



ĐÍNH CHÍNH LẠI 1 SỐ BÀI TRONG 40 BỘ ĐỀ THI
(Kèm theo bản gốc “Bài giải 40 bộ đề thi HSG cấp Tỉnh – Môn Tin THCS” đã đăng vào ngày 30/6/2024)

1. Đề số 5 – Bài 3 – Trang 49, 50, 51:

(Điều chỉnh lại tương tự đối với Đề số 39 – Bài 3 – Trang 317, 318)

Bài 3: (5 điểm) Xâu đối xứng

Cho một xâu ST (độ dài không quá 2×10^3 kí tự) chứa ít nhất một chữ cái in hoa ('A'... 'Z') hoặc một chữ cái thường ('a'... 'z') hoặc chữ số ('0'... '9').

Yêu cầu: Hãy tìm độ dài của xâu con đối xứng dài nhất.

Một xâu được gọi là đối xứng nếu nó không có ít hơn một kí tự và nếu đọc từ trái sang phải hay từ phải sang trái đều giống nhau.

Ví dụ: 'B'; 'TOT'; 'BAILIAB' là các xâu đối xứng.

'HOCBAI' là xâu không đối xứng.

♦ **Dữ liệu vào:** Từ tệp văn bản XAUDX.INP gồm một xâu ST.

♦ **Dữ liệu ra:** Ghi ra tệp văn bản XAUDX.OUT gồm một số duy nhất là độ dài của xâu con đối xứng dài nhất.

♦ **Ví dụ:**

XAUDX.INP	XAUDX.OUT
ABFGHIKMLMKIIGH	11

Ràng buộc: Với N là độ dài của xâu ST

♦ **Sub1:** Có 60% test tương ứng 60% số điểm của bài với $N < 255$;

♦ **Sub2:** Có 40% test khác tương ứng 40% số điểm còn lại của bài với $N \leq 2 \times 10^3$.

Đính chính:

Python:

```
fi = open('XAUDX.INP', 'r')
fo = open('XAUDX.OUT', 'w')
ST = fi.read()
n = len(ST)
vtd = 0
maxlen = 1
i = 0
while i < n:
    t = i
    p = i
    while p < n - 1 and ST[p + 1] == ST[p]: p = p + 1
    i = p + 1
    while p < n - 1 and t > 0 and ST[p + 1] == ST[t - 1]:
        t = t - 1
        p = p + 1
```

```

        len = p - t + 1
        if len > maxlen:
            vtd = t
            maxlen = len
fo.write(f'{maxlen}')
fi.close()
fo.close()

```

C++:

```

#include <iostream>
#include <bits/stdc++.h>
#include <vector>
using namespace std;
int main()
{
    string ST;
    freopen("XAUDX.INP", "r", stdin);
    getline(cin, ST);
    int n, vtd, maxlen, len, t, p, i;
    n = ST.length();
    vtd = 0;
    maxlen = 1;
    i = 0;
    while (i < n)
    {
        t = i;
        p = i;
        while (p < n - 1 && ST[p + 1] == ST[p])
        {
            p = p + 1;
        }
        i = p + 1;
        while (p < n - 1 && t > 0 && ST[p + 1] == ST[t - 1])
        {
            t = t - 1;
            p = p + 1;
        }
        len = p - t + 1;
        if (len > maxlen)
        {
            vtd = t;
            maxlen = len;
        }
    }
    freopen("XAUDX.OUT", "w", stdout);
    cout << maxlen;
    return 0;
}

```

2. Đề số 7 – Bài 4 – Trang 70, 71:

Bài 4. (5.0 điểm)

Chuỗi lặp lại là chuỗi mà nếu ta đọc từ trái sang phải có chuỗi ký tự liền sau giống dãy ký tự trước đó; Độ dài chuỗi lặp lại nhỏ nhất là 2 ký tự và là một số chẵn.

Ví dụ: chuỗi 'abab' là chuỗi lặp lại; chuỗi 'abcab' không phải là chuỗi lặp lại.

Viết chương trình nhập theo yêu cầu sau:

Input (dữ liệu nhập) cho trong tập tin BAI04.INP là chuỗi ký tự S có chiều dài không quá 255 ký tự.

Output (dữ liệu xuất) ghi vào tập tin BAI04.OUT là chuỗi con lặp lại dài nhất của S. Biết rằng chuỗi con của S là chuỗi gồm một số ký tự liên tiếp nhau trong S có độ dài nhỏ hơn hoặc bằng độ dài của chuỗi S. Trong trường hợp không tìm được chuỗi lặp lại thì thông báo "KHONG TIM DUOC".

