EXAMEN DE PRÁCTICAS

Convocatoria extraordinaria

ARQUITECTURA DE COMPUTADORES

1º Robótica Software, URJC

16 de Junio de 2021

AVISO: Asegúrate que tus programas **cumplen** con los siguientes criterios. Si no se cumple alguno de ellos la **nota será de 0** en ese apartado

- **Cumplimiento de especificaciones**. Se deben cumplir las especificaciones indicadas en el enunciado: nombres de funciones, nombres de archivos, , parámetros de las funciones, funcionalidad, etc. Compruébalo antes de entregar el examen
- **Respetar el convenio**. Resuelve las preguntas **sin violar** el convenio del uso de registros (ABI del RISC-V)
- **Sin errores en tiempo de ejecución** (Runtime errors). Tus programas no deben generar excepciones al ejecutarse
- **Sin errores al ensamblar**. Los ficheros entregados NO deben dar errores al ensamblarlos. Si una función la has dejado a medio hacer, asegúrate que al menos se ensambla sin errores

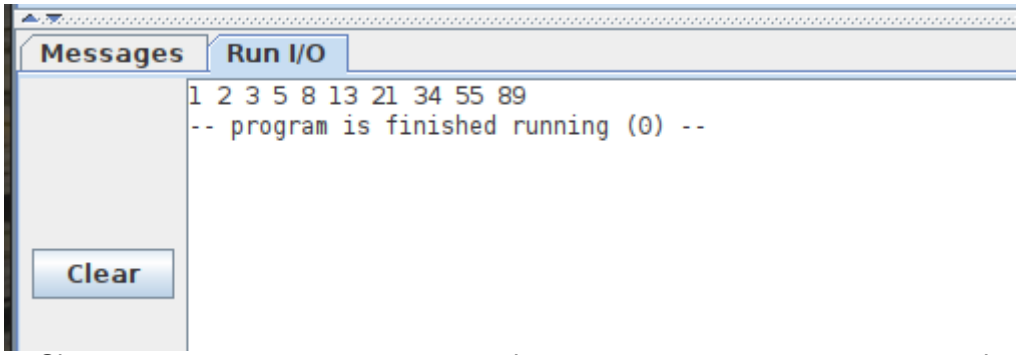
PARTE I

La sucesión de Fibonacci se caracteriza porque su término n, que llamamos F(n), se obtiene a partir de la **suma** de los dos términos anteriores: F(n-1) y F(n-2). El punto de partida son los valores F(0)=0, y F(1)=1. Con esto se genera la conocida serie: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34… Nuestro jefe de proyecto necesita que le implementemos la función **fib(a,b)** que calcula el siguiente término de la sucesión de fibonacci a partir de los dos anteriores, dados como parámetros. También necesita que implementemos un programa principal que pruebe esta función. Las **ESPECIFICACIONES** son las siguientes:

- *int fib(a,b)*: Función que calcula el **siguiente término de fibonacci** a partir de los **dos anteriores**, *a* y *b*, que se reciben como parámetros. Tiene por tanto **2 parámetros de entrada.** La función devuelve como salida el siguiente término (Tiene por tanto 1 parámetro de salida). Esta función debe suponer que tanto *a* como *b* son **términos correctos**. Es decir, que no es responsabilidad de esta función determinar si *a* y *b* son términos válidos de Fibonacci o si son consecutivos. Se asume que son correctos y se obtiene a partir de ellos el siguiente término, y se devuelve como salida de la función. Así, por ejemplo, fib(13,21) da como resultado 34

- *test_fib.s*: Programa principal para comprobar la función fib(). Al ejecutar este programa se debe mostrar en la consola la salida de los N primeros términos de la serie de fibonacci (sin contar el 0 y 1 iniciales). Este número **N** debe ser una constante definida en el código, cuyo valor por defecto será de 10. El programa **deberá tener un bucle** que se repita N veces y desde el que se

llame a la función fib() con los parámetros correctos para obtener el siguiente término de fibonacci. Los diferentes términos obtenidos se imprimen en la consola, separados mediante un espacio. Al ejecutarlo con el valor N = 10, la salida por la consola deberá ser la siguiente:

