

PREGUNTA 1 (30 puntos):

Suponiendo una cinta infinita y dado el formato para la programación de una máquina de Turing como:

<estado actual> <lectura> <escritura> <movimiento en la cinta> <estado siguiente>

utiliza 5 símbolos ("dígitos"): 0, 1, 2, 3, 4.

en base Quinaria en la cinta deter

Suponiendo una cinta infinita y dado el formato para la programación:

<estado actual> <lectura> <escritura> <movimiento en la cinta>

La base Quinaria (Base 5), es una base numérica que utiliza 5 símbolos ("digitos"): 0, 1, 2, 3, 4.

Escriba el programa para una máquina de Turing que dado un número en base Quinaria en la cinta determine el símbolo ("dígito") más grande que compone al número.

Al final del programa en la cinta solo debe quedar el valor del símbolo ganador.

- Parte del estado Q0.
- El cursor de la máquina se encuentra en el dígito más a la izquierda del número.
- Utilice guion bajo para representar el espacio.
- La máquina termina al encontrar el símbolo al final de la sentencia "!".
- Los movimientos válidos son R para derecha y L para izquierda.

			1	2	0	1	2	1	1						2							
			1	2	4	1	0	2	1	1					4							

SERIE S/S
1 - 2024
30/Abril/2023

PEP 1 - CÁTEDRA
FACULTAD DE INGENIERÍA
Departamento de Ingeniería Informática
Métodos de Programación



UNIVERSIDAD
DE SANTIAGO
DE CHILE

PREGUNTA 2 (30 puntos):

Dadas las operaciones para una máquina de registros:

$L_a R^+_x \rightarrow L_b$: La etiqueta L_a suma 1 en el registro R_x y se pasa a la etiqueta L_b .

$L_a R^-_x \rightarrow L_b, L_c$: La etiqueta L_a revisa si se puede restar 1 al registro R_x :

- Si es posible, resta 1 al registro R_x y se pasa a la etiqueta L_b .
- Si no es posible, se pasa a la etiqueta L_c .

$L_a \text{ Halt}$: La etiqueta L_a indica que el programa terminó.

Se le hace entrega de un número entero positivo en el registro $R1$, debe indicar si el último dígito de este número es un número primo o no. Para esto deje en el registro $R0$ un 1 si el último dígito del número en $R1$ es primo o un 0 en caso contrario.

Recuerde que:

- Los números primos son aquellos números naturales que tienen exactamente 2 divisores (1 y sí mismo).
- El primer número primo es el número 2.

Inicio						Resultado					
0	12345	0	0	...	0	1	?	?	?	...	?
R0	R1	R2	R3	...	Rn	R0	R1	R2	R3	...	Rn
0	65168456	0	0	...	0	0	?	?	?	...	?
R0	R1	R2	R3	...	Rn	R0	R1	R2	R3	...	Rn
0	978516898413	0	0	...	0	1	?	?	?	...	?
R0	R1	R2	R3	...	Rn	R0	R1	R2	R3	...	Rn

SERIE S/S
1 - 2024
30/Abril/2023

PEP I - CÁTEDRA
FACULTAD DE INGENIERÍA
Departamento de Ingeniería Informática
Métodos de Programación



UNIVERSIDAD
DE SANTIAGO
DE CHILE

PREGUNTA 3 (30 puntos):

En matemáticas, se puede aproximar el valor del logaritmo natural de x ($\ln(x)$) a través de sumatorias. Para esto se utiliza la siguiente expresión:

$$\ln(x) = 2 \sum_{n=0}^{\infty} \frac{1}{2n+1} \left(\frac{x-1}{x+1} \right)^{2n+1}$$

Suponiendo que la máquina de stack soporta las siguientes instrucciones:

- **PUSH(a)**: inserta en el stack el número a .
- **POP**: obtiene un elemento del stack.
- **SUM**: suma dos elementos del stack.
- **RES**: resta dos elementos del stack.
- **MUL**: multiplica dos elementos del stack.
- **DIV**: divide dos elementos del stack.
- **MOD**: obtiene el resto de la división de dos elementos del stack.
- **EXP**: realiza la potencia de dos elementos del stack.
- **RAI**: realiza la raíz cuadrada de un elemento del stack.
- **FAC**: obtiene el factorial de un elemento del stack.