Chương trình ví dụ:

BAI04.INP	BAI04.OUT
zababcabcdq	abcabc

Đính chính:

Python:

```
fi = open('BAI04.INP', 'r')
fo = open('BAI04.OUT', 'w')
S = fi.read()
S = str(S)
chuoicantim = ''
maxlen = 0
for i in range(0, len(S) - 1):
    for j in range(0, len(S)):
        S1 = S[i:(j + 1)]
        S2 = S1 + S1
        if S2 in S:
            if (len(S2) % 2 == 0 and len(S2) > maxlen):
                maxlen = len(S2)
                chuoicantim = S2
            S2 = S2 + S1
if len(chuoicantim) > 1:
    fo.write(f'{chuoicantim}')
else:
    fo.write(f'KHONG TIM THAY')
fi.close()
fo.close()
```

C++:

```
#include <iostream>
#include <bits/stdc++.h>
#include <vector>
#include <string>
using namespace std;
int main()
{
    string S, S1, S2, chuoicantim;
```

```

freopen("BAI04.INP", "r", stdin);
getline(cin, S);
int i, j, maxlen;
chuoicantim = "";
maxlen = 0;
for (i = 0; i < S.length() - 1; i++)
{
    for (j = 0; j < S.length() / 2; j++)
    {
        S1 = S.substr(i, j + 1);
        S2 = S1 + S1;
        while (S.find(S2) != string::npos)
        {
            if ((S2.length() % 2 == 0) && (S2.length() > maxlen))
            {
                maxlen = S2.length();
                chuoicantim = S2;
            }
            S2 = S2 + S1;
        }
    }
}
freopen("BAI04.OUT", "w", stdout);
if (chuoicantim.length() > 1)
    cout << chuoicantim;
else
    cout << "KHONG TIM THAY";
return 0;
}

```

3. Đề số 9 – Bài 2 – Trang 82:

(Điều chỉnh lại tương tự đối với **Đề số 35 – Câu 1 – Trang 280, 281**)

Bài 2: Chữ số tận cùng (5,0 điểm)
 Cho hai số nguyên dương P và Q ($1 < P, Q < 10^9$). Yêu cầu xác định chữ số tận cùng của số P^Q .
 Dữ liệu vào: từ file TANCUNG.INP gồm một dòng chứa 2 số P và Q ($1 < P < Q < 10^9$).
 Dữ liệu ra: ghi ra file TANCUNG.OUT gồm một dòng có một chữ số C ($0 \leq C \leq 9$) thể hiện chữ số tận cùng của số P^Q tương ứng với cặp số (P,Q).

Ví dụ:

TANCUNG.INP	TANCUNG.OUT	Giải thích
13 2	9	$13^2=169$ (tận cùng bằng 9)
26 3	6	$26^3=17576$ (tận cùng bằng 6)
2 10	4	$2^{10}=1024$ (tận cùng bằng 4)

Đính chính:

Python:

```
fi = open('TANCUNG.INP', 'r')
fo = open('TANCUNG.OUT', 'w')
P, Q = fi.read().split()
P, Q = int(P), int(Q)
P = P % 10
if (P == 0) or (P == 1) or (P == 5) or (P == 6): C = P
else:
    Q = Q % 4
    C = pow(P, 4)
    C = (C % 10) * pow(P, Q)
    C = C % 10
fo.write(f'{C}')
fi.close()
fo.close()
```

C++:

```
#include <iostream>
#include <bits/stdc++.h>
#include <vector>
#include <cmath>
using namespace std;
int main()
{
    long long int P, Q, C;
    freopen("TANCUNG.INP", "r", stdin);
    cin >> P >> Q;
    P = P % 10;
    if ((P == 0) || (P == 1) || (P == 5) || (P == 6))
        C = P;
    else
    {
        Q = Q % 4;
        C = pow(P, 4);
        C = (C % 10) * pow(P, Q);
        C = C % 10;
    }
    freopen("TANCUNG.OUT", "w", stdout);
    cout << C;
    return 0;
}
```

4. Đề số 11 – Bài 3 – Trang 98, 99:

Bài 3: Số nguyên tố (7.0 điểm)

Số nguyên tố là số nguyên dương chỉ có duy nhất hai ước là 1 và chính nó. Ví dụ: Số 11 là số nguyên tố vì nó chỉ có hai ước là 1 và 11; số 15 không phải là số nguyên tố vì nó có 4 ước gồm: 1, 3, 5, 15; số 1 không phải là số nguyên tố vì nó có 1 ước là 1.

