, 11524, 5773, 8079, 4821, 5284, 2124, 49, 1068
4, 8889, 5358]
Verification :-)
```

Figure : Result Of INTT

VERIFICATION STARTS:-

```

Verification :)
PN1 is [ 1705 1009 2375 3749 11453 4744 5948 189 12287 386 10104 10078
5642 8814 4441 2999 11311 3069 5642 2567 2342 11852 2578 6837
3804 2075 5311 397 8587 4935 1378 3733 9180 8439 9058 7165
8812 10650 11472 6251 4064 4479 3308 3454 5234 1949 6451 10031
1338 819 1974 2071 7999 9308 2219 2322 8948 1996 4202 6236
8441 5233 5782 3975 3528 11913 9382 2842 1253 480 10681 1025
5950 9517 10910 7651 5384 7504 11882 10038 1893 776 1411 6004
6354 5110 6122 7501 7954 6311 3988 1533 4108 7831 3862 1034
5804 578 6085 6409 8106 2087 3026 10624 141 4340 2460 7784
11423 6239 3220 12136 8033 4432 11190 1994 2093 414 368 3612
11116 1159 6133 2272 9712 533 10830 6638 8269 9636 6046 8355
7890 9276 11419 11538 25 12192 8822 849 4004 2008 6256 6705
2156 11192 5634 6217 6624 4974 4735 7813 7517 9224 4328 12231
5012 2791 8270 1933 11096 8006 9115 2576 12043 1903 11691 4527
6371 1556 8712 858 3763 8693 7744 8977 1514 4990 5488 713
6274 10312 6623 3898 5564 2367 393 281 7614 1126 1057 10539
6731 817 11826 1563 6719 10246 6839 9352 9569 7974 1643 8
12248 8887 3229 215 11472 3848 10226 7161 6233 1530 9771 5176
8424 685 10393 9230 2956 9782 7471 1604 1153 9090 6617 8610
209 9573 9837 11863 6210 8635 1483 5685 5815 1278 4740 9014
1462 1768 10879 10191 7615 95 7317 10894 798 6629 8032 583
4576 12204 10499 3656 5819 6894 2702 10175 2929 8211 782 10345
8572 8900 1670 9541 4660 5601 7783 4185 2849 9904 10928 10246
9638 2345 29 12205 9340 8786 221 10556 5239 10516 11923 8333
11043 5299 1068 8796 7097 6903 8378 2970 8200 1054 9199 726
11324 9054 6330 8292 8254 9250 3806 5049 2404 1403 11296 10013
11841 6813 6744 9854 10514 2169 4150 9799 11981 8948 7900 5141
7258 10836 8907 7626 1117 6067 8287 1527 6575 6547 11076 2986
6678 5355 3309 667 6948 7736 7256 6778 2499 2466 3249 8775
10412 3072 11319 4458 9070 3152 9055 4942 3816 8505 4524 4781
6779 1018 5506 3136 10559 3356 6991 2765 8397 9656 13 5050
5115 388 15 11714 540 11197 2046 6962 6268 380 12114 6745
540 3533 7303 5261 302 4387 11669 8595 9628 7533 2337 5603
6046 5522 10271 11329 8801 419 10224 8278 6161 11199 5175 4888
11584 7387 2706 8164 11749 10407 9926 2724 4087 9372 5186 1629
5959 6620 1562 3995 1881 11580 4059 4822 2040 4328 10234 118
1617 2992 1310 94 813 2017 6677 10790 1605 7007 11707 3670
559 1844 10379 12211 10936 979 10952 5636 10092 6991 1653 8289
7108 6853 2238 4028 1644 6696 7143 10207 6022 10711 11105 11943
11932 11997 2162 329 7625 3341 1197 3380 10100 11075 5124 3980
7885 4771 10188 2336 7500 70 4269 11600 7213 5300 2732 4069
1838 5406 180 5424 8575 210 522 5303 9955 4576 7945 6128
2228 9759 2533 4387 10118 1526 11408 7858]
PN2 is [ 1705 1009 2375 3749 11453 4744 5948 189 12287 386 10104 10078
5642 8814 4441 2999 11311 3069 5642 2567 2342 11852 2578 6837
3804 2075 5311 397 8587 4935 1378 3733 9180 8439 9058 7165
8812 10650 11472 6251 4064 4479 3308 3454 5234 1949 6451 10031
1338 819 1974 2071 7999 9308 2219 2322 8948 1996 4202 6236
8441 5233 5782 3975 3528 11913 9382 2842 1253 480 10681 1025
5950 9517 10910 7651 5384 7504 11882 10038 1893 776 1411 6004
6354 5110 6122 7501 7954 6311 3988 1533 4108 7831 3862 1034
5804 578 6085 6409 8106 2087 3026 10624 141 4340 2460 7784
11423 6239 3220 12136 8033 4432 11190 1994 2093 414 368 3612
11116 1159 6133 2272 9712 533 10830 6638 8269 9636 6046 8355

```

Figure : Inputs PN1 And PN2