Escriba las sentencias necesarias para que una máquina de stack obtenga el valor de $\ln(55)$ usando la expresión anterior. Utilice los 2 primeros términos de la sucesión para realizar la aproximación sin realizar operaciones matemáticas de forma manual.

$n=0 \quad n=1$

Hoja 6

Selen Acido

(A)



PEP 1 - CÁTEDRA
FACULTAD DE INGENIERIA
Departamento de Ingeniería Informática
Métodos de Programación

SERIE S/S
2 - 2023
10/Octubre/2023

PEP 1 - CÁTEDRA
FACULTAD DE INGENIERIA
Departamento de Ingeniería Informática
Métodos de Programación

impar

PREGUNTA 1 (30 puntos):

Suponiendo una cinta infinita y dado el formato para la programación de una máquina de Turing
<estado actual> <lectura> <escritura> <movimiento en la cinta> <estado siguiente>

El bit de paridad es una forma de corroborar que la transmisión de un mensaje se ha realizado correctamente. Para esto se sacrifica un bit en la representación de mensajes y se utiliza este para la validación. En el caso de los paquetes de 8 bits se utilizan los 7 primeros bits para enviar el mensaje y el octavo es el bit de paridad, en caso de que el mensaje tenga una cantidad impar de unos el bit de paridad se marca como 1 y en caso de que la cantidad de unos del mensaje sea par se marca el bit de paridad como cero. Esto se conoce como paridad par, ya que el paquete (los 8 bits), siempre termina con una cantidad par de unos. En caso de que el bit de paridad no corresponda, se detecta un error en la transmisión y se solicita que el mensaje se reenvíe.

Se le solicita que valide un paquete de 8 bits donde el último bit es el bit de paridad. El programa debe borrar el número y dejar en la cinta el mensaje "OK" si el bit de paridad es correcto o "ERROR" en caso de que no lo sea.

Para esto considere que:

- Parte del estado Q0.
- El cursor de la máquina se encuentra en el dígito más a la izquierda del número.
- Utilice guion bajo para representar el espacio.
- La máquina termina al encontrar el símbolo al final de la sentencia "I".

Ejemplos:

Par ultimo debe ser 0
↑ impar un 1

	Entrada								Salida			
Par 0	1	0	0	1	0	1	1	0	O	K		
Par 1	1	1	1	1	1	1	1	1	O	K		
	0	0	0	0	0	0	0	1	E	R	R	O
	1	0	1	0	1	0	1	0	O	K		
	0	0	1	1	0	0	0	1	E	R	R	O
	1	1	1	1	0	1	1	1	E	R	R	O

PREGUNTA 2 (30 puntos):

Dadas las operaciones para una máquina de registros:

$L_a R^+_x \rightarrow L_b$: La etiqueta L_a suma 1 en el registro R_x y se pasa a la etiqueta L_b .

$L_a R^-_x \rightarrow L_b, L_c$: La etiqueta L_a revisa si se puede restar 1 al registro R_x :

- Si es posible, resta 1 al registro R_x y se pasa a la etiqueta L_b .
- Si no es posible, se pasa a la etiqueta L_c .

$L_a \text{ Halt}$: La etiqueta L_a indica que el programa terminó.

Conjetura de Collatz

La conjetura de Collatz es un problema abierto el cuál postula que dado un número entero positivo al realizar una serie de operaciones siempre se converge. Las operaciones que se realizan son:

- Si N es impar se calcula $3N+1$
- Si N es par se calcula $N/2$

Si el número resultante es distinto de 1 se vuelven a realizar las operaciones. El problema está comprobado para los primeros 2^{68} números.

Ejemplo:

N	Análisis	Resultado
6	Par	$6/2 = 3$
3	Impar	$3 \times 3 + 1 = 10$
10	Par	$10/2 = 5$
5	Impar	$3 \times 5 + 1 = 16$
16	Par	$16/2 = 8$
8	Par	$8/2 = 4$
4	Par	$4/2 = 2$
2	Par	$2/2 = 1$
1	Fin	

Dado un número entero positivo menor a 2^{68} en el registro $R1$, compruebe si la conjetura de collatz se cumple para dicho número dejando un 1 en el registro $R0$ si converge a 1.