Yêu cầu: Cho số nguyên N ($1 \leq N \leq 10^6$) và N đoạn số nguyên $[L_i, R_i]$ ($1 \leq L_i < R_i \leq 10^7$; $1 \leq i \leq N$). Hãy tìm số lượng số nguyên tố thuộc mỗi đoạn $[L_i, R_i]$.

Dữ liệu vào: Từ tệp tin văn bản `snt.inp`, gồm:

- Dòng đầu tiên chứa số nguyên N .
- N dòng tiếp theo, dòng thứ i chứa hai số nguyên L_i, R_i (ngăn cách nhau bởi một khoảng trắng).

Dữ liệu ra: Ghi ra tệp tin văn bản `snt.out` gồm N dòng, dòng thứ i ghi một số nguyên là số lượng số nguyên tố thuộc đoạn $[L_i, R_i]$.

Ví dụ:

snt.inp	snt.out	Giải thích
2	0	Số lượng số nguyên tố thuộc 2 đoạn tương ứng:
14 16	5	- Đoạn [14, 16]: không có số nguyên tố;
11 25		- Đoạn [11, 25]: có 5 số nguyên tố là 11, 13, 17, 19, 23.

Ràng buộc:

- > Có 40% số tests tương ứng 40% số điểm của bài với $1 \leq N < 10^3$; $1 \leq L_i < R_i \leq 10^3$.
- > Có 60% số tests tương ứng 60% số điểm của bài với $10^3 \leq N \leq 10^6$; $1 \leq L_i < R_i \leq 10^7$.

Đính chính:

Python:

```
import math
fi = open('snt.inp', 'r')
fo = open('snt.out', 'w')
N = fi.readline()
N = int(N)
L = []
R = []
for i in range(0, N):
    Li, Ri = list(map(int, fi.readline().split()))
    L.append(Li)
    R.append(Ri)
def sangphandoannguyento(L, R):
    gioihantren = int(math.sqrt(R))
    kiemtranguyento = [True] * (gioihantren + 2)
    for k in range(2, gioihantren + 1):
        if kiemtranguyento[k] == True:
            for i in range(k * k, gioihantren + 1, k):
                kiemtranguyento[i] = False
    nguyento = []
    for i in range(2, gioihantren + 1):
        if kiemtranguyento[i] == True:
            nguyento.append(i)
```

```

nguyentotrongdoan = [True] * (R - L + 1)
for i in nguyento:
    for k in range(max(i*i, (L + i - 1) // i * i), R + 1, i):
        nguyentotrongdoan[k - L] = False
if (L == 1):
    nguyentotrongdoan[0] = False
soluong = 0
for i in range(0, len(nguyentotrongdoan)):
    if nguyentotrongdoan[i] == True: soluong = soluong + 1
return soluong
for i in range(0, N):
    soluong = sangphandoanguyento(L[i], R[i])
    fo.write(f'{soluong}\n')
fi.close()
fo.close()

```

C++:

```

#include <iostream>
#include <bits/stdc++.h>
#include <vector>
using namespace std;
void nhap(int L[], int R[], int &N)
{
    freopen("snt.inp", "r", stdin);
    cin >> N;
    for (int i = 0; i < N; i++)
    {
        cin >> L[i];
        cin >> R[i];
    }
}
int sangphandoanguyento(long long int L, long long int R)
{
    long long int gioihantren, k, i, soluong;
    gioihantren = floor(sqrt(R));
    boolkiemtranguyento[gioihantren + 2];
    memset(kiemtranguyento, true, sizeof(kiemtranguyento));
    for (k = 2; k < gioihantren + 1; k++)
    {
        if (kiemtranguyento[k] == true)
        {
            for (i = k*k; i < gioihantren + 1; i = i + k)
                kiemtranguyento[i] = false;
        }
    }
    vector <long long int> nguyento;
    for (i = 2; i < gioihantren + 1; i++)
    {
        if (kiemtranguyento[i] == true)
        {
            nguyento.push_back(i);
        }
    }
}

```

```

bool nguyentotrongdoan[R - L + 1];
memset(nguyentotrongdoan, true, sizeof(nguyentotrongdoan));
for (long long int i : nguyento)
    for (k = max(i*i, (L + i - 1) / i * i); k < R + 1; k = k + i)
        nguyentotrongdoan[k - L] = false;
if (L == 1)
    nguyentotrongdoan[0] = false;
soluong = 0;
for (i = 0; i < R - L + 1; i++)
    if (nguyentotrongdoan[i] == true)
        soluong = soluong + 1;
return soluong;
}
int L[10000001];
int R[10000001];
int N;
int main()
{
    nhap(L, R, N);
    freopen("snt.out", "w", stdout);
    for (int i = 0; i < N; i++)
    {
        cout << sangphandoannguyento(L[i], R[i]) << endl;
    }
    return 0;
}

```

5. Đề số 13 – Bài 1 – Trang 111, 112:

Bài 1 (5,0 điểm).

