



Piscina C

Rush 02

Sumário: Este documento é o enunciado do módulo C 02 da Piscina C da 42.

Versão: 11

Conteúdo

I	Preâmbulo	2
II	Instruções	3
III	Instruções IA	5
IV	Enunciado	8
V	Bónus	10
VI	Submissão e Avaliação	11

Capítulo I

Preâmbulo

A seguir, uma receita do bolo Quatre-Quart:

4 ingredientes (rende um bolo):

- 4 ovos
- 200g de açúcar
- 200g de farinha
- 200g de manteiga sem sal

Modo de preparo:

- Preaqueça o forno a 180°C.
- Comece derretendo a manteiga para dar tempo de esfriar.
- Bata os ovos com o açúcar por mais ou menos 5 minutos.

A mistura deve ficar clara e com espuma.

- Adicione a farinha (e caso queira uma essência de sua escolha) e bata por alguns segundos, até incorporar.

Se você bater demais, a sua massa não vai crescer.

- Adicione a manteiga derretida e misture com uma espátula ou uma colher de madeira.
- Unte a sua forma e despeje a mistura nela.
- Coloque a lâmina de uma faca na massa ao longo do comprimento do bolo.
- Asse por 10 minutos a 180°C, depois diminua a temperatura do forno para 145°C.
- Deixe assar por mais 40 minutos.

A parte debaixo do bolo deve ficar ligeiramente dourada, mas não muito.

Capítulo II

Instruções

- Tens de realizar o projeto com a equipa que tem foi atribuída. Reunir com os teus colegas de equipa faz parte das tuas tarefas, utilizando qualquer meio apropriado (Slack, e-mail, telefone, ou diretamente em pessoa).
- Se já tentaste mesmo tudo para entrar em contacto com um dos teus colegas de equipa, mas ele continua incontactável, segue as instruções da tua equipa pedagógica, se forem fornecidas. O procedimento padrão é fazeres o projeto com os colegas disponíveis e discutir a situação durante a avaliação. Mesmo que o líder de grupo esteja ausente, continuarás a ter acesso ao diretório de submissão.
- O teu trabalho deve estar em conformidade com a Norm. Se tiveres ficheiros/funções bónus, estes estão incluídos na verificação da norm, e receberás uma pontuação de 0 se houver um erro de norm.
- Deves tratar os erros de forma coerente. Podes imprimir uma mensagem de erro ou simplesmente devolver o controlo ao utilizador.
- O teu projeto deve estar concluído e enviado para o repositório Git até à data limite indicada na página principal do projeto na intranet.
- Deves seguir o procedimento de submissão descrito no final deste documento, se fornecido.
- O teu programa deve compilar utilizando o cc com as seguintes flags: -Wall -Wextra -Werror. Se existir uma moulinette, esta usará o mesmo compilador e flags.
- Se o teu programa não compilar, receberás uma pontuação de 0.
- O grupo será automaticamente inscrito para a defesa. Deves comparecer à tua avaliação com todos os teus colegas de equipa. O objetivo da defesa é apresentar e explicar detalhadamente o vosso trabalho.
- Não cancelas a tua avaliação; não terás uma segunda oportunidade.
- Cada membro do grupo deve estar plenamente consciente de todos os detalhes do projeto. Se decidirem dividir o trabalho, assegurem-se de que compreendem todas

as partes realizadas pelos outros membros da equipa. Esse conhecimento poderá ser verificado durante a avaliação.

- Deves submeter um Makefile, que irá compilar o teu projeto utilizando as regras `$NAME`, `clean` e `fclean`.

Capítulo III

Intruções IA

Contexto

A Piscina C é intensa. É o teu primeiro grande desafio na 42 — um mergulho profundo na resolução de problemas, autonomia e comunidade.

During this phase, your main objective is to build your foundation — through struggle, repetition, and especially **peer-learning** exchange.

Na era da IA, os atalhos são fáceis de encontrar. No entanto, é importante considerar se o uso da IA está realmente a ajudar-te a crescer — ou apenas a impedir-te de desenvolver competências reais.

A Piscine também é uma experiência humana — e, por agora, nada substitui isso. Nem mesmo a IA.

Para uma visão mais completa da nossa posição sobre a IA — como ferramenta de aprendizagem, como parte do currículo TIC e como uma expectativa crescente no mercado de trabalho — consulta o FAQ dedicado disponível no intranet.

● Mensagem principal

- 👉 Constrói bases sólidas sem atalhos.
- 👉 Desenvolve verdadeiramente competências técnicas e interpessoais.
- 👉 Vive a aprendizagem entre pares, começa a aprender a aprender e a resolver novos problemas.
- 👉 A jornada de aprendizagem é mais importante do que o resultado.
- 👉 Aprende os riscos associados à IA e desenvolve práticas de controlo eficazes e contramedidas para evitar os erros mais comuns.

● **Regras para os alunos:**

- Deves aplicar o raciocínio nas tarefas atribuídas, especialmente antes de recorreres à IA.
- Não deves pedir respostas diretas à IA.
- Deves aprender sobre a abordagem global da 42 em relação à IA.

● **Resultados esperados:**

Nesta fase, vais ter os seguintes resultados:

- Obter bases sólidas em tecnologia e programação.
- Compreender por que razão e de que forma a IA pode ser perigosa durante esta fase.