```

Polynomial Multiplication and modulo [11875 11205 8771 10171 5756 9266
12173 1429 3898 12143 11492 5640 5191 2058 4566 2105 2213 7033
1833 6520 11943 10730 6457 6055 2005 11218 10781 2031 4712 7814
7474 5601 1610 1851 4373 9016 7720 5770 12019 5374 8134 4992
1034 1011 2597 9164 119 11911 1231 1345 1004 1808 5617 11003
2 7072 2034 10823 3715 1465 5782 9008 93 3955 5679 9474
7879 6423 6468 818 1835 10974 3889 9379 654 6755 5382 6608
8031 4540 11085 7924 7460 5768 643 334 8576 7026 3921 6410
11323 408 3668 4824 7633 2458 9700 385 9824 2841 9164 10684
9573 6927 8597 1761 8015 12259 4144 7213 6402 4348 4912 7460
5979 7701 8256 937 1011 5055 10626 4941 3682 5033 7641 6977
603 600 6368 5863 2750 2219 6590 7494 865 618 8244 4749
12250 2470 8618 9708 3537 1897 6442 10737 11909 3891 8739 6398
12029 9802 6040 9800 2344 591 6572 3102 10608 863 2800 9794
9852 3709 2472 8259 11874 4909 4747 7804 7752 8398 2288 3392
1614 651 4525 5488 10729 4805 1374 7036 4534 11293 9411 1296
4829 11883 8448 11602 5945 4910 4522 4751 11405 4799 5957 5423
9699 5902 6455 4780 3193 439 9717 4046 10765 1585 15 11116
587 6060 207 9249 1654 8014 8206 3555 2591 3811 9333 1326
10781 9578 156 8635 1904 7466 8955 8417 1697 5331 10169 9317
9697 2343 7285 11221 12231 1821 480 8024 11733 4862 8174 5723
915 11491 4026 11979 5644 2783 9559 3927 2643 728 5325 1659
2486 2966 10388 6884 10059 8895 1549 6828 4572 5432 8931 9583
2848 2065 1191 2949 11938 3391 4245 7503 5845 4579 2146 12203
4804 5548 5823 9957 2438 516 11063 2533 9455 9860 7169 8214
9632 7132 9816 1593 7608 8866 7713 5461 9596 2420 6121 6422
11003 2381 10219 2338 11754 11973 5469 3270 8855 8063 8334 2192
4633 9057 9334 8345 6996 7852 4322 10849 2806 7197 12137 10205
5822 1275 6768 6861 5126 5613 9615 4428 10996 3114 1494 10893
7535 1955 8408 10471 2444 2382 10610 2079 1542 8668 1676 7835
4318 9119 1640 5235 11747 11242 6219 10734 6574 7317 5908 11985
1629 4350 3983 12011 10712 268 8418 7662 6073 2742 4681 637
10624 924 4165 11208 9300 6492 7887 1469 3906 11608 33 9436
4196 7103 8707 10747 3699 792 5613 3206 1479 221 7417 2332
11819 1638 10682 3050 5430 4228 1566 9658 670 6349 9580 6204
4766 10374 10450 6081 4884 8492 10463 786 11226 11247 1341 12133
9607 12118 7559 7101 7997 10530 9177 2564 7465 4212 10222 1151
2412 1050 6694 8920 9623 7151 329 6118 4789 11375 1932 5689
462 5383 10044 10208 6114 8292 2167 5830 51 5928 6423 6095
66 8073 563 9147 1519 9195 4934 2055 1989 8648 3958 1782
1523 9032 1183 11524 5773 8079 4821 5284 2124 49 10681 105
2839 10693 8659 767 7738 11774 5730 6203 8514 2633 6348 5368
6415 1575 10967 412 1992 2774 8889 5358]

```

Figure : Output Of Polynomial Multiplication And Modulo X^{n+1}

```

462 5383 10044 10208 6114 8292 2167 5830 51 5928 6423 6095
66 8073 563 9147 1519 9195 4934 2055 1989 8648 3958 1782
1523 9032 1183 11524 5773 8079 4821 5284 2124 49 10681 105
2839 10693 8659 767 7738 11774 5730 6203 8514 2633 6348 5368
6415 1575 10967 412 1992 2774 8889 5358]
True
Your Answer Is Correct
vikrantchauhan@Vikrants-MacBook-Air Assignment %

```

Figure : Final Verification Output

->We can see that the final output is coming as TRUE and Your Answer is Correct is Printed
Therefore my answer is correct.

->Hence Our Result is successfully Verified.Hence Proved

6.References

1. CS674 Class Notes
2. Algorithm Given In Assignment PDF
3. Speeding up the Number Theoretic Transform for Faster Ideal Lattice-Based Cryptography Patrick Longa and Michael Naehrig, Microsoft Research, USA
4. Compact Ring-LWE Cryptoprocessor Sujoy Sinha Roy¹, Frederik Vercauteren¹, Nele Mentens¹, Donald Donglong Chen² and Ingrid Verbauwhede¹
5. High-Performance Ideal Lattice-Based Cryptography on 8-bit ATxmega Microcontrollers Extended Version Thomas Pöppelmann, Tobias Oder, and Tim Güneysu

Name:- VIKRANT CHAUHAN

Roll No. :- 231110407

Batch:- Y23

Programe:- MS(R) CYBERSECURITY