**NASKAH SOAL DAN PEMBAHASAN**

**ALPHA TRY OUT 5**

**BIDANG: INFORMATIKA**

**Problem Setter : Abdan Hafidz**

**MESIN AJAIB**

Sebuah mesin dapat dijalankan dan dalam Q operasi mesin tersebut, Pak Dengklek akan menerima beberapa bilangan yaitu X dan Y Ia ingin membuat tumpukan bilangan dari bilangan X dan Y yang ia terima dengan cara. Jika saat ini ya menerima X > Y maka letakkan X terlebih dahulu baru Y. Jika X < Y maka letakkan Y terlebih dahulu baru X. Jika X = Y maka letakkan salah satu pada tumpukan.

Semua bilangan yang ia terima ditumpuk kemudian ia ingin mengeluarkan N bilangan teratas pada tumpukkan. Bantulah Pak Dengklek untuk mengeluarkan bilangan – bilangan tersebut dari tumpukkan.

Sebagai contoh ia melakukan 5 operasi dan mendapatkan X dan Y sebagai berikut

|  |  |  |
| --- | --- | --- |
| **Qi** | **X** | **Y** |
| 1 | 3 | 2 |
| 2 | 1 | 4 |
| 3 | 1 | 1 |
| 4 | 0 | 5 |
| 5 | 0 | 0 |

Dari operasi tersebut ia mampu membuat tumpukkan angka menjadi

|  |
| --- |
| 0 |
| 0 |
| 5 |
| 1 |
| 1 |
| 4 |
| 2 |
| 3 |

Sehingga jika ia mengeluarkan 3 angka teratas adalah deretan **0 0 5.**

1. Jika dilakukan operasi sebanyak Q = 15 kali dan diperoleh masing masing nilai (X,Y) pada setiap operasi adalah [(2, 7), (8, 3), (1, 6), (4, 9), (5, 2), (10, 4), (7, 0), (3, 8), (6, 5), (9, 10), (0, 1), (5, 3), (8, 6), (3, 0), (2, 9)] tentukan 7 angka teratas pada tumpukkan! **{tuliskan jawaban berupa daftar angka tanpa dipisahkan spasi, contohnya jika 7 tumpukkan angka adalah 11,12,13,14,1,2,3 maka tuliskan 11121314123}!**

**JAWABAN :** 2903683

**Pembahasan :**

(5, 3), (8, 6), (3, 0), (2, 9)

(5,3), (8,6), (3,0), (9,2)

Stack = Last In First Out

Untuk soal ini mudahnya adalah pada setiap tupel X,Y anda tinggal mengubah posisi (P,Q) sehingga P > Q jika X !=Y dan jika X = Y maka hapus salah satunya.

Ambil 7 angka teratas [3, (8, 6), (3, 0), (2, 9)] urutkan setiap tupel sehingga menjadi [3,(8,6),(3,0),(9,2)]

7 angka teratas adalah 2903683

1. Jika diperoleh tumpukkan angka sebanyak 98928664,berapa nilai Q minimum dan maksimum berturut – turut ?**{tuliskan jawaban dipisahkan dengan spasi}**

**JAWABAN : 49464332 197857328**

**Q minimum** adalah saat nilai X dan Y pada masing – masing operasi ke Qi berbeda, sehingga semua angka ditumpuk dan banyaknya angka pada tumpukkan adalah 2Qmin.

**Q maksimum** adalah saat nilai X dan Y pada masing – masing operasi ke Qi bernilai sama sehingga angka yang dimasukkan pada tumpukkan hanya satu angka. Ini membuat dalam Q operasi banyak angka pada tumpukkan adalah sebanyak 1/2Qmax

2Qmin = 98928664

Qmin = 49464332

1/2Qmax = 98928664

Qmax = 2.98928664 = 197857328

Diperoleh Qmin Qmax = 49464332 197857328.

1. Jika Q = 20, berapa banyak angka maksimum yang ada pada tumpukan? **{tuliskan jawaban berupa angka saja}**

JAWABAN : 40

1. **Membuat Program Sederhana (Output Only)**

Buatlah program menggunakan bahasa C/C++ untuk membantu perhitungan dari kasus uji yang diberikan. Kemudian masukkan keluaran dari setiap kasus uji sub soal pada kolom tersedia dengan melakukan *copy-paste* dari output program anda!

**Format Masukan**

Baris pertama berisi bilangan bulat Q dan N.

Q baris berikutnya masing – masing berisikan Xi dan Yi yang menyatakan bilangan X dan Y yang diterima Pak Dengklek pada operasi ke i untuk (1 < i < Q).

**Format Keluaran**

Keluarkan N bilangan teratas pada tumpukkan.

