

Assalamualaikum

Mohammad Hanif Furqan Aufa Putra

5025221161

TUGAS 1 PBO C

Friendly Numbers

Problem:

Two different natural numbers are called friendly if the first one is the sum of divisors of the second number, with the exception of the second number, and the second is the sum of divisors of the first day, except for the very first day. Want to find all the pairs of friendly numbers, both of which belong to the interval from M to N.

Input data

In the first row are integers M and N ($1 \leq M \leq N \leq 1\,000\,000$).

Output data

In each line of output from a pair of numbers separated by spaces. The first number pair must be less than the second. Rows must be sorted in ascending order of the first of the pair. If the pairs of friendly numbers in the interval not to withdraw "Absent".

Examples

Input example #1

200 300

Output example #1

220 284

Interpretasi saya setelah membaca deskripsi dan memahami definisi yang ada dalam soal dapat dipahami adalah

Tujuan dari soal ini adalah mencari semua "Friendly Numbers" dalam interval yang ditentukan dengan input yaitu M sampai N. Friendly Number ini adalah pasangan bilangan yang memiliki karakteristik yaitu jika yang bilangan pertama adalah jumlah dari semua pembagi dari bilangan kedua, kecuali bilangan kedua itu sendiri, dan sebaliknya.

Input terdiri dari dua bilangan bulat M dan N ($1 \leq M \leq N \leq 1.000.000$), yang merupakan batasan interval pencarian pasangan bilangan "friendly".

Setiap baris output akan berisi sepasang bilangan yang merupakan bilangan "friendly", dipisahkan oleh spasi. Bilangan pertama dari sepasang tersebut harus lebih kecil dari bilangan kedua. Hasil output harus diurutkan dalam urutan meningkat berdasarkan bilangan pertama

dari setiap pasangan. Jika tidak ada pasangan bilangan "friendly" dalam interval yang diberikan, maka outputnya adalah "Absent".

Untuk contoh $N = 200$ dan $M = 300$ atau range 200- 300 sebagai berikut:

Bilangan Friendly Numbers dengan Range 200-300	Faktorisasi	Jumlah Faktorisasi tanpa bilangan ini sendiri
220	1, 2, 4, 5, 10, 11, 20, 44, 55, 110, [220]	284
284	1, 2, 4, 71, 142, [284]	220

Lalu setelah itu mencari konsep yang paling mendekati dengan logika soal tersebut, setelah mencari ternyata ditemukan konsep keprimaan. Dan selanjutnya mencari algoritma yang memenuhi konsep tersebut yaitu Sieve of Eratosthenes. Algoritma Sieve of Eratosthenes adalah algoritma mencari semua bilangan prima dengan limit yang ditentukan, algoritma ini paling mendekati logika problem ini dimana semua bilangan faktorisasi merupakan bilangan prima.

Dengan interpretasi tersebut, dicoba membuat sebuah code untuk percobaan dengan algoritma Sieve of Eratosthenes sebagai berikut:

```
1 #include <stdio.h>
2 using namespace std;
3 const int MXN = 1000000;
4 int sieve[MXN+1];
5
6 int main(){
7     int m, n;
8     for(int i = 0; i <= MXN; i++) sieve[i]
    = 0;
9     for(int i = 1; i <= MXN; i++){
10         for(int j = 2*i; j <= MXN; j+=i){
11             sieve[j] += i;
12         }
13     }
14     while(scanf("%d %d", &m, &n) != EOF){
15         bool ok = 0;
16         for(int i = m; i <= n; i++){
17             int j = sieve[i];
18             if(i < j && m <= j && j <= n &&
    sieve[j] == i){
19                 printf("%d %d\n", i, j);
20                 ok = 1;
21             }
22         }
23         if(!ok) puts("Absent");
24     }
25     return 0;
26 }
27
```

Dengan code ini dan menggunakan constraints “1 1000000” sebagai uji coba diperoleh

220 284

1184 1210

2620 2924

5020 5564

6232 6368

10744 10856

12285 14595

17296 18416

63020 76084

66928 66992

67095 71145

69615 87633

79750 88730

100485 124155

122265 139815

122368 123152

141664 153176

142310 168730

171856 176336

176272 180848

185368 203432

196724 202444

280540 365084

308620 389924

319550 430402

356408 399592

437456 455344

469028 486178

503056 514736

522405 525915

600392 669688

609928 686072

624184 691256

635624 712216

643336 652664

667964 783556

726104 796696

802725 863835

879712 901424

898216 980984

Yang berjumlah 40 pasangan yang mana jika dilihat lagi dari soal tersebut tidak disebut akan ada berapa testcase interval yang akan diinput yang berarti soal menggunakan multi test case, maka dari itu akan dilakukan perubahan pada input menggunakan input dengan loop yang menggunakan “!=EOF” agar input tetap berjalan. Lalu dengan semua pasangan yang ada pada constrain hanya berjumlah 40, maka lebih baik untuk menyimpan semua pasangan bilangan "friendly" yang ada dalam constrain dalam array untuk menghindari perulangan yang tidak perlu.

