

Descripción prueba técnica

Los candidatos pueden utilizar cualquier lenguaje de programación (aunque se valorará más C# y WPF) para solucionar las diversas partes de la prueba y deberán subir el código fuente a algún repositorio Git y compartir dicho repositorio con carlos_martinez@stuller.com así como indicar las respuestas obtenidas en un fichero txt adjunto en dicho repositorio.

También subir al repositorio los binarios de los ejecutables para Windows (ficheros EXE)

Parte 1

Santa Claus ha tenido un problema de camino a su entrada gravitacional alrededor de la Luna, el ordenador de su nave emite un pitido con un mensaje de error "alarma del programa 1202". En la radio, un elfo está explicando cómo gestionar la situación: "No te preocupes, es perfectamente normal..."

Justo entonces el ordenador de la nave estalla en llamas...

El elfo explica: "Ese ordenador ejecutaba programas Intcode como el programa de asistencia gravitacional que se estaba ejecutando; seguramente hay suficientes piezas de repuesto para construir un nuevo ordenador"

Un programa Intcode es una lista de enteros separados por comas (como "1,0,0,3,99"). Para ejecutarlo, comienza mirando el primer entero (llamado posición 0). Aquí, encontrarás un **opcode** – puede ser 1, 2, o 99. El **opcode** indica qué hacer; por ejemplo, el 99 significa que el programa ha terminado y debería detenerse inmediatamente. Encontrar un **opcode** desconocido significa que algo salió mal.

El **Opcode 1** suma los números leídos desde dos posiciones y almacena el resultado en una tercera posición. Los tres números enteros inmediatamente después del código de operación te dicen estas tres posiciones - las dos primeras indican las posiciones desde donde leer los valores de entrada, y el tercero indica la posición en la que la salida debe ser almacenada.

Por ejemplo, si tu programa Intcode encuentra "1,10,20,30", debería leer los valores en las posiciones 10 y 20, sumar esos valores, y luego escribir el resultado en la posición 30.

El **Opcode 2** funciona exactamente igual el **opcode 1**, excepto que multiplica las dos entradas en lugar de sumarlas. De nuevo, los tres números enteros después del código de operación indican donde están las entradas y salidas, no sus valores. Una vez que termines de procesar un **opcode** pasa al siguiente moviéndote adelante 4 posiciones.

Una vez que tengas el programa **Intcode** funcionando, hay que restaurar el fichero de entrada gravitacional (el fichero de entrada "**Input.txt**") al estado "alarma del programa 1202" que tenía justo antes de que el ordenador se incendiase. Para hacer esto, antes de ejecutar el programa, hay que sustituir la posición 1 con el valor 12 y sustituir la posición 2 con el valor 2. ¿Qué valor queda en la posición 0 después de que el programa se ejecute?

Respuesta:

Ejemplo de programa IntCode

Supongamos que tienes el siguiente programa:

1,9,10,3,2,3,11,0,99,30,40,50

A efectos ilustrativos, aquí está el mismo programa dividido en múltiples líneas:

```
1,9,10,3
2,3,11,0
99,30,40,50
```

Los primeros cuatro números enteros, "1,9,10,3" representan el primer opcode (1, suma), las posiciones de las dos entradas (9 y 10), y la posición de la salida (3). Para solucionar este opcode, primero necesitas obtener los valores de las posiciones de entrada: la posición 9 contiene 30, y la posición 10 contiene 40. Suma estos números da como resultado 70. Ahora almacena este valor en la posición de salida; aquí, la posición de salida está definida como la posición 3, por lo que se sobrescribe a sí misma.

Después de la primera operación, el programa queda así:

```
1,9,10,70,
2,3,11,0,
99,30,40,50
```

Ahora avanza 4 posiciones para alcanzar el siguiente opcode, 2. Este opcode funciona como el anterior, pero se multiplican los valores en lugar de sumar. Las entradas son las posiciones 3 y 11; estas posiciones contienen 70 y 50 respectivamente. El resultado de la multiplicación es 3500; este valor se almacena en la posición 0:

```
3500,9,10,70,
2,3,11,0,
99,30,40,50
```

Avanzando 4 posiciones más se llega al código 99, deteniendo el programa.

