
Trabalho Prático I: Extração-Z

Trabalho Individual. Valor: 10 pontos

Entrega: 12 de Fevereiro de 2021

1 Introdução

No futuro, a Terra é um planeta bem diferente do que era no século XXI. O esgotamento das principais reservas ambientais e o colapso climático comprometeram as condições de habitação na maioria dos biomas. A esperança da humanidade concentra-se em um novo composto mineral recém-descoberto, denominado Composto Z, essencial para desenvolver as tecnologias de terraformação que podem restaurar as condições de vida na Terra.

Porém, o Composto Z só pode ser encontrado em um planeta distante, localizado em uma outra galáxia. Felizmente, os avanços tecnológicos desenvolvidos pelo Instituto de Ciências Exoplanetárias viabilizaram a construção de naves de dobra espacial capazes de fazer viagens intergaláticas em tempo hábil. Assim teve início a missão Extração-Z, cujo objetivo é explorar outros mundos em busca do precioso recurso do qual depende a humanidade.

A superfície do planeta é acidentada e habitada por formas de vida alienígenas hostis, o que torna inviável a participação de humanos na missão. Robôs semi-autônomos são ideais para conduzir essa tarefa, por serem mais resistentes à viagem e poderem minerar com muito mais facilidade. Apesar do grande progresso tecnológico alcançado até agora, ainda não há um sistema controlador de robôs para coordenar a operação. Seu trabalho é, portanto, projetar e construir tal sistema.

2 Especificações

O Instituto de Ciências Exoplanetárias fez o mapeamento prévio das regiões que seus robôs devem explorar. Os dados são expressos numa matriz, conforme a exemplificada na Figura 1. Nela, o caractere ‘R’ indica a presença de recursos; ‘O’, um espaço inatingível (obstáculo); ‘H’, uma forma de vida alienígena hostil; e o caractere ‘.’ (ponto) é uma região vazia. Cada posição é representada por um índice ‘(i, j)’, no qual ‘i’ representa a **linha** e ‘j’, a **coluna**. A origem ‘(0,0)’ é o **canto superior-esquerdo**, onde sua nave pousará e formará uma base (representada por ‘B’).

B	R	O	O	O	O	O	O	O	O	O	O
.	R	O	O	O	O	O	O	O	O	O
.	H	H	R	O	O	O	O	O
.	R	R	O	H	H	H	.	.	.	R	R	O	O	O	O	O
.	.	R	O	O	R	O	O	R	R	R	O	O	O	O
.	.	R	O	O	O	R	R	R	O	O	O	O	O	.	.
.	H	H	.	.	.	O	O	R	R
.	H	H	.	.	.	O	O	O	R	H	H	.	.
.	R	O

Figura 1: Mapa obtido através dos dados do Instituto de Ciências Exoplanetárias.

Cada base comporta exatamente **cinquenta** robôs, que podem explorar qualquer espaço do mapa, exceto obstáculos. Seu sistema também receberá uma lista de comandos a executar, categorizados em dois tipos: **ordens de comando**, e **comandos diretos**. As ordens de comando são destinadas somente aos robôs e **NÃO** são executadas no momento que são recebidas. Em vez disso, elas são acumuladas no robô de destino. Caso um robô receba uma ordem direta de **EXECUÇÃO**, ele deverá executar **TODAS** as ordens acumuladas. As ordens de comando seguem a política *first in first out* (FIFO), ou seja, a primeira que chegar será a primeira a ser executada.

2.1 Ordens de comando

Cada ordem de comando é direcionada a um robô específico que é representado por um inteiro positivo ' k ' $\in [0, 49]$. Todas as **ordens de comando** recebidas por um robô são acumuladas com política FIFO e são descritas a seguir. A cada comando executado pelo robô, ele deverá registrar no seu **histórico** a ação executada, de acordo com o formato específico para cada comando. O histórico deverá ser impresso apenas quando o comando '**RELATORIO**' (que será explicado mais adiante) for emitido.

Algumas ordens de comando podem ser enviadas com prioridade, sinalizada por um asterisco '*' antes do nome do comando. **Uma ordem desse tipo tem prioridade sobre as demais e será, portanto, a próxima a ser executada.**

- '**MOVER** k (i, j)': Ao executar esse comando o robô ' k ' se moverá até a posição ' (i, j) ' indicada.