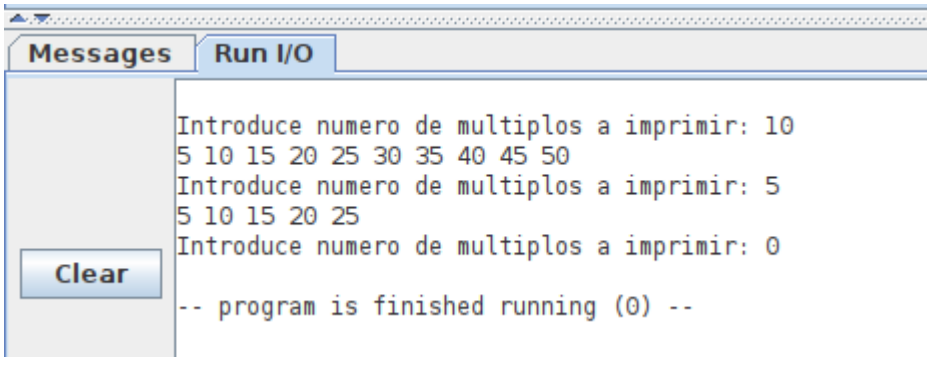


Si el valor constantes relacionadas con el acceso a los **servicios del sistema operativo** se deberán situar en el **fichero so.s**

PARTE II. Como parte de otro proyecto, hemos recibido el código hecho por uno de los ingenieros del equipo. En el fichero **mul5.s** se encuentra implementada la función mul5(x), que devuelve el valor 5 * x (multiplica x por 5). En el **fichero test_mul5.s** se encuentra implementado un programa principal que imprime en la consola los 20 primeros múltiplos de cinco: 5, 10, 15, 20, 25….100

Nuestro jefe de proyecto necesita que convirtamos el programa test_mult5.s en la **función print_mult5(n)** que imprime en la consola los primeros n múltiplos de 5. Esta función NO se debe escribir desde cero, sino que tenemos que reutilizar las partes indicadas en los comentarios. El resto de instrucciones se pueden eliminar y sustituir por otras. Para comprobar esta nueva función tenemos que hacer el programa **main.s**. Las **ESPECIFICACIONES** se muestran a continuación

- *void print_mult5(n)*. Esta función tiene un único parámetro de entrada y ninguno de salida. Imprime en la consola los n primeros múltiplos de 5, separados por un espacio. Se debe implementar a partir del código disponible en el programa test_mult5.s reutilizando el código que está comentado.
- Programa *main.s.* Programa de prueba de la función print_mult5(n). Debe pedir al usuario que introduzca el número de múltiplos de 5 que se quieren mostrar en la consola. Este número se utiliza como parámetro para llamar a la función print_mult5(n). Este proceso se repite en un bucle hasta que el usuario introduzca el valor 0, en cuyo caso el programa termina (sin imprimir nada adicional). En esta figura se muestra un pantallazo de su funcionamiento. El usuario ha introducido primero el valor 10, luego el 5 y finalmente el 0:



Se pide:

PARTE I:

1. Implementar la función **fib*()*** en el fichero **fib.s** (2 puntos)

2. Implementar el programa de prueba **test_fib.s.** Las constantes de los servicios del sistema operativo deben estar en el fichero **so.s**. (3 puntos)

PARTE II:

3. Implementar la función **print_mul5()** en el fichero **print_mul5.s**. Define todas las constantes que necesites dentro de ese fichero (No uses ningún fichero externo) (2.5 puntos)

4. Implementar el programa principal **main.s** (Define todas las constantes que necesites en el propio fichero, sin usar ningún fichero externo) (2.5 puntos)

NOTA:

Todos tus programas deben funcionar sólo con los ficheros proporcionados en las plantillas (no puedes añadir ninguno adicional). Los **ficheros a entregar** son: fib.s, test_fib.s, so.s, print_mult5.s y main.s (Envíalos por aula virtual, sin comprimir en un ZIP)

NOTA 2:

¡Y recuerda! Tienes que implementar lo pedido **sin violar** el convenio ABI del RISCV