**PROJETO LINGUAGEM DE PROGRAMAÇÃO**

**(ROBÔ LIMPEZA)**

Você foi convocado para programar um novo robô que auxiliará na limpeza de cômodos.

**CÔMODO**

- Um cômodo pode ser visto como uma matriz de 8x8 e inicialmente cada cruzamento de linha e coluna deve conter:

- 1 parede; ou

- 1 sujeira.

- Apenas os espaços que contêm sujeira e a estação de recarga são navegáveis, pois são de fato o cômodo.

- As paredes de um cômodo não podem ser destruídas.

**ROBÔ**

- O robô possui uma “estação de recarga”, localizada em apenas um dos espaços do cômodo. Essa estação também serve como o ponto de partida do robô.

- O robô só pode movimentar-se uma linha ou coluna por vez.

- O robô só é capaz de limpar sujeiras que estão até 1 espaço de distância (ao norte, ao sul, ao leste e ao oeste).

- Para auxiliar na sua movimentação, o robô possui sensores capazes de detectar o conteúdo dos espaços perpendiculares a sua localização atual (ao norte, ao sul, ao leste e ao oeste).

- Seu robô é capaz de memorizar o conteúdo dos espaços que já ocupou e/ou que foram analisados por seus sensores.

- Seu robô pode memorizar o próprio trajeto.

- Você pode informar ao seu robô o tamanho do cômodo que ele deve limpar.

- Seu robô pode memorizar o quanto do quarto já foi limpo.

- Seu robô não pode ocupar o mesmo espaço que uma parede.

**OBJETIVO PRINCIPAL (Nota 0-10 para a Unidade II)**

- Remover toda a sujeira do quarto.

**OBJETIVO EXTRA (Nota 0-10 para a Recuperação da Unidade I)**

- Após realizar toda a limpeza, o robô deve retornar para a sua “estação de recarga”. (0-5)

- O robô deve fazer o menor caminho possível até sua estação. (0-5)

**DICAS**

1. Usar valores inteiros (int) é uma ótima forma de representar os diferentes elementos que constituem o problema. Exemplo: você pode usar o valor 1 para representar uma parede, 2 para sujeira, 3 para espaço limpo, etc...
2. Use a capacidade do seu robô de memorizar os espaços já visitados ou escaneados pelo sensor para tomar decisões mais conscientes sobre onde ir em seguida.
3. Faça com que o seu robô priorize seguir na direção de espaços ainda não visitados.
4. Use a estratégia dos valores inteiros da dica 1 para representar espaços visitados e não visitados.
5. Use subprogramas para executar o projeto, assim fica mais fácil de verificar possíveis erros durante o desenvolvimento.
6. Usar **STRUCTS** é uma boa forma de evitar o excesso de parâmetros que possivelmente serão necessários nos subprogramas que você irá criar.
7. Lembre-se de usar passagem de parâmetro por referência sempre que possível, assim você sempre poderá modificar os valores de variáveis em diferentes escopos.
8. Represente a arena como uma matriz e sempre que precisar passá-la por parâmetro a outros subprogramas não esqueça de passar também o número de linhas e colunas que ela possui. Essas informações são valiosas na hora de verificar se há algum erro de acesso da matriz. Por exemplo: evitar ler ou escrever em um índice da matriz que esteja fora de seus limites.
9. Use **printf()** para “desenhar” na tela e escrever as mensagens necessárias para que possamos entender o que está ocorrendo.
10. Use a função **\_sleep(int)** para suspender temporariamente a execução do programa. Assim, você conseguirá ver o movimento do robô pelo laboratório frame a frame.
11. Segue uma lista de ideias para algumas funções/procedimentos que você pode utilizar na sua implementação:

void InitQuarto(int Quarto[][MAX\_Y],int MaxX,int MaxY,int \*Tamanho,Robo \*MeuRobo);

Procedimento que permite definir onde estão as paredes e a sujeira do cômodo, bem como a “estação de recarga do robô”.

void InitRobo(Robo \*MeuRobo,int PosInicialX,int PosInicialY);

Procedimento que inicializa as variáveis que constituem o robô.

void DesenharQuarto(int Quarto[][MAX\_Y],int MaxX,int MaxY,Robo MeuRobo);

Procedimento responsável por desenhar o cômodo e a situação de cada espaço que o compõe.

void PrintStatusRobo(Robo MeuRobo,int TamanhoQuarto);

Procedimento usado para imprimir na tela informações atualizadas sobre o robô. Ex: progresso da limpeza, posição atual, etc...

int Limpar(int Quarto[][MAX\_Y],int MaxX,int MaxY,int TamanhoQuarto,Robo \*MeuRobo);

Função responsável por controlar o percurso que o robô deve realizar enquanto limpa o cômodo.

int Carregar(int Quarto[][MAX\_Y],int MaxX,int MaxY,int TamanhoQuarto,Robo \*MeuRobo);

Função responsável por controlar o percurso que o robô deve realizar para retornar a sua “estação de recarga”.

int Sensor(int Quarto[][MAX\_Y],int MaxX,int MaxY,Robo \*MeuRobo,int Direcao);

Função que analisa o conteúdo de um espaço adjacente na direção apontada.

int MoverRobo(int Quarto[][MAX\_Y],int MaxX,int MaxY,Robo \*MeuRobo,int Direcao);

Função que controla o reposicionamento do robô quando este deve mover-se de um espaço a outro, na direção apontada.

int ChecarFim(int Tamanho,Robo MeuRobo);

Função que verifica se o robô já cumpriu a sua missão.