

گرامر نهایی و اعمال معنایی

- با توجه به دو بعدی بودن حرکت روبات از جهت های چپ (L) به جای غرب یا West، راست (R) به جای شرق یا East، بالا (U) به جای شمال یا North، پائین (D) به جای جنوب یا South استفاده کردیم.
- برای تفهیم بهتر پسوند رتبه عددی (مانند st در 1 و ...) از تابع rankpost استفاده کردیم؛ بدین گونه که عدد را گرفته و در خروجی پسوند ترتیبی مربوط به آن عدد را می دهد.
- تابع finalPositionX مختصات فعلی روبات را همراه با دستور حرکت (راست، چپ، بالا، پایین) گرفته و مولفه افقی مختصات روبات را تغییر میدهد تا مختصات جدید را نشان دهیم.
- تابع finalPositionY مختصات فعلی روبات را همراه با دستور حرکت (راست، چپ، بالا، پایین) گرفته و مولفه عمودی مختصات روبات را تغییر میدهد تا مختصات لحظه ای را نشان دهیم.
- تابع gridBoundControl محدوده صفحه مجاز و نیز مختصات لحظه ای روبات را گرفته و با مقایسه آن با کران های بالا و پایین در راستاهای قائم و افقی تعیین میکند که آیا روبات داخل محدوده مجاز است (بر میگردداند true) یا نه (بر میگردداند false).
- تابع obstControl لیست شامل مختصات موانع، تعداد موانع و مختصات فعلی روبات را گرفته و اگر مولفه افقی لحظه ای روبات با هر یک از اندیس های زوج لیست مساوی شد و همزمان مولفه عمودی لحظه ای روبات با هر یک از اندیس های فرد لیست مساوی شد، آنگاه یعنی روبات به مانع برخورد کرده (بر میگردداند false) و در غیر این صورت به مانع برخورد نشده و بر میگردداند true.
- در تابع main ابتدا رنگ فونت و background صفحه به ترتیب به زرد (كد) و آبی (كد۱) تغییر یافته است .
- محدوده های ناحیه مجاز حرکت (کران ها) از کاربر گرفته شده، همچنین ساختارهای امنیتی و کنترلی تا در هیچ راستای افقی یا قائم، کران بالا از کران پایین کمتر نشود.
- مختصات نقطه شروع روبات از کاربر گرفته شده به همراه ساختار کنترل ورودی تا مطمئن شویم روبات شروع اش از ناحیه مجاز بوده است.

- سپس با چاپ عبارت **How many obstacles exist?** کاربر باید تعداد موانع در ناحیه مجاز را وارد کند و سپس مختصات آنها را در ادامه

- بعد از ******* Start move commands with (begin) and exit with (end) *******

- ********* باید دستورات جهت دهی را به صورت مزبور (راست، چپ، پایین و بالا) وارد کنیم.
- در صورت برخورد با مانع پیام خطای **Collision with barrier** و در صورت خروج از محدوده مجاز پیام **Out Of Grids Boundaries Error** نمایش داده میشود.

- موقعیت لحظه ای تنها در صورت عدم برخورد به مانع و نیز حرکت در محدوده مجاز در صفحه نمایش داده خواهد شد.

```
if (gridBoundControl(Lx, Ux, Ly, Uy, Xf, Yf) == true && obstControl(obstaclesOrdination,
2 * ObsN, Xf, Yf) == true)
    cout << "Current position = (" << Xf << ", " << Yf << ")";
```

- هر بار حرکت مجاز که یک واحد میباشد یک واحد به مسافت اضافه میشود.

```
if (gridBoundControl(Lx, Ux, Ly, Uy, Xf, Yf) == true )
    distance ++ ;
```

- جابه جایی را نیز با استفاده از رابطه فیثاغورس محاسبه کردیم.

```
displacement = sqrt(pow(Ox - Xf, 2) + pow(Oy - Yf, 2));
```