Linh là học sinh rất yêu thích toán học và thường nghiên cứu các bài toán về tính tổng. Hôm nay, cô giáo cho một bài toán về tính tổng nhưng Linh nghĩ mãi không tìm ra lời giải nên nhờ các bạn học sinh giỏi tin học giúp, bài toán như sau:

Cho số nguyên dương N ($1 \leq N \leq 10^9$).

Yêu cầu: Tính tổng $S(N) = 1 + 2 + 3 + \dots + N$.

Dữ liệu vào: Đọc từ bàn phím theo cấu trúc sau:

- Dòng thứ nhất: Nhập số nguyên dương T là số lượng test ($1 \leq T \leq 10^5$).
- T dòng tiếp theo, mỗi dòng nhập một số nguyên dương N .

Dữ liệu ra: Xuất ra màn hình gồm T dòng, mỗi dòng một số $S(N)$ là tổng tương ứng với test đã cho.

Ví dụ:

Dữ liệu nhập vào	Kết quả xuất ra
2	6
3	15
5	

Giải thích: $T = 2$ nghĩa là có 2 test dữ liệu vào.
 Test 1: $N = 3$ thì $S(3) = 1 + 2 + 3 = 6$.
 Test 2: $N = 5$ thì $S(5) = 1 + 2 + 3 + 4 + 5 = 15$.

Đính chính:

Python:

```
T = int(input())
a = []
i = 0
while i < T:
    N = int(input())
    a.append(N)
    i = i + 1
def tinhtong(n):
    return (n * (n + 1)) // 2
for i in range(0, len(a)): print(tinhtong(a[i]))
```

C++:

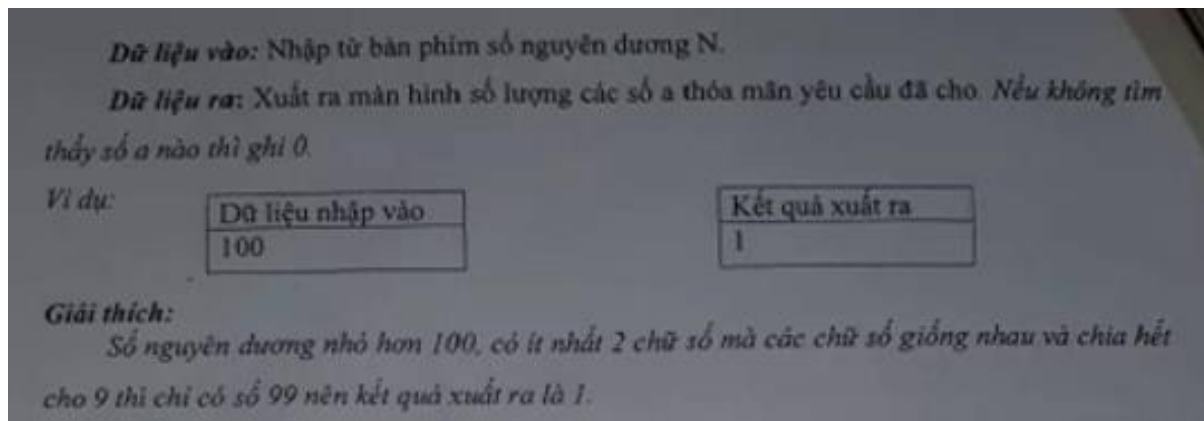
```
#include <iostream>
#include <bits/stdc++.h>
#include <vector>
using namespace std;
int main()
{
    int T, i, N;
    cin >> T;
    vector <long long int> a;
    i = 0;
    while (i < T)
    {
        cin >> N;
        a.push_back(N);
        i = i + 1;
    }
    long long int S;
    for (i = 0; i < a.size(); i++)
    {
        S = (a[i] * (a[i] + 1)) / 2;
        cout << S << endl;
    }
    return 0;
}
```

6. Đề số 14 – Bài 2 – Trang 120, 121, 122:

Bài 2 (5.0 điểm): Lưu bài với tên **BAI2.PAS**

Cho số nguyên dương N ($10 \leq N \leq 10^6$). Hãy đếm số lượng các số nguyên dương a nhỏ hơn N ($10 \leq a \leq N$) thỏa mãn điều kiện: a có ít nhất 2 chữ số, đồng thời a có tất cả các chữ số là giống nhau và a chia hết cho 9.

