

## Algoritmos y Estructuras de Datos

Cursada 1er cuatrimestre 2021 - 1er parcial - 7 de Mayo de 2021

Nombre:..... Legajo:.....

E-mail:.....

Nota Ejercicio 1	Nota Ejercicio 2	Nota Final

1)

- a) Dado el siguiente vector, realice manualmente todos los pasos necesarios para ordenarlo de menor a mayor utilizando el método Quicksort.

10	1	12	3	14	5	16	7	51	9	22	55	26	57
----	---	----	---	----	---	----	---	----	---	----	----	----	----

- b) Dado el siguiente código indique que realiza.

```
void fun(Queue *Q)
{
    Stack S;

    while (!isEmpty(Q))
    {
        push(&S, deQueue(Q));
    }
    while (!isEmpty(&S))
    {
        enqueue(Q, pop(&S));
    }
}
```

1. Remueve el último elemento de Q.
2. Mantiene a Q igual que antes de llamar a la función.
3. Vacía Q.
4. Invierte Q.

- c) ¿Cuántas filas se necesitan para implementar una pila?

1. 1
2. 2
3. 3
4. 4

- d) A continuación se presenta un código que toma una pila S como argumento. ¿Qué hace?

```
void fun(int n)
{
    while (n > 0)
    {
        push(&S, n%2);
        n = n/2;
    }
}
```

## Algoritmos y Estructuras de Datos

Cursada 1er cuatrimestre 2021 - 1er parcial - 7 de Mayo de 2021

```
}  
while (!isEmpty(&S))  
    printf("%d ", pop(&S));  
}
```

1. Imprime una representación binaria de n en orden invertido.
- ☒ 2. Imprime una representación binaria de n.
3. Imprime el log n.
4. Imprime el log n en orden invertido.

e) Considere el siguiente pseudocódigo asociado a una pila.Cuál es la salida en pantalla si la entrada es “algoritmo”.

```
declare una pila de caracteres  
while ( que hayan más caracteres en la palabra por leer )  
{  
    leer character  
    push el character a la pila  
}  
while ( la pila no esté vacía )  
{  
    pop un character de la pila  
    imprimir el character en pantalla  
}
```

1. algoritmoalgoritmo
- ☒ 2. omtirogla
3. algoritmo
4. omtiroglaomtirogla

f) ¿Cuál es la estructura de datos más adecuada para chequear si una expresión algorítmica tiene sus paréntesis correctamente balanceados ?

1. Fila
- ☒ 2. Pila
3. Lista

g) ¿Los caracteres A,B,C,D,E,F,G son pusheados a una pila en orden invertido empezando por la G. La pila es popeada 5 veces y cada elemento es insertado en una fila. Dos elementos de la fila son sacados y agregados a la pila. Luego un elemento es popeado de la pila. ¿Cuál es el caracter sacado?

1. A
- ☒ 2. B
3. F
4. G

- h) Considere la función implementada. Asuma que el arreglo listA contiene n elementos ordenados de forma ascendente.

```
int ProcessArray(int *listA, int x, int n)
{
    int i, j, k;
    i = 0;
    j = n-1;
    do
    {
        k = (i+j)/2;
        if (x <= listA[k])
            j = k-1;
        if (listA[k] <= x)
            i = k+1;
    }
    while (i <= j);
    if (listA[k] == x)
        return(k);
    else
        return -1;
}
```

1. Entra en un loop infinito cuando x no se encuentra en listA
  2. Es una implementación de búsqueda binaria
  3. Siempre va a encontrar el máximo elemento en listA
  4. Retorna -1 aún cuando x está presente en listA
- i) ¿Cuál es el número de comparaciones realizadas en una búsqueda secuencial de un elemento x en un vector de longitud L en el mejor de los casos?
1. 1
  2. L/2
  3. (L+1)/2
  4. L
- j) La recursión es una metodología a partir de la cual la solución de un problema depende de:
1. Resolver instancias mayores de distintos problemas
  2. Resolver Instancias mayores del mismo problema.
  3. Resolver instancias menores del mismo problema.
  4. Resolver Instancias menores de distintos problemas.
- k) ¿En recursión la condición por la cual una función deja de llamarse a sí misma es
1. Criterio mejor
  2. Criterio peor

- 3. Criterio base
- 4. No existe una condición así

l) ¿Cuál es la salida de la siguiente función?

```
void my_recursive_function(int n)
{
    if(n == 0)
        return;
    printf("%d ",n);
    my_recursive_function(n-1);
}

int main()
{
    my_recursive_function(10);
    return 0;
}
```

- 1. 10
- 2. 1
- 3. 10 9 8 ... 1 0
- 4. 10 9 8 ... 1

m) ¿Cuál es es la aproximación progresiva de la siguiente función recursiva?

```
void my_recursive_function(int n)
{
    if(n == 0)
        return;
    printf("%d ",n);
    my_recursive_function(n-1);
}

int main()
{
    my_recursive_function(10);
    return 0;
}
```