En caso de utilizar registros extras, todos deben comenzar con el valor cero.



PREGUNTA 3 (30 puntos):

En matemáticas, la Serie de Maclaurin para el coseno es una representación algebraica fundamental que permite expresar la función coseno en términos de una sumatoria infinita. Fue desarrollada por el matemático escocés Colin Maclaurin y es una herramienta esencial en cálculo y análisis matemático. La Serie de Maclaurin para el coseno permite aproximaciones precisas del coseno de un ángulo en función de los términos de una expansión infinita.

$$\cos(x) = 1 \ominus \frac{x^2}{2!} + \frac{x^4}{4!} \ominus \frac{x^6}{6!} + \dots = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!}$$

Suponiendo que la máquina de stack soporta las siguientes instrucciones:

- **PUSH(a)**: inserta en el stack el número a.
- **POP**: obtiene un elemento del stack.
- **SUM**: suma dos elementos del stack.
- **RES**: resta dos elementos del stack.
- **MUL**: multiplica dos elementos del stack.
- **DIV**: divide dos elementos del stack.
- **MOD**: obtiene el resto de la división de dos elementos del stack.
- **EXP**: realiza la potencia de dos elementos del stack.
- **RAI**: realiza la raíz cuadrada de un elemento del stack.
- **FAC**: obtiene el factorial de un elemento del stack.

Escriba las sentencias necesarias para que una máquina de stack obtenga el valor de $\cos(1.57)$ usando la expresión anterior. Utilice los 4 primeros términos de la sucesión para realizar la aproximación sin realizar operaciones matemáticas de forma manual.

$$1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} \quad x = 1,57$$
$$1 - \frac{(1,57)^2}{2 \cdot 1} + \frac{(1,57)^4}{2 \cdot 3 \cdot 4} - \frac{(1,57)^6}{2 \cdot 3 \cdot 4 \cdot 5 \cdot 6}$$

SERIE S/S
1 - 2023
11/Abril/2023

PEP 1 - CÁTEDRA
FACULTAD DE INGENIERIA
Departamento de Ingeniería Informática
Métodos de Programación



UNIVERSIDAD
DE SANTIAGO
DE CHILE

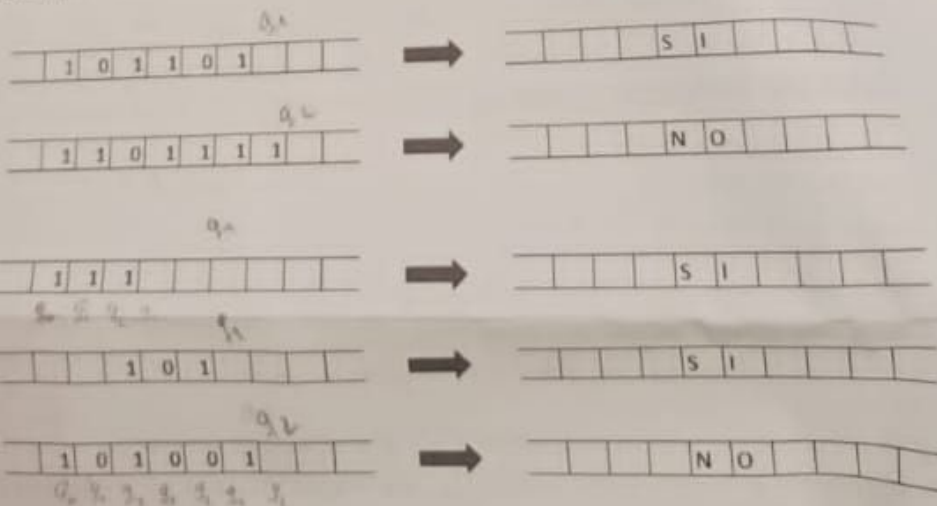
PREGUNTA 1 (30 puntos):

Suponiendo una cinta infinita y dado el formato para la programación de una máquina de Turing como:
<estado actual> <lectura> <escritura> <movimiento en la cinta> <estado siguiente>

implementar el programa de una máquina de Turing que, dado un número binario escrito en la cinta, borre dicho número e identifique si es o no un número capicúa. Los números capicúas son aquellos se pueden leer de igual forma de izquierda a derecha que de derecha a izquierda. Al final del ejercicio solo debe quedar la palabra "SI" o "NO" según corresponda en la cinta. Para esto considere que:

- Parte del estado Q_0 .
- El cursor de la máquina se encuentra en el dígito más a la izquierda del número.
- Utilice guion bajo para representar el espacio.
- La máquina termina al encontrar el símbolo "!".