**Contoh Masukan dan Keluaran**

|  |  |
| --- | --- |
| **Masukan** | **Keluaran** |
| 5 3  3 2  1 4  1 1  0 5  0 0 | 0 0 5 |

**Subsoal 1 (Nilai = 50%)**

Untuk kasus uji sebagai berikut

|  |
| --- |
| 20 15  472 53  97 779  331 409  966 861  165 58  841 561  582 86  376 921  806 307  611 885  217 866  831 78  798 137  320 765  89 122  67 126  675 943  571 682  850 358  655 809 |

JAWABAN :

|  |
| --- |
| 655  809  358  850  571  682  675  943  67  126  89  122  320  765  137 |

**{TULISKAN JAWABAN PADA KOLOM INPUT BERDASARKAN HASIL COPY PASTE DARI OUTPUT PROGRAM}**

**Subsoal 2 (Nilai = 50%)**

Q = 109

N = 20

Xi = (i\*2 + 3)

Yi = (3i)

X = Y tidak akan pernah

Pembahasan :

Untuk soal ini kita tahu bahwa karena N = 20 akan selalu ditemukan Xi < Yi kecuali saat i = 1, sehingga tumpukkan akan menjadi :

|  |
| --- |
| X109 |
| Y109 |
| .. |
| X3 |
| Y3 |
| X2 |
| Y2 |
| X1 |

Ditemukan bahwa program akan mencetak Xi Yi untuk 109 – 20 < i < 109. Sehingga didapat cetakan program adalah barisan

(109\*2 + 3)

(109\*3)

((109 – 1)\*2 + 3)

((109 – 1)\*3)

...

((109 – 19)\*2 + 3)

((109 – 19)\*3)

Solusi Program :

|  |
| --- |
| #include <iostream>  using namespace std;  int main()  {  for(long long i = 1e9; i>=1e9 - 19; i--){  cout<<i\*2+3<<endl;  cout<<3\*i<<endl;  }  return 0;  } |

JAWABAN

|  |
| --- |
| 2000000003  3000000000  2000000001  2999999997  1999999999  2999999994  1999999997  2999999991  1999999995  2999999988  1999999993  2999999985  1999999991  2999999982  1999999989  2999999979  1999999987  2999999976  1999999985  2999999973  1999999983  2999999970  1999999981  2999999967  1999999979  2999999964  1999999977  2999999961  1999999975  2999999958  1999999973  2999999955  1999999971  2999999952  1999999969  2999999949  1999999967  2999999946  1999999965  2999999943 |

**{TULISKAN JAWABAN PADA KOLOM INPUT BERDASARKAN HASIL COPY PASTE DARI OUTPUT PROGRAM}**

**Solusi Soal Secara Umum**

|  |
| --- |
| #include <iostream>  #include <stack>  using namespace std;  int main()  {  stack <int> stack;  int Q,N,X,Y;  cin>>Q>>N;  while(Q--){  cin>>X>>Y;  if(X>Y){  stack.push(X);  stack.push(Y);  }else if(X == Y){  stack.push(X);  }else{  stack.push(Y);  stack.push(X);  }  }    while(N--){  int atas = stack.top();  cout<<atas<<endl;  stack.pop();  }  return 0;  } |

**OSN TAHUN LALU OFFLINE**

Akhirnya setelah Pandemi berlalu, pelaksanaan Olimpiade Sains Nasional (OSN) tahun ini diselenggarakan secara offline / berlangsung di tempat terbuka lebih tepatnya di Kabupaten Bogor, Jawa Barat. Tentunya rangkaian acara OSN tidak hanya berkompetisi, tapi juga akan ada wisata bersama seluruh peserta OSN lainnya. Dalam wisata edukasi kali ini Panitia membutuhkan 𝑁 orang pemandu yang akan memandu 𝑋 orang peserta. Setiap peserta harus memilih satu dari beberapa pemandu yang tersedia dan harus mengikuti pemandu tersebut. Tidak ada pemandu yang tidak memandu pesertanya, sehingga nantinya akan terbentuk beberapa kelompok wisata dengan pemandunya masing – masing.

1. Jika N = 3, dan X = 9. Tentukan banyaknya konfigurasi kelompok wisata yang mungkin terbentuk! {jawaban berupa angka}

**Pembahasan :**

**Jawaban : 39 – 3 = 19680**

1. Untuk 2023 orang pemandu, berapa minimal nilai X sehingga bisa dibentuk kelompok Wisata yang sesuai?

**Jawaban : 2023**

Agar bisa dibentuk kelompok yang sesuai minimal X = N (sehingga setiap pemandu bisa memandu minimal satu orang)

1. Untuk N = 2023, X = 99 , misalkan M adalah banyak konfigurasi kelompok wisata yang mungkin terbentuk. Tentukanlah nilai M mod 11 !

