

Ejercicio de Práctica en Papel: “Amartizando”

Nombre y Apellido: Comisión:.....

El objetivo de este ejercicio es evaluar la aplicación integral de contenidos. Para esto, evaluaremos la solución algorítmica desarrollada en base a la información definida en la consigna, la forma en la que diseñas y escribes código C++, y la aplicación de buenas prácticas de programación. El puntaje final obtenido corresponderá al global de la solución entregada. No se califica por incisos.

Todas las entregas deben tener nombre y apellido. Tener en cuenta la legibilidad de la solución entregada, ya que de no comprenderse lo escrito no se podrá corregir. Deben entregarse todas las hojas que contengan código asociado a la solución.

Tiempo de Resolución: 90 minutos.

Puntaje Requerido: 20/40 puntos.

Estás en el equipo que programa la última sonda enviada a Marte, y se te solicita:

a) hacer el programa de seguimiento del amartizaje. El mismo consiste en tomar los datos de la sonda durante el descenso hasta que se amartiza, y luego entregar algunos resultados.

El programa toma cada 1 segundo una lectura de 3 valores:

- Altitud (en metros)
- Distancia al lugar de amartizaje previsto (en metros)
- Dirección (1 letra mayúscula N,S,E,O)

La distancia y dirección actualizan las coordenadas (x,y) de la sonda. Los valores iniciales de x e y son (0,0). En cada lectura, se actualiza el (x,y) previo de la siguiente manera:

- Si la dirección es norte (N), se suma su distancia al valor de y, mientras que si es sur (S) se resta su distancia al valor y.
- Si la dirección es este (E), se suma su distancia al valor de x, mientras que si es oeste (O), se resta su distancia al valor de x.

El programa debe leer las ternas de datos de cada medición llamando a la función *leerDatosTerna(...)*. Ud. debe diseñar/codificar esta función. El ingreso de datos termina cuando la sonda efectivamente amartizó y la altitud leída en la terna es 0. Se debe mostrar por pantalla:

- 1) Velocidad promedio de descenso de la sonda desde que se inició la toma de datos hasta el aterrizaje (2 decimales de precisión). La misma se calcula como (altura inicial / lecturas por segundo), en el ejemplo $5000/9$, lo que da 555.55m/s .
- 2) Valores finales de las coordenadas x e y que han sido calculadas en base a cada medición. La posición final (coordenadas x,y) se calculan como la sumatoria de $x=(4-6+3-3) = -2$ y la sumatoria de $y=(2-3+6-4+2) = 3$.

<u>Ejemplo Entrada:</u>	<u>Ejemplo Salida:</u>
5000 2 N 4500 4 E 2500 3 S 1000 6 O 500 6 N 250 4 S 100 3 E 50 2 N 0 3 O	Velocidad promedio: 555.55m/s Posición Final: -2,3

b) Se desea hacer un upgrade del software de movimiento de la sonda (ahora sólo se desplaza en un único eje por movimiento). Se te pide que codifiques una función recursiva *Movimientos(X,Y)*, que retorne la cantidad de movimientos necesarios para, dada una posición (X,Y) de la sonda ($0 \leq X$, $0 \leq Y$) determinar cuantos movimientos de acercamiento son necesarios para que la sonda llegue al origen de coordenadas. La sonda realiza un movimiento cuando:

- se acerca a (0,0) avanzando en diagonal, acercándose a la vez con respecto al eje X y al eje Y, en una unidad.
- se acerca a (0,0) con respecto al eje X en una unidad (porque ya está en $Y=0$)
- se acerca a (0,0) con respecto al eje Y en una unidad (porque ya está en $X=0$)

La sonda nunca debe llegar a ubicarse en coordenadas con algún valor negativo, y siempre que sea posible debe elegir:

- tomar la opción diagonal, antes de la vertical ú horizontal.
- tomar la opción vertical antes de la horizontal

Así, si la sonda se encuentra en la posición (5,5) su función *Movimientos()* debe retornar 5 (que corresponde a 5 pasos en diagonal, para llegar al origen de coordenadas (luego de haber pasado por (4,4), (3,3), (2,2) y (1,1)). Mientras que si la sonda se encuentra en la posición (4,2) debe retornar 4 (que corresponde a 4 pasos para llegar al origen de coordenadas (luego de haber pasado por (3,1), (2,0) y (1,0)).