Rot13 o rotar en 13 lugares es un mecanismo simple de cifrado que reemplaza una letra con la letra que viene 13 lugares después en el alfabeto latino. Lea el artículo dedicado en la wikipedia. Por ejemplo "pedro, juan, diego" se cifra como "crqeb, whna, qvrtb".

La función *sort* está programada en C y en assembler Risc-V en los archivos *sort-c.c* y *sort-rv.s* respectivamente. Esta función ordena lexicográficamente un arreglo de strings usando un algoritmo ridículamente ineficiente. En *sort-rv.s*, el código equivalente en C está comentado, mostrando la ubicación de las variables en los registros.

El encabezado de la función es: *void sort(char *a[ ], int n);*

En el arreglo de strings *a* vienen *n* nombres de personas como *"pedro", "juan", "diego"*. **Por simplicidad, considere que en el string no vienen letras mayúsculas, letras acentuadas o la ñ.** Sí pueden venir números o símbolos como * / ( ) etc. que no están cifrados.

El archivo *sort-rv-rot13.s* es una copia de *sort-rv.s* y *sort-c-rot13.c* una copia de *sort-c.c*. Modifique la función *sort* en *sort-rv-rot13.s* y la función *strCmp* en *sort-c-rot13.c* de modo que se ordene el arreglo de strings cifrados con rot13 ascendentemente. Es decir que antes de comparar letras, debe decifrarlas primero. El cuadro de la derecha muestra el ordenamiento ascendente normal versus el ordenamiento solicitado. Al descifrar la columna ordenada normalmente se obtiene incorrectamente pedro, diego y juan. Con el orden solicidado, después de descifrar se obtiene correctamente diego, juan, pedro, es decir ordenados alfabéticamente.

Orden ascendente normal	*Orden solicitado*
crqeb qvrtb whna	qvrtb whna crqeb

Instrucciones

Baje *t5.zip* de U-cursos y descomprímalo. Contiene el *Makefile* y los archivos que necesita para hacer esta tarea. Ejecute el comando *make* sin parámetros para recibir instrucciones sobre la ubicación de los archivos que debe modificar y cómo compilar, ejecutar y depurar. En particular lea los tips para la depuración y la solución de problemas.

Restricciones

Ud. solo puede modificar en *sort-rv-rot13.s* el código que compara los elementos consecutivos. Está claramente delimitado en el archivo original. No modifique nada más. Sin esta restricción la tarea sería trivial. Además para hacer la comparación *no puede* invocar otras funciones. Está prohibido retornar de la función en su propio código. Una vez hecha la comparación, la ejecución debe continuar en la etiqueta *.decision* y el resultado de la comparación debe quedar en el registro *t1*. Si *t1>0*, los números en *p[0]* y *p[1]* están desordenados y por lo tanto se intercambiarán. Si *t1≤0* no se intercambiarán.

Ayuda

- Estudie la función *rot13* en el archivo *test-sort.c*. Le ayudará a programar la función *strCmp* de *sort-c-rot13.c*. Haga un esfuerzo en llegar a la función más pequeña, porque así será menor la cantidad de líneas en assembler que deberá programar y depurar. Por ejemplo, evite construir nuevos strings descifrados con rot13. Descifre indifidualmente cada caracter y compárelos de inmediato. Pruebe *sort-c-rot13.c* ejecutando el comando: *make sort-c-rot13.run* (*make sort-c-rot13.ddd* para depurar)
- Una vez que pase exitosamente la prueba de *sort-c-rot13.c*, ejecute el comando *make sort-c-rot13.s* y luego estudie la función *strCmp* en el archivo *sort-c-rot13.s*. Podrá copiar gran parte de ese código en assembler en la parte delimitada en *sort-rv-rot13.s* para completar su tarea. Pruebe su tarea con: *make sort-rv-rot13.run*
- Revise si el toolchain para compilar y ejecutar programas para RiscV está instalado en */opt/riscv*. Si no encuentra ese directorio, descargue el archivo *riscv.tgz* (*riscv-arm.tgz* para Arm) de esta carpeta de google drive e instale el *toolchain* para RiscV con estos comandos:

```
cd ... directorio en donde descargaron riscv.tgz ...
sudo bash
cat riscv.tgz | ( cd / ; tar zxvf - )
```

- En la clase auxiliar del viernes 19 de mayo se estudió la solución de una tarea similar de un semestre pasado.

Entrega

Entregue por medio de U-cursos el archivo ***rot13.zip*** generado con el comando *make zip*. Este comando verifica que su tarea funcione correctamente y luego genera *rot13.zip* con los archivos *sort-c-rot13.c*, *sort-rv-rot13.s* y *resultados.txt* con la ejecución de la tarea. **Recuerde descargar de u-cursos lo que entregó, descargar nuevamente los archivos adjuntos** y vuelva a probar la tarea tal cual como la entregó. Esto es para evitar que Ud. reciba un 1.0 en su tarea porque entregó los archivos equivocados. Créame, sucede a menudo por ahorrarse esta verificacion. Su tarea debe ordenar correctamente, si no será rechazada. Se descontará medio punto por día de atraso (excluyendo sábados, domingos, festivos o vacaciones).