**Jawaban : 0**

**202399 − 2023 𝑚𝑜𝑑 11 = ((2023 𝑚𝑜𝑑 11)99 𝑚𝑜𝑑 11 − 2023 𝑚𝑜𝑑 11) 𝑚𝑜𝑑 11 = 0**

1. **Membuat Program Sederhana (Output Only)**

Buatlah program menggunakan bahasa C/C++ untuk membantu perhitungan dari kasus uji yang diberikan. Kemudian masukkan keluaran dari setiap kasus uji sub soal pada kolom tersedia dengan melakukan *copy-paste* dari output program anda!

**Format Masukan**

Baris pertama berisi bilangan bulat 𝑇 yang menyatakan banyaknya kasus uji.

𝑇 barisnya berisikan setiap kasus uji yang memuat dua buah bilangan bulat positif 𝑁 dan

𝑋.

**Format Keluaran**

Terdiri dari 𝑇 jawaban yang menyatakan penyelesaian banyak konfigurasi kelompok wisata yang mungkin terbentuk untuk setiap kasus uji dimodulo dengan 109.

**Contoh Masukan dan Keluaran**

|  |  |
| --- | --- |
| **Masukan** | **Keluaran** |
| 3  1 2  2 3  3 4 | 1  6  78 |

Penjelasan Contoh 1 :

1 orang pemandu hanya bisa membentuk satu kelompok yang terdiri dari 2 orang

Penjelasan Contoh 2 :

2 orang pemandu bisa membentuk 6 konfigurasi kelompok

Contoh konfigurasinya adalah Pemandu pertama memandu 2 orang siswa misalkan A dan B dan pemandu kedua memandu 1 orang siswa yaitu C. Pemandu pertama memandu 2 orang siswa, misalkan A dan C dan pemandu kedua memandu 1 orang siswa yaitu B, dan seterusnya.

**Subsoal 1 (Nilai = 50%)**

Untuk kasus uji sebagai berikut

|  |
| --- |
| 15  31 44  49 45  35 27  46 5  5 20  10 2  29 20  18 32  33 19  39 50  22 25  12 31  21 8  17 28  3 26 |

JAWABAN :

|  |
| --- |
| 79150690  0  0  0  431640620  0  0  279154158  0  491557962  312722410  497177076  0  366063024  865828326 |

**Subsoal 2 (Nilai = 50%)**

Untuk kasus uji sebagai berikut :

T = 50

Ni = floor(i/7), Xi = i – 1, untuk (1<i<T)

**JAWABAN :**

Anda cukup menjalankan kompilasi potongan program

|  |
| --- |
| for(int i = 1; i<=50; i++){  N = i/7;  X = i - 1;  solve();  } |

|  |
| --- |
| 1  0  0  0  0  0  1  1  1  1  1  1  1  8190  16382  32766  65534  131070  262142  524286  486784398  460353200  381059606  143178824  429536478  288609440  865828326  509481980  37927932  151711740  606846972  427387900  709551612  838206460  72265620  361328120  806640620  33203120  166015620  830078120  150390620  75859450  455156730  730940410  385642490  313854970  883129850  298779130  660188794  621321600 |

**Solusi soal secara umum**

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  long long T,N,X,MOD = 1000000000;  long long pangkat(long long x,long long y){  int temp;  if (y == 0)  return 1;  temp = pangkat(x, y / 2);  if (y % 2 == 0)  return (temp%MOD \* temp%MOD)%MOD;  else  return (x%MOD \* temp%MOD \* temp%MOD)%MOD;  }  void solve(){  if(X<N){  cout<<0<<endl;  }else if(N == 1){  cout<<1<<endl;  }  else{  long long res = (pangkat(N,X) - N%MOD)%MOD;  cout<<res<<endl;  }  }  int main()  {  cin>>T;  while(T--){  cin>>N>>X;  solve();  }  return 0;  } |

**Siswa Hebat SMA U.A**

Midoriya dan Uraraka duduk berdua di taman A untuk mengerjakan PR bersama. Namun sayangnya pada waktu bersamaan ada dua pasangan lainnya yang mengerjakan PR juga di taman A. Ada Kaminari dan Jiro, juga Yaoyoruzu dan Todoroki. Melihat kondisi ini Midoriya dan Uraraka ingin mengatur jadwal agar mereka bisa fokus mengerjakan PR berdua di taman dan tidak ada orang lain, begitupun dengan pasangan lainnya. Mereka harus membuat jadwal. Selama N tahun mereka membuat kesepakatan bahwa Midoriya dan Uraraka akan ke taman setiap X hari sekali, Kaminari dan Jiro Y hari sekali, Yaoyoruzu dan Todoroki Z hari sekali. Mereka bertanya - tanya selama N tahun tersebut berapa kali mereka bisa menikmati waktu bersama hanya berdua tanpa diganggu orang lain. Gunakan keterangan bahwa di dunia mereka saat ini, satu tahun sama dengan 365 hari**.**

1. Jika N=5,X=4,Y=3,Z=2. Tentukan berapa kali Midoriya dan Uraraka duduk di taman berdua tanpa ada orang lain! **{Jawaban berupa angka bulat}**

JAWABAN : 0

5 x 365 hari berapa kali Midoriya dan Ururaka hanya mereka yang ada di Taman?