Ejemplos:



Capicua = palindromo de los numeros

SERIE S/S
1 - 2023
11/Abril/2023

PEP 1 - CÁTEDRA
FACULTAD DE INGENIERÍA
Departamento de Ingeniería Informática
Métodos de Programación



UNIVERSIDAD
DE SANTIAGO
DE CHILE

PREGUNTA 3 (30 puntos):

Existen diversas formas de calcular el valor de π , algunas de ellas implican series numéricas. Una de ellas viene del problema de Basilea, el cuál busca sumar los inversos de los cuadrados de los números naturales.

$$\sum_{i=1}^{\infty} \frac{1}{i^2} = \frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \frac{1}{5^2} + \frac{1}{6^2} + \dots$$

El problema de Basilea determina que la serie numérica cuando tiende al infinito converge al valor $\frac{\pi^2}{6}$.

$$\frac{\pi^2}{6} = \frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \frac{1}{5^2} + \frac{1}{6^2} + \dots$$

Por lo que puede obtenerse el valor del π despenjándolo:

$$\pi = \sqrt{6 \left(\frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \frac{1}{5^2} + \frac{1}{6^2} + \dots \right)}$$

Suponiendo que la máquina de stack soporta las siguientes instrucciones:

- PUSH(a): inserta en el stack el número a.
- POP: obtiene un elemento del stack.
- SUM: suma dos elementos del stack.
- RES: resta dos elementos del stack.
- MUL: multiplica dos elementos del stack.
- DIV: divide dos elementos del stack.
- MOD: obtiene el resto de la división de dos elementos del stack.
- EXP: realiza la potencia de dos elementos del stack.
- RAÍ: realiza la raíz cuadrada de un elemento del stack.

Escriba las sentencias necesarias para que una máquina de stack obtenga el valor de π para $n=4$ usando la expresión anterior y que calcule el error porcentual al compararlo con el valor $\pi=3.14159$ sin realizar operaciones matemáticas de forma manual.

$$\text{Error Porcentual} = \frac{|\text{ValorObtenido} - \text{ValorEsperado}|}{\text{ValorEsperado}} \times 100$$

SERIE S/S
1 - 2023
11/Abril/2023

PEP 1 - CÁTEDRA
FACULTAD DE INGENIERÍA
Departamento de Ingeniería Informática
Métodos de Programación



UNIVERSIDAD
DE SANTIAGO
DE CHILE

PREGUNTA 2 (30 puntos):

Dadas las operaciones para una máquina de registros:

$L_a R^+_x \rightarrow L_b$: La etiqueta L_a suma 1 en el registro R_x y se pasa a la etiqueta L_b .

$L_a R^-_x \rightarrow L_b, L_c$: La etiqueta L_a revisa si se puede restar 1 al registro R_x :

- Si es posible, resta 1 al registro R_x y se pasa a la etiqueta L_b .
- Si no es posible, se pasa a la etiqueta L_c .

$L_a \text{ Halt}$: La etiqueta L_a indica que el programa terminó.

Dado 3 números enteros positivos en los registros R0, R1 y R2, determine cuál es el orden de los números. El resultado debe quedar en los registros R0, R1 y R2, dejando un 1, un 2 y un 3. Donde 1 representa el número menor, y el 3 el número mayor.

En caso de utilizar registros extras, todos deben comenzar con el valor cero.

En caso que dos o más números tengan el mismo valor, puede marcar cualquiera de ellos como el más grande de ellos.

Ejemplos:

R0	R1	R2
5	10	3



R0	R1	R2
2	3	1

R0	R1	R2
10	20	30



R0	R1	R2
1	2	3

R0	R1	R2
10	7	2



R0	R1	R2
3	2	1

Suponiendo una cinta infinita y dado el formato para la programación de una máquina de Turing como:

<estado actual> <lectura> <escritura> <movimiento en la cinta> <estado siguiente>

En la cinta se ingresan dos grupos de ceros separados por un espacio al medio. Su trabajo es determinar si ambos grupos tienen la misma cantidad de ceros. En caso que tengan la misma cantidad escriba la letra "Y" en cualquier lugar de la cinta, en caso contrario escriba la letra "N" en cualquier lugar de la cinta.