Yêu cầu: Hãy viết chương trình đếm các số a thỏa mãn yêu cầu đã cho.



Dính chính:

Python:

```
import math
N = int(input())
a = []
so = 0
dodaiN = int(math.log10(N) + 1)
for i in range(0, dodaiN):
    so = so * 10 + 1
    for j in range(1, 10):
        if (10 <= (so * j) and (so * j) <= N):
            a.append(so * j)
soluong = 0
for i in range(0, len(a)):
    if a[i] % 9 == 0: soluong = soluong + 1
print(soluong)
```

C++:

```
#include <iostream>
#include <bits/stdc++.h>
#include <vector>
using namespace std;
int main()
{
    int N, so, dodaiN, i, j, soluong;
    cin >> N;
    vector <int> a;
    so = 0;
    dodaiN = int(log10(N)) + 1;
    for (i = 0; i < dodaiN; i++)
    {
        so = so*10 + 1;
        for (j = 1; j < 10; j++)
            if (10 <= so*j && so*j <= N)
                a.push_back(so*j);
    }
}
```

```

soluong = 0;
for (i = 0; i < a.size(); i++)
    if (a[i] % 9 == 0)
        soluong = soluong + 1;
cout << soluong;
return 0;
}

```

7. Đề số 25 – Câu 4 – Trang 197, 198, 199:

Câu 4: Mua hàng khuyến mãi (4 điểm)	Tên tệp chương trình: CAU4.*																
<p>Nhân dịp năm mới 2020, trung tâm mua sắm HN mở một đợt khuyến mãi lớn. Bình là một khách hàng thân thiết, nên trung tâm gửi cho Bình một danh sách n loại mặt hàng khuyến mãi, được đánh số từ 1 đến n. Mặt hàng thứ i có giá khuyến mãi x_i đồng, giá trị thực là y_i đồng và số lượng là z_i ($1 \leq i \leq n$).</p> <p>Bình có số tiền m đồng để mua các loại mặt hàng khuyến mãi và mong muốn đạt được tổng giá trị thực của các loại mặt hàng có thể mua được là lớn nhất. Vì số lượng hàng khuyến mãi nhiều, nên Bình không biết phải chọn mua những loại mặt hàng nào, số lượng bao nhiêu cho phù hợp với số tiền của mình.</p> <p>Yêu cầu: Hãy giúp Bình mua các loại mặt hàng của trung tâm mua sắm HN, sao cho không vượt quá số tiền m và đạt tổng giá trị thực của các loại mặt hàng có thể mua được là lớn nhất.</p> <p>Dữ liệu: Vào từ tệp văn bản CAU4.INP:</p> <ul style="list-style-type: none"> - Dòng đầu tiên chứa hai số nguyên dương n và m ($1 \leq n \leq 500, 1 < m \leq 5 \times 10^4$); - Dòng thứ i trong n dòng tiếp theo ghi ba số nguyên dương x_i, y_i và z_i lần lượt là giá trị khuyến mãi, giá trị thực và số lượng của loại mặt hàng thứ i ($1 \leq x_i \leq 10^4, 1 \leq y_i \leq 10^4, 1 \leq z_i \leq 100, 1 \leq i \leq n$). <p>Kết quả: Ghi ra tệp văn bản CAU4.OUT:</p> <ul style="list-style-type: none"> - Dòng đầu là tổng giá trị đạt được; - Dòng thứ i trong n dòng tiếp theo ghi số k_i ($1 \leq i \leq n$) là số lượng loại mặt hàng thứ i được chọn mua. Nếu có nhiều cách chọn thỏa mãn, thì đưa ra cách chọn có chỉ số loại mặt hàng nhỏ nhất là nhiều nhất. <p><i>Ví dụ:</i></p>																	
<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th style="padding: 5px;">CAU4.INP</th> <th style="padding: 5px;">CAU4.OUT</th> <th style="padding: 5px;">Giải thích</th> </tr> </thead> <tbody> <tr> <td style="padding: 5px;">5 14</td> <td style="padding: 5px;">28</td> <td rowspan="6" style="padding: 5px; vertical-align: top;">Bình chọn ba loại mặt hàng thứ 4 và một loại mặt hàng thứ 5, tổng giá trị là: 28.</td> </tr> <tr> <td style="padding: 5px;">9 10 1</td> <td style="padding: 5px;">0</td> </tr> <tr> <td style="padding: 5px;">2 3 3</td> <td style="padding: 5px;">0</td> </tr> <tr> <td style="padding: 5px;">2 3 3</td> <td style="padding: 5px;">0</td> </tr> <tr> <td style="padding: 5px;">3 6 4</td> <td style="padding: 5px;">3</td> </tr> <tr> <td style="padding: 5px;">5 10 4</td> <td style="padding: 5px;">1</td> </tr> </tbody> </table>	CAU4.INP	CAU4.OUT	Giải thích	5 14	28	Bình chọn ba loại mặt hàng thứ 4 và một loại mặt hàng thứ 5, tổng giá trị là: 28.	9 10 1	0	2 3 3	0	2 3 3	0	3 6 4	3	5 10 4	1	
CAU4.INP	CAU4.OUT	Giải thích															
5 14	28	Bình chọn ba loại mặt hàng thứ 4 và một loại mặt hàng thứ 5, tổng giá trị là: 28.															
9 10 1	0																
2 3 3	0																
2 3 3	0																
3 6 4	3																
5 10 4	1																