Ups semua hari berdua Midoriya dan Uraraka juga hari berdua Yaoyoruzu dan Todoroki

Jika N=3,X=5,Y=2,Z=2.

1. Tentukan berapa kali Yaoyoruzu dan Todoroki duduk di taman berdua tanpa ada orang lain! **{Jawaban berupa angka bulat}**

JAWABAN : 0

Perhatikan Y = Z

1. Jika N=6,X=4,Y=3,Z=5. Apakah dapat dipastikan minimal ada dua pasangan yang bertemu di taman secara bersamaan **{Jawaban berupa YA / TIDAK}**

JAWABAN : YA

1. **Membuat Program Sederhana (Output Only)**

Buatlah program menggunakan bahasa C/C++ untuk membantu perhitungan dari kasus uji yang diberikan. Kemudian masukkan keluaran dari setiap kasus uji sub soal pada kolom tersedia dengan melakukan *copy-paste* dari output program anda!

**Format Masukan**

Baris pertama berisi bilangan bulat N, X,Y, dan Z.

**Format Keluaran**

Berisi satu baris berisikan jawaban selama N tahun tersebut berapa kali masing – masing pasangan ( Midoriya dan Uraraka, Kaminari dan Jiro, serta Yaoyoruzu dan Todoroki) bisa menikmati waktu bersama hanya berdua tanpa diganggu orang lain dipisahkan oleh spasi.

**Contoh Masukan dan Keluaran**

|  |  |
| --- | --- |
| **Masukan** | **Keluaran** |
| 1 1 2 3 | 122 0 0 |
| 1 3 4 5 | 73 49 37 |
| 2 1 1 1 | 0 0 0 |

**Subsoal 1 (100%)**

Hanya berisikan kasus uji sebagai berikut

|  |
| --- |
| 98927784882 1988929884 989928994 3887728843 |

JAWABAN

|  |
| --- |
| 18154 36475 9287 |

Penjelasan Contoh 1 :

Selain hari kelipatan 2 atau 3 Midoriya dan Uraraka bisa berdua di taman A. Sedangkan pasangan lainnya sama sekali tidak bisa berdua karena pada hari mereka akan selalu ada Midoriya dan Uraraka yang setiap harinya berdua di taman.

Solusi Umum :

|  |
| --- |
| #include <bits/stdc++.h>  #define ll long long  using namespace std;  ll kpk(ll x, ll y){  return (x\*y)/(\_\_gcd(x,y));  }  int main()  {  ll N,X,Y,Z,p1,p2,p3,inter;  cin>>N>>X>>Y>>Z;  ll hari = N\*365;  inter = (hari/kpk(kpk(X,Y),Z));  p1 = (hari/X) - ( (hari/kpk(X,Y))+ (hari/kpk(X,Z)) - inter);  p2 = (hari/Y) - ( (hari/kpk(X,Y))+ (hari/kpk(Y,Z)) - inter);  p3 = (hari/Z) - ( (hari/kpk(X,Z))+ (hari/kpk(Y,Z)) - inter);  cout<<p1<<" "<<p2<<" "<<p3<<endl;  return 0;  } |

PEMBAHASAN :

**Dengan menggunakan konsep teori himpunan, aturan inklusi – eksklusi kita bisa terapkan :**

* Untuk menentukan berapa kali Midoriya dan Uraraka duduk di taman berdua tanpa ada orang lain, maka kita akan mencari banyak hari yang hanya berkelipatan saja di antara hari namun bukan kelipatan atau . Sebanyak
* Untuk menentukan berapa kali Kaminari dan Jiro duduk di taman berdua tanpa ada orang lain, maka kita akan mencari banyak hari yang hanya berkelipatan saja di antara hari namun bukan kelipatan atau .
* Untuk menentukan berapa kali Yaoyoruzu dan Todoroki duduk di taman berdua tanpa ada orang lain, maka kita akan mencari banyak hari yang hanya berkelipatan saja di antara hari namun bukan kelipatan atau .