Algumas observações devem ser notadas:

- Vários robôs diferentes podem ocupar uma mesma posição do mapa, mesmo que no local haja um recurso ou alienígena.
- Você deve assumir que o robô se move para a posição designada automaticamente. Dessa forma, não é necessário calcular caminhos, nem planejar trajetórias da posição atual até a indicada.
- Caso exista um obstáculo na posição ' (i, j) ', o robô deverá registrar em seu histórico:

ROBO k : IMPOSSIVEL MOVER PARA (i, j)

- Caso não exista um obstáculo na posição ' (i, j) ', o robô deverá registrar em seu histórico:

ROBO k : MOVEU PARA (i, j)

- '**COLETAR** k ': Ao executar esse comando o robô ' k ' tentará coletar recursos na posição em que se encontra. Algumas observações devem ser notadas:

- Caso não existam recursos na posição ' (i, j) ' o robô deverá registrar em seu histórico:

ROBO k : IMPOSSIVEL COLETAR RECURSOS EM (i, j)

- Caso existam recursos na posição ' (i, j) ' o robô deverá somar **UMA** unidade a seu contador de coleta e registrar em seu histórico:

ROBO k : RECURSOS COLETADOS EM (i, j)

- Recursos coletados devem ser removidos do mapa e substituídos pelo caractere . (ponto).

- '**ELIMINAR** k ': Ao executar esse comando, o robô ' k ' tentará eliminar alienígenas na posição em que se encontra. Algumas restrições devem ser notadas:

- Caso não existam alienígenas na posição ' (i, j) ', o robô deverá registrar em seu histórico:

ROBO k: IMPOSSIVEL ELIMINAR ALIEN EM (i,j)

- Caso existam alienígenas na posição '(i,j)', o robô deverá registrar em seu histórico:

ROBO k: ALIEN ELIMINADO EM (i,j)

- Alienígenas eliminados devem ser removidos do mapa e substituídos pelo caractere . (ponto).

2.2 Ordens diretas

Todas as ordens diretas são enviadas diretamente da base para os robôs e têm prioridade sobre ordens de comando, ou seja, mesmo que ainda haja comandos acumulados na fila no robô 'k', ele deverá ignorá-los, obedecer imediatamente à ordem direta e imprimir a saída correspondente.

- 'ATIVAR k': Envia o robô 'k' para explorar o planeta.

- Caso o robô já tenha saído para explorar, a base deverá imprimir na saída:

BASE: ROBO k JA ESTA EM MISSAO

- Caso o robô não esteja em missão, a base deverá imprimir na saída:

BASE: ROBO k SAIU EM MISSAO

- Uma vez que a base será sempre construída na localização identificada pela origem do mapa, todos os robôs ativados iniciam suas atividades de exploração no ponto '(0,0)'.

- 'EXECUTAR k': Faz com que o robô 'k' execute todas as ordens de comando acumuladas.

- Caso o robô não esteja em uma missão, a base deverá imprimir na saída:

BASE: ROBO k NAO ESTA EM MISSAO

caso contrário, não é necessário emitir uma saída neste ponto, pois as ordens de comando executadas serão emitidas no relatório.

- 'RELATORIO k': Faz com que o robô 'k' imprima na saída o seu **histórico**.

- 'RETORNAR k': Faz com que o robô 'k' volte para a base.

- Ao retornar para a base, a base deverá imprimir na saída quantos alienígenas foram eliminados e quantos recursos foram coletados:

BASE: ROBO k RETORNOU ALIENS x RECURSOS y

- Caso o robô não esteja em uma missão, a base deverá imprimir na saída:

BASE: ROBO k NAO ESTA EM MISSAO

- O **histórico** de comandos também deve ser **apagado**.
 - Os recursos capturados pelo robô 'k', assim como o número de alienígenas eliminados por ele, devem ser transferidos para a base. Os contadores do robô devem ser zerados ao retornar para a mesma.

Ao final da execução de todos os comandos, a base deverá imprimir na saída um relatório geral informando quantos recursos foram coletados e quantos alienígenas foram eliminados:

BASE: TOTAL DE ALIENS x RECURSOS y

É importante notar que esses valores representam quantidades totais, a soma de **todos os robôs**.

3 Entrada

A entrada é dividida em dois arquivos: ‘`mapa.txt`’, que contém o mapa, e ‘`comandos.txt`’, que contém todos os comandos que devem ser executados pela base e pelos robôs de exploração.

3.1 ‘`mapa.txt`’

No arquivo de entrada ‘`mapa.txt`’, a primeira linha contém a altura e a largura do mapa, isto é, a quantidade de linhas e colunas, separadas por um espaço. A partir da linha seguinte, cada linha do arquivo corresponde a uma linha do mapa, com as colunas separadas por um espaço. Na Entrada 1, encontra-se exemplificado um arquivo ‘`mapa.txt`’ para um mapa de tamanho 10×10 .

```
10 10
B . . . . . 0 0 0 .
. . . . . 0 0 0 .
. . . . R 0 0 0 R .
. . . R 0 0 0 0 R .
. H . 0 0 0 0 R . .
. . . R 0 0 R . . .
. . . . . . . . . .
. . . H R R . . . .
. . . H R R . . . .
. . . . H . . . . .
```

Entrada 1: Exemplo de arquivo para o mapa de entrada.