Đính chính:

Python:

```

fi = open('CAU4.INP', 'r')
fo = open('CAU4.OUT', 'w')
n, m = fi.readline().split()
n, m = int(n), int(m)
x = []
y = []
z = []
x.append(0)
y.append(0)
z.append(0)

```

```

for i in range(1, n + 1):
    xi, yi, zi = list(map(int, fi.readline().split()))
    x.append(xi)
    y.append(yi)
    z.append(zi)
dp = [[0] * (m + 1) for i in range(n + 1)]
luu = [[0] * (m + 1) for i in range(n + 1)]
for i in range(1, n + 1):
    for j in range(0, m + 1):
        for u in range(0, min(z[i], j // x[i]) + 1):
            if (dp[i][j] < dp[i - 1][j - (u*x[i])] + u*y[i]):
                dp[i][j] = dp[i - 1][j - (u*x[i])] + u*y[i]
                luu[i][j] = u
fo.write(f'{dp[n][m]}\n')
soluong = []
j = m
for i in range(n, 0, -1):
    soluong.append(luu[i][j])
    j = j - luu[i][j] * x[i]
for i in range(len(soluong) - 1, -1, -1):
fo.write(f'{soluong[i]}\n')
fi.close()
fo.close()

```

C++:

```

#include <iostream>
#include <bits/stdc++.h>
#include <vector>
using namespace std;
int x[505], y[505], z[505], n, m, i, j, u;
int dp[505][100001];
int luu[505][100001], soluong[505];
int main()
{
    freopen("CAU4.INP", "r", stdin);
    cin >> n >> m;
    for (int i = 1; i <= n; i++)
    {
        cin >> x[i];
        cin >> y[i];
        cin >> z[i];
    }
    for (i = 1; i <= n; i++)
    {
        for (j = 0; j <= m; j++)
        {
            for (u = 0; u <= min(z[i], j / x[i]); u++)
            {
                if (dp[i][j] < dp[i - 1][j - (u * x[i])] + u * y[i])
                {
                    dp[i][j] = dp[i - 1][j - (u * x[i])] + u * y[i];
                    luu[i][j] = u;
                }
            }
        }
    }
}

```

```

    }
}
freopen("CAU4.OUT", "w", stdout);
cout << dp[n][m] << endl;
j = m;
for (i = n; i >= 1; i--)
{
    soluong[i] = luu[i][j];
    j = j - luu[i][j] * x[i];
}
for (i = 1; i <= n; i++)
    cout << soluong[i] << endl;
return 0;
}

```

8. Đề số 26 – Câu 2 – Trang 203, 204:

Câu 2: Đua Robot (6 điểm) *Tên tệp chương trình: CAU2.PAS*

Trong cuộc đua tốc độ có n Robot tham gia được đánh số từ 1 đến n . Đường đua có độ dài d (mét). Robot thứ i ($1 \leq i \leq n$) có vận tốc đua không đổi là v_i (mét/phút). Các Robot xuất phát theo thứ tự từ 1 đến n và cách nhau 1 phút. Robot i gọi là vượt Robot j ($1 \leq j \leq n$) nếu i xuất phát sau j và về đích trước j .

Yêu cầu: Xác định số lần vượt nhau của tất cả các Robot trong cuộc đua.

