

Análise dos Problemas do Treino de 07/07/17

G – Par ou Ímpar

- **Problema**

Dados os números de dedos que cada jogador jogou, imprimir quem venceu a partida.

- **Solução**

Apenas implementar o que é pedido.

```
for (i = 0; i < n; i++)  
{  
    scanf("%d %d", &a, &b);  
    printf("%s\n", (a+b % 2 == 0) ? nome1 : nome2);  
}
```



B - Tetris

- **Problema**

Dados os jogadores do campeonato e suas 12 pontuações, imprimir a classificação final do campeonato, sendo que a pontuação de cada um é a soma das 12 pontuações menos a menor e a maior.



- **Solução**

Implementar o que é pedido.

Ao ler as 12 pontuações, já podemos realizar sua soma e encontrar a maior e a menor para fazer a subtração no final. Como o limite é 1000, podemos utilizar um método lento de ordenação.

B - Tetris

Tomar cuidado com os empates na classificação: em caso de empate, os jogadores devem ser ordenados em ordem alfabética, mas seu ranking na saída deve ser igual.

```
for (i = 0; i < n; i++)
{
    scanf("%s", nome[i]);
    int maior = -1, menor = 1001, soma = 0, aux;
    for(int j = 0; j < 12; j++)
    {
        scanf("%d", &aux);
        if(aux > maior) maior = aux;
        if(aux < menor) menor = aux;
        soma += aux;
    }
    pontuacao[i] = soma - menor - maior;
}

...

int menor(int a, int b)
{
    if(pontuacao[a] != pontuacao[b])
        return pontuacao[a] < pontuacao[b];
    return strcmp(nome[a], nome[b]) == -1;
}
```

J – TV da Vovó

- **Problema**

Dados os comandos de correção da TV, imprimir a matriz resultante após todos os comandos.

- **Solução**

Para calcular a nova posição dos valores após o deslocamento, realizamos uma soma e tomamos o módulo do índice em relação a N/M. Subtraímos y pois no exercício, y positivo representa deslocamento para cima.

```
for (i = 0; i < n; i++)  
    for(j = 0; j < m; j++)  
        mat2[(i - y + n) % n][(j + x) % m] = mat[i][j];
```



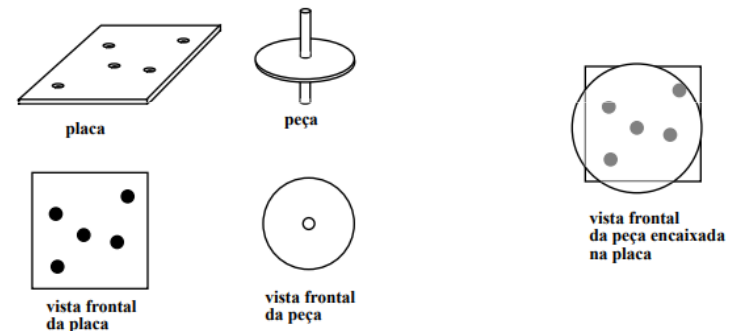
J – TV da Vovó

- O tamanho do matriz pode ir até 1000x1000, e pode haver até 1000 comandos de correção. Logo atualizar a matriz após todo comando é muito lento;
- Os comandos podem ser acumulados. Fazer uma translação +1 em X e depois fazer outra translação +1 em X é a mesma coisa que fazer uma translação +2 em X.
- Assim, mantemos duas variáveis indicando o deslocamento horizontal e vertical, que são atualizados conforme lemos o comando.
- Ao final de todos os comandos, atualizamos a matriz uma vez só e imprimimos seus valores.

H – Cubra os Furos

- **Problema**

Dado uma placa com furos, encontrar o menor diâmetro tal que, se encaixarmos a peça circular em um dos furos, todos os outros furos estão cobertos.



- **Solução**

- Se fixarmos um furo como o centro do círculo, como calcularíamos o menor diâmetro que cobriria todos os outros furos?
- Calculando a distância do centro a todos os demais pontos: a maior distância será o raio mínimo necessário para cobrir todos os pontos.

H – Cubra os Furos

- Ou seja, tendo o centro do círculo fixado, precisamos de N operações para descobrir o menor diâmetro para aquele centro.
- O número de pontos vai até 1000, logo é rápido suficiente testar todos os centros possíveis (todos os pontos) = N^2 operações.
- Para cada ponto pegamos a maior distância daquele ponto até outro ponto; pegamos a menor dessas maiores distâncias.

```
double resp = INF;

for (i = 0; i < n; i++)
{
    double maior = 0;
    for(j = 0; j < n; j++)
    {
        double d = dist(x[i], y[i], x[j], y[j]);
        if(d > maior)
            maior = d;
    }
    if(2 * maior < resp)
        resp = 2 * maior;
}
```


C – Festa Junina

- **Problema**

Dadas as relações de amizade entre os alunos, encontrar o maior grupo tal que todos os alunos no grupo são amigos.

- **Solução**

Este problema pode ser visto como um problema de grafos (maior clique de um grafo). Mas encontrar o maior clique de um grafo é um problema NP-completo.

A observação chave para resolver o problema está no limite da entrada: $N \leq 20$

C – Festa Junina

- Quantas comissões podemos fazer com N alunos?
- Cada aluno pode estar dentro ou fora da comissão = 2^n comissões diferentes
- Como $N \leq 20$, o maior número de comissões é $2^{20} = 1.048.576$ (ordem 10^6)
- Então é viável uma solução por força bruta: geramos todas as comissões possíveis. Testamos se essa comissão é viável (podemos armazenar as relações em uma matriz: $\text{mat}[a][b] = 1$ se a e b são amigos, $\text{mat}[a][b] = 0$ caso contrário). Armazenamos a maior das comissões viáveis.
- Para gerar todas as comissões: algoritmo recursivo ou truque com bits (Galante passa binário em ICC por favor).

Demais problemas

- A serem discutidos futuramente:
Número de Erdos, Manutenção – Grafos
Parque Jurássico, Palíndrome – Programação Dinâmica
Elástico – Geometria Computacional (Convex Hull)