3.2 ‘`comandos.txt`’

O arquivo ‘`comandos.txt`’ contém todos os comandos diretos e ordens de comando executadas pela base e robôs, distribuídos um por linha.

Como exemplificado na Entrada 2, a sequência ativa o robô ‘0’ na posição ‘(0,0)’ com o comando direto ‘`ATIVAR 0`’. Posteriormente, é enviado uma sequência de ordens de comando. A ordem de comando ‘`MOVER 0 (1,1)`’ é executada assim que o comando direto ‘`EXECUTAR 0`’ é lido. É importante notar que a ordem de execução dos comandos ‘`*COLETAR 0`’ e ‘`*MOVER 0 (2,4)`’ será alterada pois estão sendo chamados com prioridade, ao entrarem na lista de comandos, eles entram na frente dos que já estiverem lá. Desta forma, quando o comando ‘`*COLETAR 0`’ entrar, ele irá para o início da lista. Em seguida, o comando ‘`*MOVER 0 (2,4)`’ entrará na frente de ‘`*COLETAR 0`’. Assim, quando o comando ‘`EXECUTAR 0`’ for chamado, o comando ‘`*MOVER 0 (2,4)`’ será executado, seguido pelo comando ‘`*COLETAR 0`’ e os demais ‘`MOVER 0 (0,6)`’, ‘`MOVER 0 (4,1)`’ e ‘`ELIMINAR 0`’, respectivamente.

O comando ‘`RELATORIO 0`’ solicita a impressão do histórico do robô 0. Por fim, ‘`RETORNAR 0`’ faz com que a base reporte quantos alienígenas foram eliminados e quantos recursos foram coletados pelo robô ‘0’.

```
ATIVAR 0
MOVER 0 (1,1)
EXECUTAR 0
MOVER 0 (0,6)
*COLETAR 0
*MOVER 0 (2,4)
MOVER 0 (4,1)
ELIMINAR 0
EXECUTAR 0
RELATORIO 0
RETORNAR 0
```

Entrada 2: Arquivo de exemplo com as ordens de comando e os comandos diretos a serem executados.

4 Saída

Utilize a **saída padrão** ('stdout') para imprimir todas as informações. A saída deve conter a resposta de cada comando, conforme explicado na Seção 2 (Especificações). Por exemplo, para o mapa da Entrada 1 e comandos da Entrada 2, obtém-se a saída ilustrada na Saída 1.

```
BASE: ROBO 0 SAIU EM MISSAO
ROBO 0: MOVEU PARA (1,1)
ROBO 0: MOVEU PARA (2,4)
ROBO 0: RECURSOS COLETADOS EM (2,4)
ROBO 0: IMPOSSIVEL MOVER PARA (0,6)
ROBO 0: MOVEU PARA (4,1)
ROBO 0: ALIEN ELIMINADO EM (4,1)
BASE: ROBO 0 RETORNOU ALIENS 1 RECURSOS 1
BASE: TOTAL DE ALIENS 1 RECURSOS 1
```

Saída 1: Saída esperada para uma execução cujas entradas são o mapa apresentado na Figura 1 e comandos apresentados na Figura 2.

5 Observações importantes e dicas

Eis a seguir uma lista de aspectos implementacionais relevantes que merecem sua atenção:

- Robôs podem ocupar a mesma posição de outros robôs, alienígenas, e recursos.
- Robôs não executam ordens de comando imediatamente após recebê-las. Elas são acumuladas conforme a ordem de entrada (**FIFO**) e são executadas somente ao receber um comando 'EXECUTA k', em que 'k' é o identificador do robô.
- O relatório de ações do robô só é impresso na saída após o um comando direto de 'RELATORIO k', no qual 'k' é o identificador do robô.
- Ao terminar a execução, deve-se imprimir o relatório geral da base no seguinte formato: 'BASE: ALIENS x RECURSOS y'.
- Após retornar à base em decorrência de uma ordem direta de 'RETORNO', o histórico de ações executadas pelo robô, assim como seus contadores de recursos e aliens eliminados são limpos e os recursos e alienígenas coletados são transferidos para a base.