Dữ liệu: Vào từ tệp văn bản CAU2.INP:

- Dòng đầu chứa hai số nguyên dương n và d , $n \leq 10^3$, $d \leq 10^9$;
- Dòng tiếp theo chứa n số nguyên dương v_i , $1 \leq i \leq n$, mỗi số không vượt quá 1000.

Kết quả: Ghi ra tệp văn bản CAU2.OUT số lần vượt nhau của tất cả các Robot trong cuộc đua.

Ví dụ:

CAU2.INP	CAU2.OUT	Giải thích
5 10 1 2 4 3 8	7	Robot 2 vượt Robot 1; Robot 3 vượt các Robot 1, 2; Robot 4 vượt Robot 1; Robot 5 vượt các Robot 1, 2, 4. Tổng số lần vượt là 7.

Đính chính:

Python:

```

fi = open('CAU2.INP', 'r')
fo = open('CAU2.OUT', 'w')
n, d = fi.readline().split()
n, d = int(n), int(d)
v = fi.read().split()
v = list(map(int, v))
t = []
for i in range(0, len(v)):
    t.append(d / v[i] + i)
solan = 0
i = len(t) - 1

```

```

while i > 0:
    for j in range(0, i):
        if t[i] < t[j]: solan = solan + 1
    i = i - 1
fo.write(f'{solan}')
fi.close()
fo.close()

```

C++:

```

#include <iostream>
#include <bits/stdc++.h>
#include <vector>
using namespace std;
void nhap(int v[], int &n, int &d)
{
    freopen("CAU2.INP", "r", stdin);
    cin >> n >> d;
    for (int i = 0; i < n; i++)
        cin >> v[i];
}
int v[1003], n, d;
int main()
{
    nhap(v, n, d);
    vector <float> t;
    int i, solan;
    for (i = 0; i < n; i++)
        t.push_back(float(d) / float(v[i]) + i);
    solan = 0;
    i = t.size() - 1;
    while (i > 0)
    {
        for (int j = 0; j < i; j++)
            if (t[i] < t[j])
                solan = solan + 1;
        i = i - 1;
    }
    freopen("CAU2.OUT", "w", stdout);
    cout << solan;
    return 0;
}

```

9. Đề số 28 – Câu 2 – Trang 220, 221:

Câu 2: Dãy số (6 điểm)

Tên file chương trình: CAU2.PAS

Trong một buổi sinh hoạt câu lạc bộ Tin học của Cung thiếu nhi Hà Nội, thầy giáo ra một bài toán như sau: Từ n số hạng đầu tiên của dãy số Fibonacci (là dãy số có quy luật: số hạng thứ nhất và thứ hai bằng 1, từ số hạng thứ ba trở đi bằng tổng hai số hạng đứng ngay trước nó) thành lập dãy số mới gồm n số bằng cách lần lượt thay mỗi số hạng bằng số dư của số hạng đó khi chia cho 100.

Ví dụ, với $n = 13$ ta có 13 số hạng đầu tiên của dãy số Fibonacci là:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233

Dãy số mới nhận được sau khi thay là:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 44, 33

Để kiểm tra bài làm, thầy giáo yêu cầu các bạn trả lời hai thông tin sau:

1) Số hạng thứ n trong dãy mới là số nào?

2) Có bao nhiêu giá trị khác nhau trong dãy số mới?

Nhập vào Input.inp một số nguyên dương n ($3 \leq n \leq 10000$).

Xuất ra output.inp

+ Dòng 1: in ra số hạng thứ n trong dãy mới.

+ Dòng 2: in ra số lượng các giá trị khác nhau trong dãy mới.

Nhập $n = 13$	
Output	Giải thích
33	Số hạng thứ 13 trong dãy mới là số 33
12	Có 12 giá trị khác nhau trong dãy số mới

Đính chính:

Python:

```
fi = open('Input.inp', 'r')
fo = open('output.inp', 'w')
n = fi.read()
n = int(n)
fibonacci = []
fibonacci.append(0)
fibonacci.append(1)
for i in range(2, n + 1):
    fibonacci.append(fibonacci[i - 1] + fibonacci[i - 2])
    fibonacci[i] = fibonacci[i] % 100
fo.write(f'{fibonacci[n]}\n')
dem = 0
for i in range(1, len(fibonacci)):
    if (fibonacci[i] != fibonacci[i - 1]): dem = dem + 1
fo.write(f'{dem}')
fi.close()
fo.close()
```