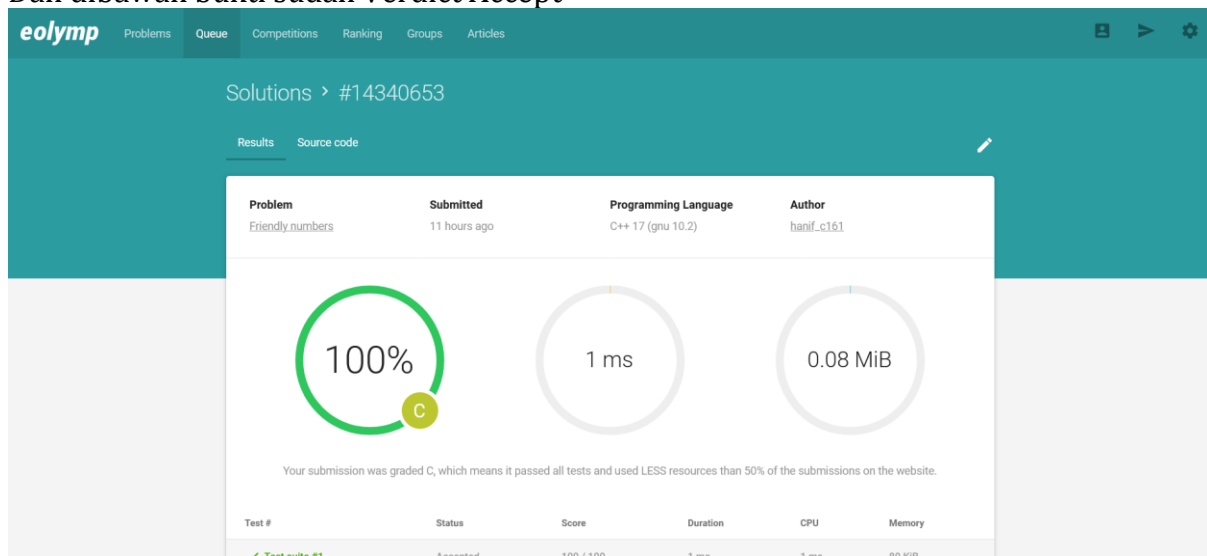
Dan dibawah ini merupakan keseluruhan code

```
1  #include <stdio.h>
2  #include <algorithm>
3  using namespace std;
4
5  pair<int, int> arr[41] = {
6      {220, 284}, {1184, 1210}, {2620, 2924}, {5020, 5564}, {6232, 6368}, {10744,
7      10856}, {12285, 14595}, {17296, 18416}, {63020, 76084}, {66928, 66992}, {67095,
8      71145}, {69615, 87633}, {79750, 88730}, {100485, 124155}, {122265, 139815},
9      {122368, 123152}, {141664, 153176}, {142310, 168730}, {171856, 176336}, {176272,
10     180848}, {185368, 203432}, {196724, 202444},
11     {280540, 365084}, {308620, 389924}, {319550, 430402}, {356408, 399592}, {437456,
12     455344}, {469028, 486178}, {503056, 514736},
13     {522405, 525915}, {600392, 669688}, {609928, 686072}, {624184, 691256}, {635624,
14     712216}, {643336, 652664}, {667964, 783556},
15     {726104, 796696}, {802725, 863835}, {879712, 901424}, {898216, 980984}};
16
17 int main()
18 {
19     int m, n;
20     while (scanf("%d %d", &m, &n) != EOF)
21     {
22         bool ok = 0;
23         for (int i = 0; i < 40; i++)
24         {
25             if (m <= arr[i].first && arr[i].first <= n &&
26                 m <= arr[i].second && arr[i].second <= n)
27             {
28                 printf("%d %d\n", arr[i].first, arr[i].second);
29                 ok = 1;
30             }
31         }
32         if (!ok)
33             puts("Absent");
34     }
35 }
```

Dengan menjalankan program tersebut diperoleh hasil seperti dibawah ini

```
PS F:\Coding> cd "f:\Coding\PBO-2023\Friendly-Numbers\" ; if ($?) { g++ fncorrect.cpp -o fncorrect } ; if ($?) { .\fnincorrect }
200 300
220 284
```

Dan dibawah bukti sudah Verdict Accept



Terima Kasih telah membaca, Waalaikumsalam wr wb.