- Parte del estado Q0.
- El cursor de la máquina se encuentra en el dígito más a la izquierda del número.
- Utilice guion bajo para representar el espacio.
- La máquina termina al encontrar el símbolo al final de la sentencia “!”.
- Los movimientos válidos son R para derecha y L para izquierda.

Ejemplos:



		0	0	0		0	0	0								?	?	?	?	?	?	?	?	?				
		0	0	0	0		0	0	0	0	0	0				?	N	?	?	?	?	?	?					



PREGUNTA 2 (30 puntos):

Dadas las operaciones para una máquina de registros:

$L_a R_x^+ \rightarrow L_b$: La etiqueta L_a suma 1 en el registro R_x y se pasa a la etiqueta L_b .

$L_a R_x^- \rightarrow L_b, L_c$: La etiqueta L_a revisa si se puede restar 1 al registro R_x :

- Si es posible, resta 1 al registro R_x y se pasa a la etiqueta L_b .
- Si no es posible, se pasa a la etiqueta L_c .

$L_a \text{ Halt}$: La etiqueta L_a indica que el programa terminó.

Se le hace entrega de dos números enteros positivos en los registros $R1$ y $R2$, debe indicar en $R0$ el resto de la división de $R1/R2$.

Inicio						Resultado					
0	12	2	0	...	0	0	?	?	?	...	?
$R0$	$R1$	$R2$	$R3$...	Rn	$R0$	$R1$	$R2$	$R3$...	Rn
0	6	4	0	...	0	2	?	?	?	...	?
$R0$	$R1$	$R2$	$R3$...	Rn	$R0$	$R1$	$R2$	$R3$...	Rn
0	10	13	0	...	0	10	?	?	?	...	?
$R0$	$R1$	$R2$	$R3$...	Rn	$R0$	$R1$	$R2$	$R3$...	Rn

$$6:4 = 6-1$$

$$12:2 = 12 - (2 \cdot 6) = 12 - 12 = 0$$

PREGUNTA 3 (30 puntos):

El capitán Barbazulcielitolindo, el pirata más temido de la web, finalmente ha decidido retirarse y dejar atrás la vida de saqueos. Pero antes de retirarse a su isla paradisíaca, ha decidido invertir su botín de 1000 doblones de ethereum en una cuenta bancaria para asegurarse una jubilación tranquila. Barbazulcielitolindo encuentra una oferta tentadora en Kraken Bank, donde le ofrecen un interés compuesto del 5% anual. Sin embargo, como buen pirata, no le gusta esperar mucho, así que decide que el interés se componga 4 veces al año.

¿Cuánto dinero tendrá Barbazulcielitolindo en su cofre al cabo de 10 años si deja el oro en el banco?

Fórmula del interés compuesto:

$$A = P \left(1 + \frac{r}{n} \right)^{nt}$$

Donde:

- A es el monto final (capital más intereses).
- P es el capital inicial (inversión o préstamo). ≈ 1000
- r es la tasa de interés anual (expresada en decimal, por ejemplo, 5% sería 0.05). $\approx 0,05$
- n es el número de veces que el interés se compone por año. ≈ 4
- t es el tiempo que dura la inversión o el préstamo, en años. ≈ 10

Suponiendo que la máquina de stack soporta las siguientes instrucciones:

- PUSH(a): inserta en el stack el número a.
- POP: obtiene un elemento del stack.
- SUM: suma dos elementos del stack.
- RES: resta dos elementos del stack.
- MUL: multiplica dos elementos del stack.
- DIV: divide dos elementos del stack.
- MOD: obtiene el resto de la división de dos elementos del stack.
- EXP: realiza la potencia de dos elementos del stack.
- RAI: realiza la raíz cuadrada de un elemento del stack.
- FAC: obtiene el factorial de un elemento del stack.

Escriba las sentencias necesarias para que una máquina de stack obtenga la cantidad de doblones que tendrá en capitán sin realizar operaciones matemáticas de forma manual.