C++:

```
#include <iostream>
#include <bits/stdc++.h>
#include <vector>
using namespace std;
int main()
{
    int n, i, dem;
    freopen("Input.inp", "r", stdin);
    cin >> n;
    vector <long long int> fibonacci;
```

```

fibonacci.push_back(0);
fibonacci.push_back(1);
for (i = 2; i < n + 1; i++)
{
    fibonacci.push_back(fibonacci[i - 1] + fibonacci[i - 2]);
    fibonacci[i] = fibonacci[i] % 100;
}
freopen("output.inp", "w", stdout);
cout << fibonacci[n] << endl;
dem = 0;
for (i = 1; i < fibonacci.size(); i++)
    if (fibonacci[i] != fibonacci[i - 1])
        dem = dem + 1;
cout << dem;
return 0;
}

```

10. Đề số 37 – Câu 4 – Trang 301, 302, 303, 304:

Câu 4. ĐƯỜNG THẲNG (2 điểm)

Trên cùng một mặt phẳng tọa độ cho n đường thẳng phân biệt đánh số từ 1 tới n . Đường thẳng i có dạng $y = a_i x + b_i$ ($1 \leq i \leq n$).

Yêu cầu: Đếm số cặp đường thẳng song song trong n đường thẳng trên.

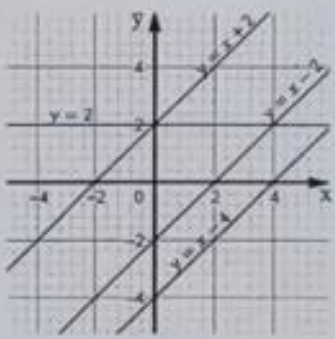
Dữ liệu: Vào từ tệp CAU4.INP gồm:

- Dòng đầu tiên là số nguyên n ($2 \leq n \leq 3 \times 10^6$);
- n dòng sau, mỗi dòng ghi 2 số nguyên a_i, b_i biểu diễn đường thẳng thứ i ($|a_i|, |b_i| \leq 10^9$; $1 \leq i \leq n$).

Kết quả: Ghi ra tệp CAU4.OUT một số nguyên là đáp số của bài.

Ví dụ:

CAU4.INP	CAU4.OUT	CAU4.INP	CAU4.OUT
3	1	3	3
1 2		1 2	
1 -2		1 -2	
0 2		1 -4	



Ràng buộc:

- Có 50% số test ứng với 50% số điểm của bài có $2 \leq n \leq 10^3$; $|a_i|, |b_i| \leq 10^5$;
- Có 25% số test ứng với 25% số điểm của bài có $10^3 < n \leq 10^5$; $|a_i|, |b_i| \leq 10^5$;
- Có 25% số test ứng với 25% số điểm của bài có $10^5 < n \leq 3 \times 10^6$; $|a_i|, |b_i| \leq 10^9$.

Đính chính:

Python:

```

fi = open('CAU4.INP', 'r')
fo = open('CAU4.OUT', 'w')
n = fi.readline()
n = int(n)
a = []
b = []
for i in range(0, n):
    ai, bi = list(map(int, fi.readline().split()))
    a.append(ai)
    b.append(bi)
m = []
for i in range(0, len(a)):
    if a[i] not in m: m.append(a[i])

```

```

mang = []
for i in range(0, len(m)):
    if (m[i] != 0) and (a.count(m[i]) > 1):
        mang.append(a.count(m[i]))
soluong = 0
for i in range(0, len(mang)):
    soluong = soluong + (mang[i] * (mang[i] - 1)) // 2
fo.write(f'{soluong}')
fi.close()
fo.close()

```

C++:

```

#include <iostream>
#include <bits/stdc++.h>
#include <vector>
#include <unordered_map>
using namespace std;
void nhap(int a[], int b[], int &n)
{
    freopen("CAU4.INP", "r", stdin);
    cin >> n;
    for (int i = 0; i < n; i++)
    {
        cin >> a[i];
        cin >> b[i];
    }
}
int a[3000006], b[3000006], n, i;
int main()
{
    nhap(a, b, n);
    long long int soluong;
    unordered_map <int, int> m;
    for (i = 0; i < n; i++)
    {
        m[a[i]] = m[a[i]] + 1;
    }
    vector <int> mang;
    for (auto so : m)
    {
        if ((so.first != 0) && (so.second > 1))
            mang.push_back(so.second);
    }
    freopen("CAU4.OUT", "w", stdout);
    soluong = 0;
    for (i = 0; i < mang.size(); i++)
        soluong = soluong + (mang[i] * (mang[i] - 1)) / 2;
    cout << soluong;
    return 0;
}

```

————— **Tạm dừng** —————