

---

# MC102: Algoritmos e Programação de Computadores

PRIMEIRO SEMESTRE DE 2012

PROF. LUIZ E. BUZATO  
BUZATO@IC.UNICAMP.BR  
TURMAS S E T

**Páginas da Disciplina:** “<http://www.ic.unicamp.br/~buzato/teaching/2012/mc102/mc102.html>” é a página da disciplina na teia. Visite-a sempre. A partir desta página você terá acesso a informações sobre a disciplina, inclusive **avisos**, listas de exercícios, laboratórios, notas, e todo o resto do material de apoio. Mensagens de e-mail, enviadas ao email oficial da UNICAMP, serão também usadas como meio de comunicação alternativo. Se você não está recebendo as mensagens desta disciplina mas seus colegas estão, verifique com o professor ou o monitor se o seu endereço de e-mail está correto. A visita a esta página é parte integrante de suas tarefas nessa disciplina. Não serão aceitas reclamações se você for prejudicado, em qualquer aspecto, caso não tenha visitado a página frequentemente.

**Horários:** O horário de aulas é:

Turma	AULAS		
	Dia	Sala	Hora
ST	3 <sup>a</sup>	CB-14	10:05–11:45
ST	5 <sup>a</sup>	CB-13	10:05–11:45
LABS			
S	2 <sup>a</sup>	CC-02, CC-03	16:00–18:00
T	2 <sup>a</sup>	CC-02, CC-03	16:00–18:00

**Atendimentos:** Durante os laboratórios.

**Laboratórios:** Os exercícios práticos de laboratório, abreviadamente *labs*, farão parte integrante da avaliação semestral, como especificado abaixo. Inclusive, questões semelhantes aos exercícios de lab poderão ser exigidos em provas. **Não haverá reposição de laboratórios, nem serão aceitos laboratórios em atraso.**

Contaremos com um ambiente computadorizado que fará as correções e verificações de soluções dos labs de forma inteiramente automática. A submissão e correção das soluções deverá ser feita sempre via Internet. Cada lab receberá uma nota, no intervalo  $[0, 8]$ , e proporcional ao número de testes certificados com sucesso pelo seu programa quando for submetido ao corretor automático. Valerá sempre a última submissão que for feita. Até três pontos adicionais poderão ser atribuídos a cada lab, dependendo da *qualidade e clareza* do código apresentado, bem como da qualidade dos algoritmos entregues junto a cada lab.

Note que:

- Dia 05.03 o lab não pontuará para nota. Nesse dia os monitores estarão no lab, no horário normal, e farão um exercício com todos os presentes. Esse exercício visa familiarizar os alunos com tarefas simples imediatas: *login/logout*, uso de um editor de textos, compilação local, como usar o corretor automático SuSy, como proteger suas áreas de trabalho, como acessar o servidor para obter material de apoio da disciplina, ... *Recomendamos fortemente sua participação.*

Além disso, os monitores repassarão vários outros avisos importantes e será uma oportunidade para os alunos se familiarizarem com a rotina de preparo e submissão de labs.

Os exercícios de lab serão **sempre individuais** e a tarefa de cada lab deve ser completada estritamente no tempo estipulado para cada lab.

O ambiente para desenvolvimento dos labs pode ser tanto o Windows como o Linux. Porém, note que o programa corretor de labs (Susy) funciona sob Linux, e algumas incompatibilidades entre Windows e Linux, ou mesmo entre versões diferentes do Linux, ou mesmo versões diferentes dos compiladores, poderão criar algumas dificuldades. Alunos da Unicamp podem obter gratuitamente o ambiente Windows para instalação em suas máquinas pessoais, desde que dedicados apenas à tarefas acadêmicas. Também podem obter gratuitamente várias ferramentas da MS para ambientes Windows, tais como o VISUAL C++ e a plataforma .NET. Consulte sua unidade para saber como obter esses softwares, se tiver necessidade deles. Ambientes de programação C e C++ para Linux, podem ser obtidos gratuitamente pela Internet. Veja mais detalhes na página da disciplina.

**Qualquer tentativa de fraude em qualquer um dos labs implicará em nota 0.0 (zero) no lab em questão para TODOS os implicados. Reincidência implicará em nota 0.0 (zero) na média de labs para TODOS os implicados. O sistema de correção automático dispõe de ferramentas sofisticadas para detectar fraudes (“colas”) entre todos os labs submetidos ao longo do semestre, e estas ferramentas estarão sempre ativas. Detectada a fraude, os alunos envolvidos serão comunicados e a punição específica será imediatamente aplicada.**

**Provas e Exame:** Serão aplicadas duas provas teóricas, em classe. Dependendo do aproveitamento do aluno no semestre, este deverá prestar um exame final. As provas teóricas e o exame final serão realizados no local e horário das aulas teóricas. Veja a tabela a seguir.

Teoria	Dia	Teoria	Dia	Exame	Dia
$PT_1$	19/04	$PT_2$	19/06	$E$	10/07

onde  $PT_i$  indica a  $i$ -ésima prova teórica,  $E$  indica o exame. **Não** será possível: (i) realizar novas provas ou provas substitutivas; e (ii) trocar o horário das provas e/ou do exame final.

Qualquer tentativa de fraude nas provas teóricas ou no exame implicará em **nota 0.0 (zero)** na **prova ou exame em questão** e para **todos** os implicados.