Jika X = Y , maka tentu saja untuk pasangan Midoriya dan Uraraka dan pasangan Kaminari dan Jiro keduanya sama sekali tidak mendapatkan kesempatan berdua saja di taman

Jika X = Z , maka tentu saja untuk pasangan Midoriya dan Uraraka dan pasangan Yaoyoruzu dan Todoroki keduanya sama sekali tidak mendapatkan kesempatan berdua saja di taman

Jika Y = Z , maka tentu saja untuk pasangan Kaminari dan Jiro dan pasangan Yaoyoruzu dan Todoroki keduanya sama sekali tidak mendapatkan kesempatan berdua saja di taman.

Anda bisa melakukan optimasi solusi dengan menguji sifat relatif prima antar X,Y, dan Z; tapi tidak terlalu peduli, cukup menggunakan teori himpunan inklusi – eksklusi di atas anda sudah bisa mendapatkan jawaban mengingat kasus uji yang tidak begitu besar.

**Ngulinya Dengklek**

Pak Dengklek sedang berada di dunia Anime berupa grid berukuran N x M. Ia ingin bergerak dari (1,1) menuju petak (X,Y) di mana X menyatakan baris ke-X dan Y menyatakan kolom ke-Y. Sembari berjalan ia ingin mengumpulkan item sebanyak mungkin. Setiap petak berisikan bilangan bulat non negatif Ai,j yang menyatakan banyak item yang ada pada petak (i,j) tersebut. Diketahui dari suatu posisi ia dapat bergerak ke petak selanjutnya yang berada pada sebelah kanan atau di bawah dari posisinya saat ini.

Bantulah Pak Dengklek dalam menemukan item sebanyak mungkin.

Sebagai contoh Pak Dengklek dapat menempuh rute dari (1,1) menuju (4,4) (berikut ini:

**Start**

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 1 | 2 | 3 |
| 2 | 3 | 2 | 3 |
| 4 | 5 | 2 | 4 |
| 7 | 6 | 1 | 3 |

Sehingga ia mendapatkan item sebanyak 0 + 2 + 4 + 7 + 6 + 1 + 3 = 23

1. Jika diberikan grid berukuran 7 x 8 di bawah ini!

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 3 | 3 | 1 | 4 | 4 | 2 | 0 |
| 2 | 5 | 4 | 4 | 3 | 4 | 2 | 0 |
| 1 | 2 | 4 | 3 | 2 | 1 | 1 | 1 |
| 0 | 2 | 2 | 3 | 1 | 4 | 3 | 2 |
| 3 | 5 | 3 | 1 | 1 | 3 | 6 | 3 |
| 3 | 4 | 2 | 3 | 2 | 3 | 1 | 1 |
| 2 | 2 | 0 | 2 | 1 | 2 | 9 | 5 |

Pak Dengklek ingin bepergian dari (1,1) menuju baris ke-7 dan kolom ke-8 (7,8) tentukan berapa item terbanyak yang bisa ia kumpulkan!**{tuliskan jawaban dalam bentuk angka saja}**

Untuk soal ini kita bisa menggunakan metode Dynamic Programming Bottom – up.

Misalkan

*DP(i,j) = Total Item terbanyak yang bisa didapat jika bepergian dari (1,1) ke (i,j)*

*item[i][j] = Banyak item pada petak (i,j)*

*DP(i,j) = max(DP(i – 1,j),DP(i,j – 1)) + item[i][j]*

*Dengan kasus dasar*

*DP(1,1) = 1*

*DP(1,2) = 1 + 3 = 4*

*DP(2,1) = 1 + 2 = 3*

*DP(7,8) = 49*

*Anda dapat menggunakan bantuan program untuk mempermudah*

JAWABAN : 49

1. Pak Dengklek diberikan grid 4 x 4 berikut ini

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 |

Jika ia berjalan dari petak ujung kiri atas menuju ujung kanan bawah ada berapa banyak rute perjalanan yang bisa ia lakukan sehingga mendapatkan jumlah item maksimum? **{tuliskan jawaban dalam bentuk angka saja}**

**JAWABAN : 6**

**PEMBAHASAN :**

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 |

1. Diberikan grid berukuran 6 x 6. Pak Dengklek ingin berjalan dari petak ujung kiri atas menuju petak pada baris ke-3 dan kolom ke-5. Ada berapa banyak cara yang bisa lakukan?**{tuliskan jawaban dalam bentuk angka saja}**

**JAWABAN : 15**

(1,1) menuju (3,5)

Ia bisa bergerak 2 langkah ke bawah lalu 4 langkah ke kanan (DDRRRR) berikutnya kita tinggal menghitung permutasinya saja yaitu sebanyak :

Total cara yang bisa ia lakukan adalah sebanyak 15 cara.

1. **Membuat Program Sederhana (Output Only)**

Buatlah program menggunakan bahasa C/C++ untuk membantu perhitungan dari kasus uji yang diberikan. Kemudian masukkan keluaran dari setiap kasus uji sub soal pada kolom tersedia dengan melakukan *copy-paste* dari output program anda!

