

# Maratona de Programação



O que é?



# Regras de participação - 2015

```
if mundiais >= 2 or regionais >= 5:  
    print('Não elegível')  
elif entrou_universidade <= 2010 and nascimento <= 1991:  
    print('Não elegível')  
else:  
    print('Elegível')
```

<http://maratona.ime.usp.br/regras15.html>

# Fase Brasileira - 2015

- Primeira fase (12 de setembro)
  - Acontece em várias regiões do Brasil
  - Melhores times de cada sede se classificam
- Segunda fase ou Final brasileira (13 e 14 de novembro)
  - Melhores times se classificam para o mundial
  - Normalmente vão 6 times brasileiros para o mundial
  - Em 2015 será na USP

# Competição

- Cada time é composto por 3 pessoas + 1 reserva
- Cada time terá acesso a 1 computador
- Lista de problemas
- Ao resolver um problema, ganha-se um balão
- Só pode levar material impresso (exemplo: livros)
- Os problemas são submetidos através de um sistema
- O sistema indicará se acertou ou não o problema
- Mais informações: <http://maratona.ime.usp.br>

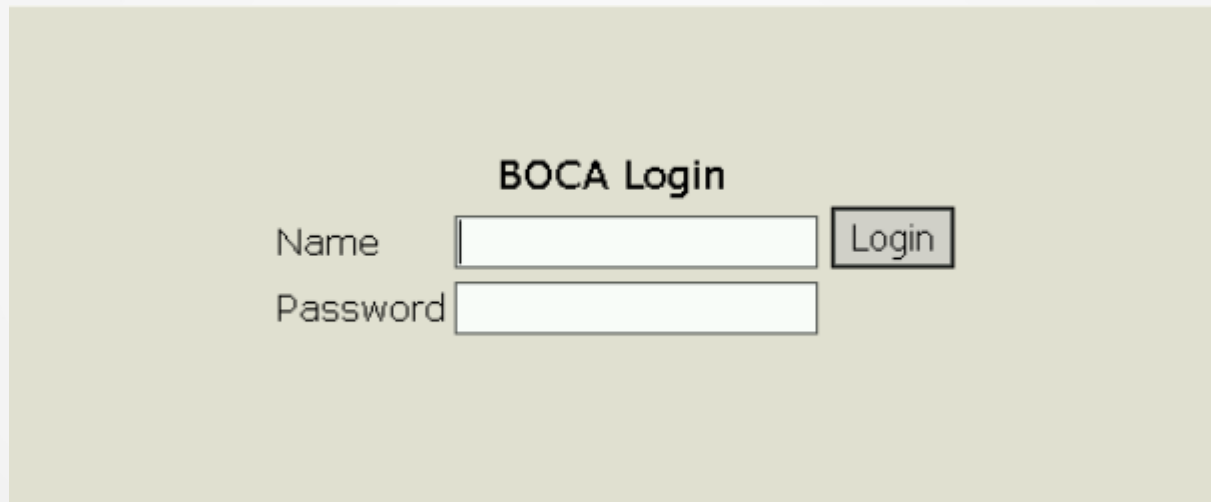


# Motivação

- Aplicar os conhecimentos
- Desenvolver novas técnicas/soluções
- Eficiência na resolução dos problemas
- Trabalho em equipe
- Conhecer pessoas (networking)
- “A maratona de programação diferencia meros programadores de verdadeiros hackers.”

# BOCA

- Sistema utilizado nas competições de programação da SBC (Sociedade Brasileira de Computação)

A screenshot of the BOCA Login interface. It features a light beige background with the title "BOCA Login" centered at the top. Below the title, there are two input fields: "Name" and "Password". The "Name" field is a white rectangle with a thin black border. To its right is a grey button with the text "Login" in white. Below the "Name" field is the "Password" label, followed by another white input field with a thin black border.

BOCA Login

Name

Password

# BOCA – envio de soluções

BOCA Username: Universidade A (site=2) 157 minute(s) left

Runs Score Clarifications Tasks Options Logout

Run #	Time	Problem	Language	Answer
2	37	Problema 1	C	No - Compilation error
3	37	Problema 3	Java	Yes
4	83	Problema 1	C++	Not answered yet

To submit a program, just fill in the following fields:

Problem: Problema 1

Language: C

Source code:  Browse...

