Challenge Name:

Amicable Numbers

Problem Statement:

From Wikipedia: “**Amicable numbers** are two different numbers so related that the sum of the proper divisors of each is equal to the other number. (A proper divisor of a number is a positive factor of that number other than the number itself. For example, the proper divisors of 6 are 1, 2, and 3.) A pair of amicable numbers constitutes an aliquot sequence of period 2.

The smallest pair of amicable numbers is (220, 284). They are amicable because the proper divisors of 220 are 1, 2, 4, 5, 10, 11, 20, 22, 44, 55 and 110, of which the sum is 284; and the proper divisors of 284 are 1, 2, 4, 71 and 142, of which the sum is 220.”

In this challenge, you have to print the amicable numbers in an n*m grid. For example, if n = 2, m = 2, the output should be :

```
220   284
1184  1210
```

Hint: It is enough to generate amicable numbers upto **10^7** for any of the test cases.

Input Format :

The first line of the input will contain an integer **T (T<10)**, number of test cases. Each case will contain two integers **n** and **m (1 <= n <= 10, 1 <= m <= 20).**

Constraints:

1 <= T <= 10
1 <= n <= 10
1 <= m <= 20

Output Format:

For each case, print the case number and then n*m amicable numbers in an n*m grid. See sample for more clarification.

Sample Input 0:

```
2
1 1
2 2
```

Sample Output 0:

```
Case 1:
220
Case 2:
220 284
1184 1210
```

Sample Input 1:

```
1
2 3
```

Sample Output 1:

```
Case 1:
220 284 1184
1210 2620 2924
```

Test Case 1:

Input :

```
2
1 1
2 2
```

Output:

```
Case 1:
220
Case 2:
220 284
1184 1210
```

Test Case 2:

Input :

```
2
1 1
2 2
```

Output:

```
Case 1:
220
Case 2:
220 284
1184 1210
```

Test Case 3:

Input:

```
1
5 5
```

Output:

```
Case 1:
220 284 1184 1210 2620
2924 5020 5564 6232 6368
10744 10856 12285 14595 17296
18416 63020 66928 66992 67095
69615 71145 76084 79750 87633
```

Test Case 4:

Input:

```
2
10 10
10 20
```

Output:

```
Case 1:
220 284 1184 1210 2620 2924 5020 5564 6232 6368
10744 10856 12285 14595 17296 18416 63020 66928 66992 67095
69615 71145 76084 79750 87633 88730 100485 122265 122368 123152
124155 139815 141664 142310 153176 168730 171856 176272 176336 180848
185368 196724 202444 203432 280540 308620 319550 356408 365084 389924
399592 430402 437456 455344 469028 486178 503056 514736 522405 525915
600392 609928 624184 635624 643336 652664 667964 669688 686072 691256
712216 726104 783556 796696 802725 863835 879712 898216 901424 947835
980984 998104 1043096 1077890 1099390 1125765 1154450 1156870 1175265 1185376
1189150 1280565 1286744 1292570 1328470 1340235 1358595 1392368 1438983 1464592
Case 2:
220 284 1184 1210 2620 2924 5020 5564 6232 6368 10744 10856 12285 14595 17296 18416
63020 66928 66992 67095
69615 71145 76084 79750 87633 88730 100485 122265 122368 123152 124155 139815 141664
142310 153176 168730 171856 176272 176336 180848
185368 196724 202444 203432 280540 308620 319550 356408 365084 389924 399592 430402
437456 455344 469028 486178 503056 514736 522405 525915
600392 609928 624184 635624 643336 652664 667964 669688 686072 691256 712216 726104
783556 796696 802725 863835 879712 898216 901424 947835
980984 998104 1043096 1077890 1099390 1125765 1154450 1156870 1175265 1185376 1189150
1280565 1286744 1292570 1328470 1340235 1358595 1392368 1438983 1464592
1466150 1468324 1483850 1486845 1511930 1598470 1669910 1747930 1749212 1798875
1870245 2062570 2082464 2090656 2236570 2429030 2652728 2723792 2728726 2739704
2802416 2803580 2874064 2928136 2941672 2947216 3077354 3276856 3606850 3716164
3721544 3786904 3805264 3892670 4006736 4238984 4246130 4259750 4300136 4314616
4445050 4482765 4488910 4532710 4604776 5120595 5123090 5147032 5162744 5232010
5357625 5385310 5459176 5495264 5504110 5684679 5726072 5730615 5799542 5812130
```

```
5843048 5864660 6088905 6135962 6329416 6369928 6371384 6377175 6680025 6955216
6993610 7158710 7275532 7288930 7418864 7471508 7489112 7489324 7577350 7674088
7677248 7684672 7800544 7850512 7916696 8052488 8221598 8262136 8369864 8493050
8619765 9071685 9199496 9339704 9363584 9437056 9498555 9592504 9627915 9892936
```

Test Case 5:

Input:

```
5
1 5
5 1
3 4
4 3
6 8
```

Output:

```
Case 1:
220 284 1184 1210 2620
Case 2:
220
284
1184
1210
2620
Case 3:
220 284 1184 1210
2620 2924 5020 5564
6232 6368 10744 10856
Case 4:
220 284 1184
1210 2620 2924
5020 5564 6232
6368 10744 10856
Case 5:
220 284 1184 1210 2620 2924 5020 5564
6232 6368 10744 10856 12285 14595 17296 18416
63020 66928 66992 67095 69615 71145 76084 79750
87633 88730 100485 122265 122368 123152 124155 139815
141664 142310 153176 168730 171856 176272 176336 180848
185368 196724 202444 203432 280540 308620 319550 356408
```

Test Case 6:

Input:

```
1
7 7
```

Output:

```
Case 1:
```

```
220 284 1184 1210 2620 2924 5020
5564 6232 6368 10744 10856 12285 14595
17296 18416 63020 66928 66992 67095 69615
71145 76084 79750 87633 88730 100485 122265
122368 123152 124155 139815 141664 142310 153176
168730 171856 176272 176336 180848 185368 196724
202444 203432 280540 308620 319550 356408 365084
```

Solution :

```
#include <bits/stdc++.h>
using namespace std;

#define MAX 10000007
typedef long long ll;

ll divisor[MAX],vis[MAX],mark[MAX];
ll v[300],x=0;

void genDiv(ll n) {
     for(ll i=1; i<=n; ++i) {
          for(ll j = i+i; j<=n; j += i) {
               divisor[j] += i;
          }
     }
     for(ll i=1; i<=n; ++i) {
          if(divisor[divisor[i]] == i && i != divisor[i] &&
divisor[i] <= MAX) {
               vis[i] = 1;
               vis[divisor[i]] = 1;
          }
     }
     for(ll i=1; i<=n; ++i ) {
          if(vis[i])
               v[x++] = i;
     }
}

int main()
{
     genDiv(MAX);
     ll T,n,m,i,p;
     scanf("%lld",&T);
     for(i = 1; i<= T; ++i )  {
          p=0;
          scanf("%lld %lld",&n,&m);
          printf("Case %lld:\n",i );
          for(ll j = 0; j < n; ++j ) {
               for(ll k = 0; k < m; ++k ) {
                    printf("%lld ", v[p++]);
               }
               printf("\n");
          }
     }
     return 0;
}
```