**Format Masukan**

Baris pertama berisikan bilangan M, N, X, dan Y.

M baris berikutnya berisikan sebanyak N bilangan Ai,j.

**Format Keluaran**

Keluarkan jawaban berupa total item terbanyak yang bisa Pak Dengklek dapatkan.

**Contoh Masukan dan Keluaran**

|  |  |
| --- | --- |
| **Masukan** | **Keluaran** |
| 4 4 4 4  0 1 2 3  2 3 2 3  4 5 2 4  7 6 1 3 | 23 |

**Subsoal 1 (100%)**

Hanya berisikan kasus uji sebagai berikut

|  |
| --- |
| 10 10 10 10  58 16 26 69 83 90 32 91 85 32  71 91 84 72 65 38 28 73 34 33  96 80 11 54 48 49 98 48 62 78  93 64 77 62 30 93 84 99 45 54  28 35 16 67 82 47 77 62 82 38  59 39 74 68 98 28 34 35 47 58  22 36 89 27 85 86 32 19 94 85  88 40 91 13 21 73 44 93 23 67  11 34 15 24 94 72 23 46 68 66  41 31 40 30 18 66 42 55 39 80 |

JAWABAN : 1273

Solusi soal secara umum :

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  #define ll long long  int N,M;  ll memo[100][100];  ll item[100][100];  ll dp(ll i, ll j){  if(i - 1 < 0 or j - 1 < 0 ) return 0;  else{  if(memo[i][j] != -1){  return memo[i][j];  }else{  return memo[i][j] = max(dp(i-1,j), dp(i,j-1)) + item[i][j];  }  }  }  int main()  {  int N,M,X,Y;  cin>>N>>M>>X>>Y;  for(int i = 0; i<N; i++){  for(int j = 0; j<M; j++){  memo[i][j] = -1;  cin>>item[i][j];  }  }  memo[0][0] = item[0][0];  memo[0][1] = item[0][0] + item[0][1];  memo[1][0] = item[0][0] + item[1][0];  cout<<dp(X - 1,Y - 1);  return 0;  } |

**Membeli Barang**

Bu Chanek sedang berbelanja di sebuah swalayan yang menjual N barang yaitu barang dengan harga P1,P2,P3, ... ,PN. Swalayan ini memiliki teknik pemasaran yang khusus yaitu menjual barang dengan nominal negatif. Bu Chanek yang merupakan istri seorang sultan yaitu Pak Chanek diberikan uang oleh Pak Chanek yang mana uangnya juga dapat bernilai negatif. Ada M lembar uang diberikan untuk Bu Chanek. Uang – uang tersebut masing – masing lembaran pecahan bernilai C1,C2,C3, ... ,CM

Pak Chanek yang sangat perhitungan ingin tahu, dengan skenario belanja yang ada berapakah hutang paling banyak yang mungkin didapatkan oleh Bu Chanek. Ia menghitung hutang ini dengan tujuan menetapkan batasan anggaran belanja. Bantulah ia dalam menyusun skenario anggaran belanjanya ini!

1. Jika tersedia barang dengan harga masing – masing [6, -14, -12, -8, 28, -13, 10, 26, -17, 22, -4, -6, -19, 23, 19, -5, 27, 11, 0, 24] dan uang yang diberikan kepada Bu Chanek adalah masing – masing lembar pecahan [11, 3, 6, -1, -13, 0, -9, -4, 3, 5, 10, 9, 8, 6, -16] berapa hutang paling banyak yang didapatkan Bu Chanek [hasil dapat berupa bilangan negatif] ? **{tuliskan jawaban dalam bentuk angka saja}**

JAWABAN : -239

Kita tahu bahwa hutang adalah saat

Uang dibayar - Harga Barang < 0

Dengan paradigma greedy kita akan membeli barang dengan harga semaksimal mungkin dan pembayaran seminimal mungkin (berhoetang).

Pilih barang yang positif bayar pakai uang negatif.

Beli barang dengan harga maksimum (+) : 6 + 28 + 10 + 26 + 22 + 23 + 19 + 27 + 11 + 24 = 196

Bayar dengan uang minimum / berhutang (-) : -1 -13 -9 -4 -16 = -43

Hutang maksimum yang ia peroleh adalah : -43 – 196 = -239

1. Tersedia barang dengan harga masing – masing [1,4,5,6,9,2] dan lembar pecahan uang [1,4,0,-2,5,2] ada berapa banyak cara Bu Chanek membeli tiga buah barang dengan total harga minimal 10 tanpa berhutang jika ia diwajibkan membayar barang apapun yang dibeli? **[tuliskan jawaban dalam bentuk angka saja}**

JAWABAN : 10

Pembahasan :