Aquí están los estados iniciales y finales de algunos programas más:

- 1,0,0,0,99 se convierte en 2,0,0,0,99 ($1 + 1 = 2$).
- 2,3,0,3,99 se convierte en 2,3,0,6,99 ($3 * 2 = 6$).
- 2,4,4,5,99,0 se convierte en 2,4,4,5,99,9801 ($99 * 99 = 9801$).
- 1,1,1,4,99,5,6,0,99 se convierte en 30,1,1,4,2,5,6,0,99.

Parte 2

Los programas Intcode se dan como una lista de números enteros; estos valores se usan como el estado inicial de la memoria del ordenador. Cuando se ejecuta un programa Intcode, asegúrate de empezar inicializando la memoria con los valores iniciales del programa.

Una posición en la memoria se llama una dirección (por ejemplo, el primer valor en la memoria está en la "dirección 0").

Los opcodes (como el 1, 2 o 99) marcan el comienzo de una instrucción. Los valores utilizados inmediatamente después de un opcode, si lo hay, se llaman los parámetros. Por ejemplo, en la instrucción 1,2,3,4:

- 1 es el opcode;
- 2, 3 y 4 son los parámetros.

La instrucción 99 contiene sólo un opcode y no tiene parámetros. La dirección a la instrucción actual se llama puntero de instrucción y comienza en 0. Cuando una instrucción termina, el puntero de la instrucción aumenta en el número de valores de la instrucción y esto es siempre 4 (1 opcode + 3 parámetros) para las operaciones de suma y multiplicación. "Con la terminología explicada, estamos listos para proceder. Para completar la entrada gravitacional, necesitas determinar qué par de entradas producen la salida **19690720**".

Las entradas deben seguir siendo proporcionadas al programa reemplazando los valores en las direcciones 1 y 2. En este programa, el valor colocado en la **dirección 1** se llama el **sustantivo**, y el valor colocado en la **dirección 2** es llamado el **verbo**. Cada uno de los dos valores de entrada estará entre **0** y **99** ambos incluidos.

Una vez que el programa se ha detenido, su resultado estará disponible en la dirección 0, como antes. Cada vez que pruebes un par de entradas, asegúrate de que debes reiniciar la memoria del ordenador a los valores por defecto del programa (no utilices valores de una ejecución anterior).

Encuentra el **sustantivo** y el **verbo** de entrada que hacen que el programa produzca la salida **19690720**. ¿Cuáles serían el sustantivo y el verbo que producen dicha salida?

Respuesta:

Parte 3

¡Oh no! Los elfos no saben controlar los programas Intcode por consola... necesitamos generar una interfaz de usuario para gestionar nuestros programas Intcode de una forma visual para ellos.

Dicha interfaz de usuario debe constar de:

- Una ventana con opción a cerrar la aplicación o minimizarla
- Parámetros
 - **Programa:** Ruta al fichero IntCode
 - **Sustantivo:** Entero (valor de la posición 1 en el programa)
 - **Verbo:** Entero (valor de la posición 2 en el programa)
 - **Resultado:** Entero (valor en la posición 0 en el programa)
- Un botón “Calcular”
 - **Cálculo 1:** Si **Programa**, **Sustantivo** y **Verbo** están especificados. Calcula y muestra el resultado de la posición 0 en el parámetro **Resultado**
 - **Cálculo 2:** Si **Programa** y **Resultado** están especificados. Calcula el **sustantivo y verbo** que producen dicho resultado y los muestra en sus respectivos campos. Sustantivo y verbo siempre deberían estar entre 0 y 999. Si no encuentra ningún resultado posible, se debe mostrar un mensaje indicándolo.
 - *NOTA: En ambos casos, se debe mostrar una ventana de error si se produce algún comportamiento indeseado.*
- Cosas que se tendrán en cuenta:
 - Cálculo asíncrono para no bloquear la interfaz de usuario
 - Cálculo en paralelo para obtener los resultados más rápidamente
 - Se aceptan múltiples implementaciones en paralelo (threading, tasks u otros)
 - Comparación del tiempo en ejecutar las diferentes implementaciones en paralelo vs implementación con 1 único hilo para el **Cálculo 2**
 - ¿Y si **Sustantivo** y **Verbo** pudieran tener valores entre 0 y 9999999?