

# CSCE 231/2303 – Computer Organization and Assembly Language Programming – Summer 2020

## Homework 2

Using RARS, develop:

- 1) [10%] An application that finds a sum of the odd numbers in a given range of values. The application must prompt the user to enter the first value of the range, then the last value of the range. For example, if the user enters 2, and 10 for these two arguments, the output will be 24 (3+5+7+9).
- 2) [10%] An application that prompts the user to enter a string then prints out a reversed version of that string.
- 3) [10%] Write a program that reads a sentence from the user then prints it out with the first letter of each word being capitalized.
- 4) [25%] An application that accepts 2 unsigned integers from the user, multiplies them then prints the product on the console. Assume that the product is a 32-bit value. The multiplication must be implemented using the shift and add algorithm. You may learn about this algorithm @ [http://users.utcluj.ro/~baruch/book\\_ssce/SSCE-Shift-Mult.pdf](http://users.utcluj.ro/~baruch/book_ssce/SSCE-Shift-Mult.pdf) (the first page is enough).
- 5) [25%] A word guessing game. The application picks a random word from a list of words (a static list of 5-8 letters words in your application), displays the word to the user after replacing 3 random letters by "\*" then asks the user to guess the word. The user is allowed only 2 guesses per word (iteration). The application must ask the player whether she/he wants to play again or exits after every word (whether the word was guessed correctly or incorrectly).
- 6) [20%] A program that converts a null terminated string into a base-64 encoded string. To encode a string using base-64, every 3 characters (24 bits) are divided into 4 segments each of 6 bits (4x6=24). Each 6-bit segment is represented using an ASCII character according to the following table.

Binary	ASCII	Binary	ASCII	Binary	ASCII	Binary	ASCII
000000	A	010000	Q	100000	g	110000	w
000001	B	010001	R	100001	h	110001	x
000010	C	010010	S	100010	i	110010	y
000011	D	010011	T	100011	j	110011	z
000100	E	010100	U	100100	k	110100	0
000101	F	010101	V	100101	l	110101	1
000110	G	010110	W	100110	m	110110	2
000111	H	010111	X	100111	n	110111	3
001000	I	011000	Y	101000	o	111000	4
001001	J	011001	Z	101001	p	111001	5
001010	K	011010	a	101010	q	111010	6
001011	L	011011	b	101011	r	111011	7
001100	M	011100	c	101100	s	111100	8
001101	N	011101	d	101101	t	111101	9
001110	O	011110	e	101110	u	111110	+
001111	P	011111	f	101111	v	111111	/

### Guidelines:

- You can only use the instructions discussed in the lectures.
- The submission must be a single zip file that contains the six programs (name them: 1.s, 2.s, ... 6.s). Add to the zip file a text file (call it readme.txt) that contains any information you want to share about your submission. Any unresolved bug(s) or logical issues in your programs must be stated in the readme.txt file (if any).
- You may use any of the environment calls listed @ <https://github.com/TheThirdOne/rars/wiki/Environment-Calls>
- If any of your programs gives errors while assembling it using RARS, you will receive 0 for that program.
- Deadline: Thursday June 21<sup>st</sup> before 11:59PM (online through BB) – hard deadline.