Pak Dengklek membeli barang dengan harga {1,4,5} = 10 bisa dibayar dengan 4 cara {(5,1,4),(5,2,4),(5,2,4,1),(5,-2,2,4,1)}, Total Cara pembelian = 1 x 4 = 4 cara

Pak Dengklek membeli barang dengan harga {1,4,6} = 11 bisa dibayar dengan 2 cara {(5,2,4),(5,2,4,1)}, Total Cara pembelian = 1 x 4 = 4 cara

Pak Dengklek membeli barang dengan harga {1,5,6} atau {1,9,2} = 12 bisa dibayar dengan 1 cara

{(1,4,5,2)}

Total cara pembelian = 2 x 1 = 2 cara

Total cara pembelian = 4 + 4 + 2 = 10 cara.

1. Tersedia barang dengan harga [1,9,8,-7,6,-2,5,3] dan uang dengan lembaran pecahan

[-3,2,0,1,5,6,7]. Berapa lembar minimal yang harus di keluarkan oleh Bu Chanek jika ia ingin membeli sebuah barang sehingga dapat dipastikan ia tidak berhutang sama sekali? {**tuliskan jawaban dalam bentuk angka saja}**

**JAWABAN : 6**

Pigeonhole Principle (PHP)

Kasus terburuk adalah ia membeli barang dengan harga 9 lalu ia membayar dengan uang [-3,0,1,2,5] sehingga ia berhutang -4. Agar ia tidak berhutang ia mengambil satu lembar lagi uang dengan nominal 6 atau 7. Sehingga minimal ia harus mengambil 5 + 1 = 6 lembar uang.

1. **Membuat Program Sederhana (Output Only)**

Buatlah program menggunakan bahasa C/C++ untuk membantu perhitungan dari kasus uji yang diberikan. Kemudian masukkan keluaran dari setiap kasus uji sub soal pada kolom tersedia dengan melakukan *copy-paste* dari output program anda!

**Format Masukan**

Baris pertama berisi bilangan bulat N dan M.

Baris kedua berisi N bilangan bulat Pi untuk (1<i<N)

Baris kedua berisi M bilangan bulat Cj untuk (1<j<M)

**Format Keluaran**

Keluarkan satu baris jawaban berupa hutang paling banyak yang mungkin didapatkan oleh Bu Chanek

**Contoh Masukan dan Keluaran**

|  |  |
| --- | --- |
| **Masukan** | **Keluaran** |
| 3 4  5 2 -1  -1 -2 3 2 | -10 |

Penjelasan Contoh :

Bu Chanek membeli barang pertama dan barang kedua dengan total harga 5 + 2 = 7

Kemudian ia membayarnya dengan uang pecahan -1 dan -2 dengan total uang = -3. Sehingga dengan membayar belanjaan senilai 7 dengan uang -3 mendapatkan hutang -10.

**Subsoal 1 (Nilai = 50%)**

Hanya berisi kasus uji sebagai berikut

N = 12, M = 15

P = [5, -8, 17, -2, 10, -15, 20, -20, -8, 12, 3, -11]

C = [6, -19, 13, -8, 17, 4, -11, -1, 10, 20, -5, -15, 8, 2, -7]

JAWABAN : -133

**Subsoal 2 (Nilai = 50%)**

Hanya berisi kasus uji sebagai berikut

|  |
| --- |
| 93 81  18 -15 -5 -9 10 -7 -11 -16 -2 2 14 -7 20 -2 6 0 -19 8 -19 15 4 12 -4 -16 -13 19 -17 -4 -20 -13 -12 12 4 11 -9 -4 18 2 -9 -5 8 2 -16 11 -15 14 14 -6 1 -8 6 18 11 -5 -1 -12 -10 -6 -10 13 -1 -13 -12 -10 -15 -19 -1 13 17 -7 7 -9 -5 -3 7 20 -19 -11 3 -1 -7 -9 7 -14 -5 1 -3 20 -15 -6 0 20 -19  35 -9 14 27 -17 68 -13 84 -2 76 -8 41 45 56 23 30 70 88 71 -19 47 92 91 -8 33 85 29 -10 97 60 -14 -2 69 3 77 74 41 14 -5 91 0 68 95 51 34 57 32 81 64 -8 17 29 -20 42 -9 88 36 94 66 57 24 -6 85 14 -7 -13 46 73 54 3 78 63 7 74 39 85 82 80 41 55 -16 |

JAWABAN : -855

Cukup lakukan penyelesaian secara Greedy, solusi secara umum adalah sebagai berikut :

|  |
| --- |
| #include <iostream>  using namespace std;  int main(){  int N,M;  cin>>N>>M;  int harga = 0, bayar = 0;  int barang,uang;  for(int i = 0; i<N ; i++){  cin>>barang;  if(barang > 0){  harga+=barang;  }  }  for(int i = 0; i<M ; i++){  cin>>uang;  if(uang < 0){  bayar+=uang;  }  }  cout<<bayar - harga<<endl;  } |

