



Piscina C

Rush 00

Sumário: Este documento é o enunciado do módulo Rush 00 da Piscina C da 42.

Versão: 8.0

Conteúdo

I	Preâmbulo	2
II	Instruções	4
III	Intruções IA	6
IV	Enunciado Principal	9
V	Rush 00	11
VI	Rush 01	13
VII	Rush 02	14
VIII	Rush 03	15
IX	Rush 04	16
X	Submissão e avaliação	17

Capítulo I

Preâmbulo

Here are the lyrics of a famous TV show for everyone:

[Verse 1]

I wanna be the very best
Like no one ever was
To catch them is my real test
To train them is my cause

I will travel across the land
Searching far and wide
Each pokemon to understand
The power that's inside

[Chorus]

Pokemon! Gotta catch 'em all! It's you and me
I know it's my destiny,
Pokemon! Oh you're my best friend
In a world, we must defend
Pokemon! A heart so true
Our courage will pull us through,

You teach me and I'll teach you,
Pokemon! Gotta catch'em all

[Chorus]

Every challenge along the way
With courage I will face.
I will battle every day
To claim my rightful place.
Come with me,
The time is right,
There's no better team.
Arm in arm we'll win the fight!
It's always been our dream!

[Chorus]

I bet you were singing just now! But for now, let's stay focused; this subject has nothing to do with *Pocket Monsters*!

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 canceles 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.

- Este documento contém **cinco** versões diferentes do enunciado.
- A vossa equipa deve completar uma versão, determinada pelo seguinte cálculo:
 - Encontrar o índice numérico (de 1 a 26) da primeira letra do login do líder da equipa.
 - Calcular o módulo 5 desse número.
 - O resultado determina qual o enunciado a completar.
- Consulte o ficheiro `groups_subject.txt` em anexo para mais detalhes e exemplos.

Bonus Point:

É possível ganhar pontos extra ao submeter outras versões do projeto com os respetivos nomes. Também é possível criar um único binário que aceite um argumento da linha de comandos para alternar entre as diferentes versões.



Deve fazer o enunciado obrigatório de forma perfeita para apresentar os enunciados bônus: Se um enunciado bônus funcionar mas o original falhar os testes, o grupo terá 0.

Capítulo III

Intruições 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.

Nesta fase, o teu principal objetivo é obter uma base sólida — através do esforço, da repetição e, acima de tudo, da partilha de aprendizagens com os teus colegas.

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 recorrereres 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 Principal

	Exercício : 00
rush0X	
Pasta de entrega : <i>ex00/</i>	
Ficheiros para entregar : <code>main.c</code> , <code>ft_putchar.c</code> , <code>rush0X.c</code>	
Funções autorizadas : <code>write</code>	

O programa deve mostra um rectângulo no ecrã e deve seguir estes requisitos:

Ficheiros a submeter:

- `main.c`
- `ft_putchar.c`
- `rush0X.c`, onde "0X" representa o número do rush (e.g., `rush00.c`).
 - O próximo capítulo vai descrever as especificações de cada número do rush.

Compilação:

- Estes **três** ficheiros vão ser compilados juntos.
- O ficheiro `ft_putchar.c` tem de conter a função `ft_putchar`.

Exemplo do ficheiro `main.c`:

Requisitos das funções:

- Tens de escrever uma função chamada `rush` com as seguintes especificações:

```
int    main()
{
    rush(5, 5);
    return (0);
}
```

- Deve aceitar dois argumentos inteiros, chamados x e y.
- A função tem de ser colocada no ficheiro rush0X.c.
- A função deve ter o seguinte protótipo:

```
void    rush(int x, int y);
```

- A função rush tem de mostrar um retângulo no ecrã com a largura de x caracteres e o comprimento de y caracteres.
- A tua função nunca deve falhar nem entrar num loop infinito.
- A função main irá ser modificada durante a avaliação para verificar se trataram de todos os casos necessários.

Está aqui um exemplo dos testes que iremos realizar:

```
int    main()
{
    rush(123, 42);
    return (0);
}
```

Capítulo V

Rush 00

- `rush(5,3)` deverá mostrar:

```
$> ./a.out
o---o
|   |
o---o
$>
```

- `rush(5, 1)` deverá mostrar:

```
$> ./a.out
o---o
$>
```

- `rush(1, 1)` deverá mostrar:

```
$> ./a.out
o
$>
```

- `rush(1, 5)` deverá mostrar:

```
$> ./a.out
o
|
|
|
o
$>
```

- `rush(4, 4)` deverá mostrar:

```
$> ./a.out  
o--o  
|  |  
|  |  
o--o  
$>
```

Capítulo VI

Rush 01

- `rush(5,3)` deverá mostrar:

```
$> ./a.out
/***\
*   *
\***/
$>
```

- `rush(5, 1)` deverá mostrar:

```
$> ./a.out
/***\
$>
```

- `rush(1, 1)` deverá mostrar:

```
$> ./a.out
/
$>
```

- `rush(1, 5)` deverá mostrar:

```
$> ./a.out
/
*
*
*
\
$>
```

- `rush(4, 4)` deverá mostrar:

```
$> ./a.out
/**\
*  *
*  *
\**/
$>
```

Capítulo VII

Rush 02

- `rush(5,3)` deberá mostrar:

```
$> ./a.out
ABBBA
B  B
CBBBC
$>
```

- `rush(5, 1)` deberá mostrar:

```
$> ./a.out
ABBBA
$>
```

- `rush(1, 1)` deberá mostrar:

```
$> ./a.out
A
$>
```

- `rush(1, 5)` deberá mostrar:

```
$> ./a.out
A
B
B
B
C
$>
```

- `rush(4, 4)` deberá mostrar:

```
$> ./a.out
ABBA
B  B
B  B
CBBC
$>
```

Capítulo VIII

Rush 03

- `rush(5,3)` deberá mostrar:

```
$> ./a.out
ABBBC
B  B
ABBBC
$>
```

- `rush(5, 1)` deberá mostrar:

```
$> ./a.out
ABBBC
$>
```

- `rush(1, 1)` deberá mostrar:

```
$> ./a.out
A
$>
```

- `rush(1, 5)` deberá mostrar:

```
$> ./a.out
A
B
B
B
A
$>
```

- `rush(4, 4)` deberá mostrar:

```
$> ./a.out
ABBC
B  B
B  B
ABBC
$>
```


Capítulo IX

Rush 04

- `rush(5,3)` deberá mostrar:

```
$> ./a.out
ABBBC
B  B
CBBBA
$>
```

- `rush(5, 1)` deberá mostrar:

```
$> ./a.out
ABBBC
$>
```

- `rush(1, 1)` deberá mostrar:

```
$> ./a.out
A
$>
```

- `rush(1, 5)` deberá mostrar:

```
$> ./a.out
A
B
B
B
C
$>
```

- `rush(4, 4)` deberá mostrar:

```
$> ./a.out
ABBC
B  B
B  B
CBBA
$>
```

Capítulo X

Submissão e avaliação

Entrega o teu trabalho no teu repositório `Git`, como habitual. Apenas o trabalho dentro do teu repositório será avaliado durante a defesa. Não hesites em confirmar os nomes dos teus ficheiros para ter a certeza que estão corretos.

Como estes trabalhos não são avaliados por um programa, sinta-se livre de organizar os ficheiros como preferir, desde que entregue os ficheiros obrigatórios e que cumpra com os requisitos.



Apenas precisas de entregar os ficheiros pedidos no enunciado deste projeto.