Send Clear



# BOCA – perguntas aos juízes

Use the following fields to answer the clarification:

Clarification Site: 2  
Clarification Number: 1  
Clarification Time: 86  
Problem: Problema 1

Clarification:

Answer:

Answer all users in the site ☐  
Answer all users in all sites ☐

Answer No response Cancel Clear

# BOCA - placar

BOCA		Username: Placar (site=2)				117 minute(s) left	
Score				Logout			
#	Username	User Site	User	Solved	Problem Details	Total Time	
1	team2	2	Universidade B	1	Problema 1	98	
2	team1	2	Universidade A	0		0	

Mais informações: <http://www.ime.usp.br/~cassio/boca/>

# Tempo

- O sistema faz a contabilização do tempo
- Velocidade ao resolver os problemas
- Se empatar na quantidade de problemas resolvidos?
  - Analisa-se o tempo (critério de desempate)
- A cada submissão incorreta de um problema é atribuída uma penalidade de tempo.
- O time que conseguir resolver o maior número de problemas (no menor tempo com as penalidades) será declarado o vencedor.

# Tempo - penalidade

- Acumula-se 20 minutos por submissão incorreta
- Exemplo de ranking:
  - problemas resolvidos (tempo)

Total
10 (1010)
8 (1025)
7 (638)
7 (667)
7 (768)
7 (802)
7 (810)

# Treinamento

- Competições remotas
- Sites disponíveis para o treinamento:
  - <https://www.urionlinejudge.com.br>
  - <https://uva.onlinejudge.org>
  - <https://icpcarchive.ecs.baylor.edu>
  - <http://www.spoj.com>

# Notebook

- Importante um bom material impresso
- Algoritmos implementados
- Fórmulas
- Algoritmos com detalhes “chatos” de implementação
- Não basta apenas copiar, mas sim entender!
- Exemplos de notebook:
  - <http://bit.ly/1dLOHXc>
  - <http://bit.ly/1N5gwpC>



# Linguagens de programação

- Linguagens que podem ser utilizadas:
  - C
  - C++
  - Java
- Nos sites como por exemplo o URI:
  - C, C++, Java, Python 2, Python 3
- Curso de C++ (> 70 vídeo-aulas):
  - <http://bit.ly/1FpmD1M>

# Entrada e saída

- Cada problema tem suas entradas e saídas respectivas
- As entradas e saídas possuem um formato
- Exemplo:
  - Entrada: dois números inteiros.
  - Saída: imprima SOMA com um espaço em branco antes e depois do sinal de igual.

Samples Input	Samples Output
30 10	SOMA = 40

# Tipos de problemas

- Ad-hoc
- Manipulação de strings
- Estruturas de dados
- Matemática (teoria dos números, geometria ...)
- Paradigmas (programação dinâmica, gulosos ...)
- Grafos
- (...)

# Problema 1

- Acessem:
  - URI – Problema 1037
  - Menu Search (Buscar) e digite 1037
  - Solução?

# Problema 1 - Solução

```
float n;

scanf("%f", &n);
if(n >= 0 && n <= 25)
    printf("Intervalo [0,25]\n");
else if(n > 25 && n <= 50)
    printf("Intervalo (25,50]\n");
else if(n > 50 && n <= 75)
    printf("Intervalo (50,75]\n");
else if(n > 75 && n <= 100)
    printf("Intervalo (75,100]\n");
else
    printf("Fora de intervalo\n");
```

# Códigos


- Os códigos dos problemas estarão no Github:
  - [https://github.com/marcoscastro/maratona\\_unifesp/](https://github.com/marcoscastro/maratona_unifesp/)
- Compartilhe suas soluções:
  - Basta fazer um fork do projeto (cópia)
  - Adicione a sua solução ao projeto (commits e push's)
  - Adicione a sua solução ao repositório original através de um pull request
  - Cada problema terá uma pasta
  - Nome: código + seu nome. Ex.: 1037\_marcoscastro.cpp







# Códigos


- Dentro da pasta de cada problema poderemos adicionar referências, algoritmos que ajudem a resolver etc.

solucao para o problema do URI 1037

 marcos authored 7 minutes ago

latest commit 2e08cfb015 

 URI-1037	solucao para o problema do URI 1037	6 minutes ago
 .gitignore	solucao para o problema do URI 1037	6 minutes ago
 README.md	Create README.md	15 minutes ago

 README.md

## Treinamento para Maratona de Programação 2015 - Unifesp

---

Códigos dos problemas resolvidos no treinamento para a maratona de programação 2015.

## Problema 2

- Acessem:
  - URI – Problema 1564
  - Menu Search (Buscar) e digite 1564
  - Solução?