**Temenin Dengklek**

Di dalam kelas ada N orang bernama 1,2,3,4, ..., N. Dalam rangka keperluan acara kampus, Pak Dengklek mempersiapkan kelompok panitia di mana di dalamnya terdiri dari minimal satu orang dari kelas. Sebuah kelompok Panitia dapat terbentuk jika setiap anggotanya minimal mengenal atau dikenal oleh satu orang lainnya di dalam kelompok atau di kelompok tersebut hanya berisi satu orang saja. Seseorang U bisa saja mengenal orang lainnya yaitu V dengan syarat U V. Pak Dengklek penasaran jika diberikan informasi beberapa keterangan pasangan (U,V) dari N orang yang ada berapa banyak kelompok panitia minimal yang bisa Pak Dengklek bentuk.

Bantulah ia menghitung banyak kelompok minimal yang dapat Pak Dengklek bentuk!

1. Jika diberikan N = 10 dan beberapa keterangan orang yang saling mengenal :

**1 mengenal 2 , 1 mengenal 3 , 2 mengenal 4, 4 mengenal 3, 5 mengenal 6, 6 mengenal 7, 7 mengenal 9, 8 mengenal 10**

Berapa banyak kelompok minimal yang dapat terbentuk?**{tuliskan jawaban dalam bentuk angka saja}**

**JAWABAN : 3**

Pembahasan :

**Kelompok 1 : 1,2,3,4**

**Kelompok 2 : 5,6,7,9**

**Kelompok 3 : 8,10**

1. Untuk N = 100 kemudian ditemukan bahwa seseorang i mengenal orang lainnya yaitu i + 3 jika dan hanya 1 < i, i+3 < N berapa banyak kelompok minimal yang dapat terbentuk? **{tuliskan jawaban dalam bentuk angka saja}**

**JAWABAN : 3**

Akan terbentuk 3 kelompok

Kelompok 1 : 1,4,7,10, ..., 100

Kelompok 2 : 2,5,8, ... , 98

Kelompok 3 : 3,6,9, ..., 99

1. Diberikan N = 103 dan beberapa keterangan orang yang saling mengenal :

**1 mengenal 3 , 2 mengenal 3 , 2 mengenal 5, 5 mengenal 6, 7 mengenal 9, 8 mengenal 10**

Berapa banyak kelompok minimal yang dapat dibentuk?**{tuliskan jawaban dalam bentuk angka saja}**

JAWABAN : 993

Kelompok 1 : 1,2,3,5,6

Kelompok 2 : 7,9

Kelompok 3 : 8,10

Sedangkan 11 – 1000 terbagi pada satu kelompok satu orang sehingga ada 990 kelompok. Total kelompok minimal yang terbentuk : 3 + 990 = 993

1. **Membuat Program Sederhana (Output Only)**

Buatlah program menggunakan bahasa C/C++ untuk membantu perhitungan dari kasus uji yang diberikan. Kemudian masukkan keluaran dari setiap kasus uji sub soal pada kolom tersedia dengan melakukan *copy-paste* dari output program anda!

**Format Masukan**

Baris pertama berisi bilangan bulat N, dan Q.

Q baris berikutnya masing – masing berisikan Ui dan Vi yang menyatakan bahwa Ui mengenal Vi untuk (1 < i < Q)

**Format Keluaran**

Keluarkan satu baris jawaban berupa banyaknya kelompok minimal yang dapat dibentuk.

**Contoh Masukan dan Keluaran**

|  |  |
| --- | --- |
| **Masukan** | **Keluaran** |
| 5 3  1 2  2 3  4 5 | 2 |

Penjelasan Contoh :

Pak Dengklek bisa membentuk 2 kelompok yaitu kelompok 1 : [1,2,3] dan kelompok 2 : [4,5].

**Subsoal 1 (Nilai = 100%)**

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

JAWABAN : 50

Solusi soal secara umum :

|  |
| --- |
| #include <bits/stdc++.h>  using namespace std;  vector <int> adj[1005];  int visited[1005];  void dfs(int x){  visited[x] = 1;  for(auto u : adj[x]){  if(!(visited[u])){  dfs(u);  }  }  }  int main(){  int N,Q;  memset(visited,0,sizeof(visited));  cin>>N>>Q;  int U,V;  while(Q--){  cin>>U>>V;  adj[U].push\_back(V);  adj[V].push\_back(U);  }  int cnt = 0;  for(int i = 1; i<=N; i++){  if(visited[i]) continue;  dfs(i);  cnt++;  }  cout<<cnt<<endl;  return 0;  } |