- Você deve assumir que seu robô é capaz de se deslocar até as posições designadas instantaneamente, dessa forma **NÃO** é necessário fazer buscas ou computar caminhos.
- Note que as saídas **não utilizam** caracteres acentuados e estão escritas em letras **MAIÚSCULAS**. Preste atenção aos espaçamentos e sinais de pontuação, como parênteses e dois-pontos, a saída deve corresponder **exatamente** ao que está especificado neste documento.
- Certifique-se de **sempre** liberar eventuais espaços de memória alocada manualmente, caso utilize ‘`malloc / new`’, ou funções similares.
- Tome cuidado ao manipular ponteiros para evitar erros de acesso.
- Ferramentas de *debug* podem ser bastante úteis para identificar problemas no código, utilize-as à vontade.

6 Entregáveis

Você deve utilizar a linguagem C ou C++ para o desenvolvimento do seu sistema. O uso de estruturas pré-implementadas pelas bibliotecas-padrão da linguagem ou terceiros é **terminantemente vetado**. Utilize apenas as variáveis de tipos primitivos e derivados para criar **suas próprias implementações** para todas as classes, estruturas, e algoritmos.

Organize seu código-fonte em arquivos separados conforme a responsabilidade de cada objeto e entidade, com nomenclatura condizentes ao que executam. A fim de padronizar a compilação e o projeto, você **DEVE utilizar** a estrutura de projeto abaixo junto ao ‘`Makefile`’ disponibilizado no *Moodle*:

```
- TP1
  |- src
  |- bin
  |- obj
  |- include
  Makefile
```

A pasta ‘`TP1`’ é a raiz do projeto; a pasta ‘`bin`’ deve conter os executáveis gerados após a compilação; ‘`src`’ deve armazenar arquivos de código (‘`*.c`’, ‘`*.cpp`’, ou ‘`*.cc`’); e ‘`include`’, os cabeçalhos (*headers*) do projeto, com extensão ‘`*.h`’.

O executável do seu programa deverá receber como parâmetro duas strings, sendo a primeira o nome do arquivo do mapa a ser utilizado e a segunda o nome do arquivo contendo os comandos:

```
run.out mapa.txt comandos.txt
```

Procure seguir boas práticas de programação: cada bloco deve ser indentado apropriadamente; utilize nomes descritivos para variáveis; e, se julgar necessário, deixe breves comentários em pontos relevantes do código, mas **evite excessos**.

6.1 Documentação

A documentação do trabalho deve ser entregue em formato **pdf** e também deverá seguir o modelo de relatório que será postado no Moodle.

Ela deve conter **todos** os itens descritos abaixo. Procure escrever de forma sucinta e objetiva.

- Título, nome, e matrícula.
- Introdução com apresentação do problema e visão geral sobre o programa.

- Instruções para compilação e execução.
- Descrição da implementação que detalhe as estruturas, classes e métodos. Comentários a respeito do eventual tratamento dado a casos e detalhes que não tenham sido previstos na especificação do trabalho.
- Estudo da complexidade de tempo e espaço dos procedimentos implementados, formalizado pela notação assintótica.
- Conclusões, observações e considerações finais.
- Bibliografia com as fontes consultadas para realização do trabalho.

6.2 Submissão

Todos os arquivos relacionados ao trabalho devem ser submetidos na atividade designada para tal no *Moodle* dentro do prazo estipulado. A entrega deve ser feita **em um único arquivo**, com nomenclatura ‘nome_sobrenome_matricula.zip’. Este deve conter um arquivo ‘pdf’, para a documentação, e uma pasta ‘projeto’, contendo tanto o código-fonte e um ‘Makefile’ para compilar o mesmo.

7 Considerações Finais

1. Leia **atentamente** o documento de especificação, pois o descumprimento de quaisquer requisitos obrigatórios aqui descritos causará penalizações na nota final.
2. Os principais aspectos avaliados na correção serão:
 - Quantidade de acertos para cada teste realizado e corretude da solução;
 - Implementação adequada das estruturas especificadas;
 - Conteúdo da documentação e estudos de complexidade;
 - Boas práticas de implementação e organização do código.
3. Certifique-se de garantir que seu arquivo foi submetido corretamente no sistema.
4. Em caso de dúvida, não hesite em perguntar nos fóruns de discussão da disciplina ou aos monitores. Encorajamos a participação através do fórum para estimular a comunicação da turma no regime de ensino remoto emergencial.
5. **Plágio é CRIME.** Trabalho onde o plágio for identificado serão **automaticamente anulados** e as medidas administrativas cabíveis serão tomadas. Discussões a respeito do trabalho entre colegas são permitidas. É permitido consultar fontes externas, desde que exclusivamente para fins didáticos e devidamente registradas na sessão de bibliografia da documentação. **Cópia e compartilhamento de código não são permitidos.**
6. Procure começar o trabalho com antecedência, pois atrasos sofrerão penalização de $2^d - 1$ pontos, com d = dias de atraso.

Bom trabalho!