● **Comentários e exemplos:**

- Sim, sabemos que a IA existe — e sim, pode resolver os teus projetos. Mas estás aqui para aprender, não para provar que a IA já aprendeu. Não percas tempo (nem o nosso) apenas para demonstrar que a IA consegue resolver o problema.
- Aprender na 42 não é sobre saber a resposta — é sobre desenvolver a capacidade de encontrar uma. A IA dá-te a resposta diretamente, mas isso impede-te de construir o teu próprio raciocínio. E o raciocínio exige tempo, esforço e envolve falhas. O caminho para o sucesso não deve ser fácil.
- Lembra-te que nos exames a IA não está disponível — sem internet, sem telemóveis, etc. Vais perceber rapidamente se dependeste demasiado da IA no teu processo de aprendizagem.
- A aprendizagem entre pares expõe-te a ideias e abordagens diferentes, melhorando as tuas competências interpessoais e a tua capacidade de pensar de forma divergente. Isso é muito mais valioso do que conversar com um bot. Por isso, não sejas tímido — fala, faz perguntas e aprende em conjunto!
- Sim, a IA fará parte do currículo — tanto como ferramenta de aprendizagem como tema de estudo. Terás até a oportunidade de construir o teu próprio software de IA. Para saberes mais sobre a nossa abordagem em crescendo, consulta a documentação disponível no intranet.

✓ Boa prática:

Estou com dificuldades num novo conceito. Pergunto a alguém ao meu lado como o abordou. Falamos durante 10 minutos — e de repente faz sentido. Percebo.

✗ Má prática:

Uso a IA em segredo, copio algum código que parece estar certo. Durante a avaliação por pares, não consigo explicar nada. Falho. Durante o exame — sem IA — fico novamente bloqueado. Falho.

Capítulo IV

Enunciado

	Exercício : 00
	rush-02
Pasta de entrega :	<i>ex00/</i>
Ficheiros para entregar :	Makefile e todos os ficheiros necessários
Funções autorizadas :	write, malloc, free, open, read, close

- Deve produzir um programa que receba um número como argumento de entrada e que o transforme no seu valor escrito.
- Nome do executável: **rush-02**
- O código será compilado pelo comando:

```
make fclean  
make
```

- O programa pode receber até 2 argumentos:
 - Se existir apenas um argumento, será o valor a converter;
 - Caso existam dois argumentos, o primeiro argumento refere ao novo dicionário a utilizar, e o segundo argumento será o valor a converter.
- Se o argumento não for um inteiro positivo válido, o programa deve devolver **Error** seguido de um "**\n**".



O programa deve ir além do `unsigned int`.

- O programa deve analisar o dicionário enviado nos ficheiros anexos ao projecto. Os valores dados devem ser utilizados para imprimir os resultados. Estes valores podem ser modificados.
- Qualquer memória manualmente alocada (com malloc(3)) deve ser apropriadamente libertada. Isto será verificado durante a avaliação.
- Por uma questão de harmonização, seu programa falará em inglês.
- O dicionário seguirá estas regras:

```
[a number] [0 to n spaces]:[0 to n spaces] [any printable characters]\n
```

- Os números devem ser tratados da mesma forma que a função atoi os trata.
- Os espaços antes e depois de um valor no dicionário devem ser suprimidos.
- O dicionário deverá manter sempre as chaves de origem. O seu valor pode ser modificado, e novas entradas podem ser adicionadas, mas as chaves iniciais nunca podem ser removidas.
- Apenas precisa de utilizar as entradas iniciais (por exemplo, se for adicionado 54: fifty-four, terá que utilizar 50: fifty e 4: four).
- Podem existir linhas vazias no dicionário.
- No caso de erro na interpretação do dicionário, o programa deve devolver "Dict Error\n".
- Se não for possível devolver um dos valores pedidos a partir de um dicionário dado, o programa deve devolver "Dict Error\n".
- Exemplo:

```
$> ./rush-02 42 | cat -e
forty two$
$> ./rush-02 0 | cat -e
zero$
$> ./rush-02 10.4 | cat -e
Error$
$> ./rush-02 100000 | cat -e
one hundred thousand$
$> grep "20" numbers.dict | cat -e
20 :      hey      everybody !$
$> ./rush-02 20 | cat -e
hey everybody !$
```

Capítulo V

Bónus

- Utilizar `,`, `-`, `and` para ficar mais próximo do que é sintaticamente correto.
- Ter a possibilidade de mudar a língua (ex. para o português). Pode, para este fim, adicionar o seu próprio dicionário que conterá as entradas necessárias.
- Utilizar `read` para ler a partir do *standard input* quando o argumento correspondente ao número for “`-`”. Example:

```
$> ./rush-02 -
42
forty two
0
zero
^D
$>
```

Capítulo VI

Submissão e Avaliação

Entrega a tua tarefa no teu repositório **Git** como habitualmente. Apenas o trabalho dentro do teu repositório será avaliado durante a defesa. Não hesites em verificar duas vezes os nomes dos teus ficheiros para garantir que estão corretos.

Uma vez que estas tarefas não são verificadas por um programa, sente-te à vontade para organizar os teus ficheiros como desejas, desde que entregues os ficheiros obrigatórios e cumpras os requisitos exigidos.



Deves devolver apenas os ficheiros solicitados pelo enunciado deste projeto.