- 1. return
- 2. printf("%d",n)
- 3. if(n==0)
- 4. my recursive function(n-1)

n) ¿Qué hace el siguiente código recursivo?

```
void my_recursive_function(int n)
{
    if(n == 0)
        return;
    printf("%d ",n);
    my_recursive_function(n-1);
}
```

```
}  
int main()  
{  
    my_recursive_function(10);  
    return 0;  
}
```

1. Imprime los números del 10 al 1
2. Imprime los números del 10 al 0
3. Imprime los números del 1 al 10
4. Imprime los número del 0 al 10

o) ¿Cuál es la salida del siguiente código?

```
void my_recursive_function(int *arr, int val, int idx, int len)  
{  
    if(idx == len)  
    {  
        printf("-1");  
        return ;  
    }  
    if(arr[idx] == val)  
    {  
        printf("%d",idx);  
        return;  
    }  
    my_recursive_function(arr,val,idx+1,len);  
}  
int main()  
{  
    int array[10] = {7, 6, 4, 3, 2, 1, 9, 5, 0, 8};  
    int value = 2;  
    int len = 10;  
    my_recursive_function(array, value, 0, len);  
    return 0;  
}
```

1. 3
2. 4
3. 5
4. 6

2) La gran mayoría de los compiladores poseen un algoritmo de verificación de expresiones matemáticas en los cuales se verifica si la misma está correctamente codificada, es decir, se identifica si los paréntesis, corchetes y llaves están balanceados. Existe un compilador que permite realizar esta verificación, aunque la expresión sea presentada en formato matricial. Dada una cierta expresión matemática generar el algoritmo que permita:

a) Mapear la expresión matricial para obtener una expresión vectorial.

## Algoritmos y Estructuras de Datos

Cursada 1er cuatrimestre 2021 - 1er parcial - 7 de Mayo de 2021

- b) Identificar si la expresión está correctamente codificada, es decir si los paréntesis, corchetes y llaves están balanceados, utilizando **una única pila** o **una única fila**. Tener en cuenta que una de las estructuras nombradas es la más adecuada para resolver el problema.
- c) Guardar los operandos en un vector llamado "vectop" y obtener **de forma recursiva** el máximo de los operandos presentes en la expresión junto con su posición.
- d) Obtener **de forma recursiva** el mínimo de los operandos presentes en la expresión junto con su posición.
- e) Programar un algoritmo de ordenamiento que tenga como argumentos el intervalo dentro del cual se encuentran los datos a ordenar (el mínimo y el máximo encontrados en los puntos anteriores) y ordene a partir de crear un vector de números enteros con tantos elementos como valores se tenga en el intervalo y de contar la repetición de cada uno de los elementos existentes. Tener en cuenta que si el valor no existe se le asigna valor cero. Luego, al recorrer el vector se tienen los elementos ordenados. Recorrerlo de forma tal de ordenarlos de **mayor a menor** y generar el vector de operandos ordenados vectord.
- f) Guardar el vector vectop en una lista.
- g) Insertar en la lista generada en el punto f) el número 7 en la posición anterior a la del máximo y el número 2 en la posición siguiente a la del mínimo.

Nota: La expresión matemática tiene un total de 21 elementos entre operandos, operadores, corchetes y llaves. Los operandos son de un sólo dígito y en todas las expresiones suman un total de 7 valores. La forma matricial de la expresión es la siguiente:

0	0	0	0	0	0	0
E1	0	0	0	0	0	0
E2	E3	0	0	0	0	0
			0	0	0	0
	E8		E10	0	0	0
					0	0
					E21	0

Para corroborar el correcto funcionamiento del algoritmo presentado probar con las siguientes expresiones:

0	0	0	0	0	0	0
---	---	---	---	---	---	---

# Algoritmos y Estructuras de Datos

Cursada 1er cuatrimestre 2021 - 1er parcial - 7 de Mayo de 2021

7	0	0	0	0	0	0
.	{	0	0	0	0	0
8	+	[	0	0	0	0
(	9	-	2	0	0	0
)	.	(	5	+	0	0
2	+	6	)	]	}	0

7	.	{	8	+	[	(	9	-	2	)	.	(	5	+	2	+	6	)	]	}
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

0	0	0	0	0	0	0
(	0	0	0	0	0	0
5	+	0	0	0	0	0
6	+	3	0	0	0	0
)	.	{	(	0	0	0
5	.	[	2	+	0	0
3	)	]	/	2	}	0

(	5	+	6	+	3	)	.	{	(	5	.	[	2	+	3	)	]	/	2	}
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

0	0	0	0	0	0	0
{	0	0	0	0	0	0
{	(	0	0	0	0	0
8	+	3	0	0	0	0
)	.	(	1	0	0	0
+	6	)	.	(	0	0
5	+	8	+	2	)	0

{	{	(	8	+	3	)	.	(	1	+	6	)	.	(	5	+	8	+	2	)
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

NOTA: Se considera entregado el examen luego de recibir la confirmación de los docentes.