**Avaliação:** As avaliações serão feitas de modo independente para cada uma das partes da disciplina.

**Labs:** A média dos exercícios de lab *realizados em sala* será computada pela fórmula  $EL = (\sum_{i=1}^n l_i)/n$ , onde  $n$  é o número de labs realizados, e  $l_i$  indica a nota do lab  $i$ ,  $i = 1, \dots, n$ .

A média dos exercícios de lab complementares será computada pela fórmula  $ELC = (\sum_j^m lc_j)/m$ , onde  $m$  é o número de exercícios complementares realizados, e  $lc_j$  indica a nota do lab complementar  $j$ ,  $j = 1, \dots, m$ .

A média de exercícios de laboratório será calculada por  $ML = (7EL + 3ELC)/10$ .

**Teoria:** A média das provas de teoria será computada por  $MT = (2PT_1 + 3PT_2)/5$ .

**No semestre:** A média semestral será calculada por

1.  $MS = (6MT + 4ML)/10$ , **desde que**  $MT \geq 3.5$  e  $ML \geq 3.5$ .
2. Caso contrário, a média semestral será calculada por  $\min\{4.9, MS\}$ , portanto, colocando o aluno automaticamente em exame.

Quem não conseguir  $MS \geq 5.0$  está obrigado a prestar exame final.

**Média final:** A média final será computada como:

$$MF = \begin{cases} MS & \text{se } MS \geq 5.0 \\ (MS + E)/2 & \text{caso contrário,} \end{cases}$$

onde  $E$  é a nota do exame.

**Aprovação:** Terão SE APROVADO na disciplina aqueles alunos que conseguirem  $MF \geq 5.0$ . Terão SE REPROVADO na disciplina aqueles alunos cuja média final satisfaça  $MF < 5.0$ .

**Programa da disciplina:** Os tópicos programados para o semestre são:

### **Algoritmos**

1. Noção de algoritmo
2. Algoritmos e linguagens de programação
3. Noção de correção de algoritmos
4. Noção de algoritmos
5. Noção de problemas decidíveis e não decidíveis
6. Noção de problemas tratáveis e intratáveis

### **Linguagem C**

1. Variáveis simples (tipos inteiros e reais); comando de atribuição
2. Comandos de entrada e de saída
3. tipo char; constantes
4. Variáveis booleanas; comandos de desvio
5. Comandos repetitivos
6. Vetores e matrizes
7. Registros; tipos enumerados; cadeias
8. Procedimentos e funções
9. Recursão
10. Alocação dinâmica de memória
11. Arquivos de texto
12. Arquivos e comandos de sistema

Se houver disponibilidade de tempo, discutiremos também noções de orientação a objetos.

**Referências:** Existem muitos textos sobre algoritmos e a linguagem C, especialmente textos escritos em inglês. Não vamos seguir um livro texto em particular. A sugestão é que cada aluno se dirija à biblioteca, examine os livros que lá encontre e use aquele ao qual melhor se adapte, inclusive outros textos que não estão incluídos na lista a seguir.

Outra sugestão: use um mecanismo de busca, como por exemplo <http://www.google.com>, e procure textos e tutoriais acerca algoritmos e da linguagem C. Com certeza, você vai encontrar muitos destes textos, tanto em inglês como em português. Escolha aqueles que mais se adaptam ao seu estilo de estudar.

Veja a seção de material didático, na página da disciplina. Lá você vai encontrar uma explicação de como usar o servidor de SVN para acessar todo o material didático disponível. Também vai encontrar *links* a partir dos quais você poderá obter o *software* necessário para usar o servidor SVN, para várias plataformas. O material didático disponível cobre tanto a parte inicial, de noções sobre algoritmos, quanto a segunda parte, versando sobre como usar a linguagem C para programar algoritmos. Para cada uma das partes está disponível o conjunto de transparências que será usado em classe, durante as aulas, bem como uma apostila, em português, com explicações bem mais detalhadas sobre o conteúdo das transparências. Todos esses arquivos estão no formato PDF. Para ilustrar a parte de programação em C, há também disponível um conjunto de programas em C. Alguns deles serão discutidos em classe, outros ficarão como suporte para estudo. Note que versão atual de cada programa pode conter erros propositais, visto que o material se destina à discussão em aula e, portanto, o formato final de cada programa no servidor SVN reflete o seu estado ao final da última discussão em que o programa foi usado. Todos os programas foram elaborados no ambiente DeVC++, que é gratuito, embora os arquivos fonte principais possam ser compilados em outros ambientes, talvez com pequenos ajustes. Veja o atalho para *softwares* na página da disciplina para mais informação sobre como obter o DeVC++.

Outras referências:

1. A. Kelley, I. Pohl *A Book on C*, Addison Wesley, 1998.
2. H. M. Deitel, P. J. Deitel *C: How to Program*, Prentice Hall, 2004.

3. B. W. Kernighan *Programming in C: A Tutorial*.  
Veja também em <http://www.lysator.liu.se/c/bwk-tutor.html>
4. D. Harel *Algorithmics*, Addison Wesley-2<sup>a</sup>. edição, 1993.
5. D. Harel *Computers Ltd.: what they really can't do*, Oxford Univ. Press, 2000.