## Problema 2 - Discussão

- O problema termina com EOF (end of file)
- Para tratar isso em C/C++ basta fazer:
  - `while(scanf("entradas") != EOF) { ... }`

## Problema 2 - Solução

```
int N;  
  
while(scanf("%d", &N) != EOF)  
{  
    if(N == 0)  
        printf("vai ter copa!\n");  
    else  
        printf("vai ter duas!\n");  
}
```

## Problema 2 - Execução

- Salve a entrada em um arquivo
- Execute no terminal (cmd):
  - nome\_programa < entrada

```
marcos@marcos-not:~$ ./1564_marcoscastro < entrada
vai ter copa!
vai ter duas!
vai ter copa!
vai ter duas!
vai ter duas!
vai ter copa!
```

# Problema 3

- Acessem:
  - URI – Problema 1557
  - Menu Search (Buscar) e digite 1557
  - Solução?
  - **Atenção** especial para o formato da saída!



## Problema 3 - Discussão

- Problema de fácil entendimento
- O que pode complicar? O formato da saída
- Os valores da matriz devem está formatados em um campo de tamanho T onde T é igual ao número de dígitos do maior número da matriz
- Função que retorna a quantidade de dígitos
  - enquanto  $N \neq 0$ :
    - divida N por 10
    - incremente a quantidade de dígitos

## Problema 3 - Discussão

- Retorna a quantidade de dígitos

```
int obter_num_digitos(int N)
{
    int q_digitos = 0;

    while(N)
    {
        N /= 10;
        q_digitos++;
    }
    return q_digitos;
}
```

## Problema 3 - Discussão

- Mas como saber qual número é o maior?
- O maior número da matriz é o último da diagonal principal
- A diagonal principal é uma PG de razão 4
- Como calcular o último termo de uma PG?
  - $a_n = a_1 * q^{(n - 1)}$
  - “^” é o símbolo da exponenciação
  - Como  $a_1$  é sempre 1, então:
    - $a_n = q^{(n - 1)}$

## Problema 3 - Discussão

- Como calcular a exponenciação em C/C++ ?
- Fácil e rápido:
  - `int maior = pow(4, N - 1);`
  - “pow” é uma função da biblioteca `math.h`
- Para obter o número de dígitos do “maior”:
  - `int T = obter_num_digitos(maior);`

## Problema 3 - Discussão

- Formatar em um campo de tamanho T:
  - `printf("%*d", T, elemento);`
  - “T” é a quantidade de dígitos do “maior”
  - “elemento” é o elemento da matriz a ser impresso
  - Resolve?
    - Presentation error!

## Problema 3 - Discussão

- Perceba que o problema diz:
  - Os valores, além de estarem em um campo de tamanho T, precisam estar separados por **um espaço**.
  - `printf(" %*d", T, elemento);`
  - E agora?
    - Presentation error!



## Problema 3 - Discussão

- O espaço só é para ser colocado **entre** os valores.
- Para o primeiro número de cada linha:
  - `printf("%*d", T, elemento);`
- Para os restantes:
  - `printf(" %*d", T, elemento);`
- Não esqueça de deixar uma linha em branco entre cada impressão de matriz.

## Problema 3 - Discussão

- O final da entrada é indicada pelo 0 (zero)
  - Se o usuário entrar com 0, sai do loop (break).
- Dica para imprimir os elementos da matriz:
  - As linhas/colunas são uma PG de razão 2

# Problema 4

- Acessem:
  - URI – Problema 1171
  - Menu Search (Buscar) e digite 1171
  - Solução?

## Problema 4 – Discussão

- Máximo de elementos: 2000 (dois mil)
- Declaração de um vetor de inteiros:
  - `int v[2000];`
- Inicialização de todos os elementos com 0:
  - `memset(v, 0, sizeof(v));`
  - “memset” é do cabeçalho `string.h`

## Problema 4 - Discussão

- Obtendo os dados:
  - `scanf("%d", &X);`
  - `v[X - 1]++;` // incrementa a quantidade
- Após obter os dados:
  - Basta imprimir se `v[i] != 0`
  - A ordem de impressão é a ordem crescente
- **Accepted!**
  - Tempo: 0.004 (pode melhorar?)

# Problema 5

- Acessem:
  - URI – Problema 1397
  - Menu Search (Buscar) e digite 1397
  - Solução?

## Problema 5 - Discussão

- “N” é o número de partidas (rounds)
- 2 jogadores
- Cada jogador diz um número de 0 a 10
- Acumula as vitórias do jogador 1 e 2
- Em caso de empate, **não** acumula
- Ao final, imprime as pontuações
- O final da entrada é indicado por  $N = 0$

## Problema 5 - Solução

```
while(1)
{
    int N, A, somaA = 0, B, somaB = 0;

    scanf("%d", &N);
    if(N == 0) break;
    for(int i = 0; i < N; i++)
    {
        scanf("%d %d", &A, &B);
        if(A > B)
            somaA++;
        else if(A < B)
            somaB++;
    }
    printf("%d %d\n", somaA, somaB);
}
```



# Dúvidas?

