
MC102: Algoritmos e Programação de Computadores

PRIMEIRO SEMESTRE DE 2016

PROF. ARNALDO MOURA
ARNALDO@IC.UNICAMP.BR
TURMAS QRST

PROF. LEHILTON L. C. PEDROSA
LEHILTON@IC.UNICAMP.BR
TURMAS WY

Páginas da Disciplina: “<http://ze.ic.unicamp.br/WWW/Disciplinas/MC102>” (para as turmas QRST) e “<http://www.ic.unicamp.br/~lehilton/mc102wy>” (para as turmas WY) são as páginas da disciplina na teia. Visite-a sempre. A partir destas páginas você terá acesso a informações sobre a disciplina, inclusive **avisos**, listas de exercícios, laboratórios, notas e todo o resto do material de apoio.

Horários: O horário de aulas é:

Turma	AULAS		
	Dia	Sala	Hora
QRST	3 ^a		10:00–12:00
QRST	5 ^a		10:00–12:00
	LABS		
QR	2 ^a	CC-02, CC-03	14:00–16:00
ST	2 ^a	CC-02, CC-03	16:00–18:00

Turma	AULAS		
	Dia	Sala	Hora
WY	3 ^a		21:00–23:00
WY	5 ^a		19:00–21:00
	LABS		
WY	4 ^a	CC-02, CC-03	19:00–21:00

Atendimentos: Haverá atendimento de monitoria nos dias e horários a seguir:

Turma	MONITORIA		
	Dia	Sala	Hora
QRSTWY	3 ^a a 5 ^a		13:00–14:00
QRSTWY	3 ^a a 5 ^a		18:00–19:00

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

Correção: Contaremos com um ambiente computadorizado que fará correções e verificações de soluções de forma automática. A submissão e correção das soluções deverá ser feita via Internet. Somente valerá a última submissão de cada lab. Cada lab terá nota no intervalo $[0, 10]$, composta por:

- uma parte com valor no intervalo $[0, 6]$ e proporcional ao número de testes certificados com sucesso pelo seu programa quando for submetido ao corretor automático;
- e uma segunda parte com valor no intervalo $[0, 4]$, que será atribuído dependendo da *qualidade, clareza e correção* do código apresentado, bem como de critérios específicos de cada lab.

Entrega: Haverá dois tipos de lab:

LABS SEMANAIS: Nesse caso, o tempo para completar o lab será de uma semana. A tarefa estará disponível na Internet a partir das 8h das segundas-feiras, devendo ser completada até o domingo seguinte, às 22h. O prazo já contempla eventuais falhas do sistema de correção, problemas de rede etc. A tarefa é **individual**. Nas semanas em que houver lab semanal, as aulas de lab serão de prática e orientação, com auxílio dos monitores. Durante as sessões de lab e atendimento, os monitores estarão à disposição para tirar dúvidas sobre o material visto em classe e sobre as tarefas propostas para a semana em curso. No entanto, os monitores *não vão* fazer as tarefas de labs junto com os alunos. Eles estão lá para orientar e tirar dúvidas. As tarefas podem ser desenvolvidas em casa, em qualquer horário, respeitados os dias/horas de início e término. Os labs semanais podem ser submetidos a partir de qualquer máquina com acesso à Internet.

LABS EM SALA: Em semanas específicas, a serem anunciadas, o exercício de lab deverá ser totalmente desenvolvido em sala, durante o período designado para a aula de lab, de forma **individual** sem qualquer tipo de consulta. As soluções, também nesses casos, serão submetidas pela Internet, mas apenas a partir das máquinas fisicamente presentes no laboratório.

Testes pós-laboratórios: Serão aplicados minitestes cujos conteúdos são os mesmos dos labs realizados e com duração de até 30 minutos durante a aula teórica na semana imediatamente posterior ao lab, isto é, nas terças-feiras para as turmas QRST e nas quintas-feiras para as turmas WY. Cada miniteste poderá valer até 1,5 pontos-extra que serão somados à nota do lab correspondente. Observe que a nota de cada lab será no máximo 10.

Observações:

- Em dia e em sala a serem informados pela Seção de Informática do IC, haverá uma palestra para alunos iniciantes dada pela Comissão de Informática do IC. Serão passadas informações importantes. Aproveite para obter seu *login* e sua senha para usar as máquinas do Instituto. Sem *login* e senha, *você não vai poder participar do primeiro lab*.
- Dia 07/03 (turmas QRST) e 09/03 (turmas WY) o lab não pontuará para nota. Nesse dia os monitores estarão no laboratório 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 etc. *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.

O ambiente para desenvolvimento dos labs pode ser tanto o Windows como o Linux. Note que o programa corretor de labs (Susy) funciona sob Linux e que algumas incompatibilidades entre Windows e Linux, entre versões diferentes do Linux, ou mesmo entre 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 a tarefas acadêmicas. Também podem obter gratuitamente várias ferramentas da MS para ambientes Windows, tais como o VISUAL C++ e a plataforma .NET. Se tiver necessidade deles, consulte sua unidade para saber como obter esses softwares. 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 essas 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	26/04	PT_2	21/06	E	12/07

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

Labs semanais: A média dos *lab semanais* será dada pela fórmula $ECL = (\sum_{j=1}^m cl_j)/m$, onde m é o número de labs semanais e cl_j indica a nota do lab semanal j , para $j = 1, \dots, m$.

Teoria: A média das provas de teoria será computada por $MT = (P_1 + 2P_2)/3$.

No semestre: A média semestral será calculada por

1. $MS = (3MT + 2EL + 2ECL)/7$, **desde que** $MT \geq 4$, $EL \geq 4$ e $ECL \geq 4$.
2. Caso contrário, a média semestral será calculada por $\min\{4,9 ; MS\}$, portanto, colocando o aluno automaticamente em exame.

Se $2,5 \leq MS < 5,0$ o aluno deverá prestar um exame final, em classe.

Média final: A média final será computada como:

$$MF = \begin{cases} MS & \text{se } MS \geq 5,0 \text{ ou se } MS < 2,5 \\ (MS + E)/2 & \text{caso contrário,} \end{cases}$$

onde E é a nota do exame.

Aprovação: Terá SE APROVADO o aluno que conseguir média final $MF \geq 5,0$. Terá SE REPROVADO o aluno cuja média final satisfaça $MF < 5,0$.

Programa da disciplina: Os tópicos que serão discutidos no decorrer do 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

Havendo 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 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 êrros 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^a. edição, 1993.
5. D. Harel *Computers Ltd.: what they really can't do*, Oxford Univ. Press, 2000.