	UNIVERSIDAD CATÓLICA ANDRÉS BELLO – GUAYANA Escuela de Ingeniería Informática Manual de Prácticas Laboratorio de Algoritmos y Programación II		FORMA: P-GC-01/6		
			VIGENCIA	REVISION	No.
			31-01-12	1	
			DOCUMENTO		
			VIGENCIA	REVISION	No.
			24-09-12		
CÓDIGO: LIFPAPI03					

Práctica No.: 3

Apuntadores y Memoria Dinámica

Objetivo de la práctica

Utilizar recursión en la solución de problemas, uso de diferentes tipos de datos en C, aplicar los conceptos relacionados con el uso de apuntadores y la gestión de memoria dinámica.

Desarrollo de la práctica:

Actividad 1


Dadas las siguientes declaraciones en C:

```
float x, *px, a[5];
```

Cuales de las siguientes asignaciones son correctas y cuales son incorrectas. En el caso de las correctas explique qué hacen y para las incorrectas justifique su respuesta.

```
x = *px;
*px = x;
px = &x;
&x = px;
&(x+1) = x;
&(x)+1 = x;
*(&(x+1)) = x;
*(&(x)+1) = x;
x = a;
x = a[0];
x = *(a[1]);
x = (*a)[2];
x = a[3+1];
x = a[3]+1;
x = &((a[3])+1);
x = &(a[3])+1;
x = *(&(a[3])+1);
px = a;
px = a[0];
px = &(a[4]);
```

REVISADO: NOMBRE: JESUS J. LAREZ M. FIRMA: _____ <div style="text-align: center; margin-top: 10px;">Responsable y/o Profesor de la Cátedra</div>	APROBADO: NOMBRE: MARIA CORA URDANETA FIRMA: _____ <div style="text-align: center; margin-top: 10px;">Directora de Escuela de Ing. Informática</div>
--	--

	UNIVERSIDAD CATÓLICA ANDRÉS BELLO – GUAYANA Escuela de Ingeniería Informática Manual de Prácticas Laboratorio de Algoritmos y Programación II			FORMA: P-GC-01/6		
				VIGENCIA	REVISIÓN	No.
				31-01-12	1	
				DOCUMENTO		
				VIGENCIA	REVISIÓN	No.
	24-09-12					
CÓDIGO: LIFPAPII03						

Actividad 2

Dado el siguiente programa en C:

```
#include <stdio.h>

char string1[12] =
    {'H','o','l','a',' ','M','u','n','d','o',' ','\0'};
char string2[] = "Hola Mundo";
char *string3 = "Hola Mundo";
char *ap_string[] = {"Uno","Dos","Tres"};
char abc[][6] = {"Uno","Dos","Tres"};

int main(void){
    printf("Tamaño de string1:%d\n",sizeof(string1));
    printf("Tamaño de string2:%d\n",sizeof(string2));
    printf("Tamaño de string3:%d\n",sizeof(string3));
    printf("Tamaño de ap_string:%d\n",sizeof(ap_string));
    printf("Tamaño de ABC:%d\n\n",sizeof(abc));
}
```

Se desea que usted lo compile y explique la salida.

Actividad 3

Dado las siguientes declaraciones y funciones en C:

```
typedef struct node {
    char name[20];
    int value;
    struct node *next;
} Node;

/*
 * New_item: Crea un nuevo elemento a partir
 * del nombre y un valor
 */

Node *new_item(char *name, int value){
    Node *newp;


    if((newp=(Node *)malloc(sizeof(Node)))==NULL){
        fprintf(stderr,
            "new_item: error en malloc\n");
        exit(1);
    }
    strcpy(newp->name,name);
    newp->value = value;
```

REVISADO:
NOMBRE: JESUS J. LAREZ M.
FIRMA:

Responsable y/o Profesor de la Cátedra

APROBADO:
NOMBRE: MARIA CORA URDANETA
FIRMA:

Directora de Escuela de Ing. Informática

	UNIVERSIDAD CATÓLICA ANDRÉS BELLO – GUAYANA Escuela de Ingeniería Informática Manual de Prácticas Laboratorio de Algoritmos y Programación II			FORMA: P-GC-01/6		
				VIGENCIA	REVISION	No.
				31-01-12	1	
				DOCUMENTO		
				VIGENCIA	REVISION	No.
24-09-12						
CÓDIGO: LIFPAPII03						

```

newp->next=NULL;
return newp;
}

/*
 * add_front: añade newp al frente de listp
 */

Node *add_front(Node *listp, Node *newp){
    newp->next=listp;
    return newp;
}

/*
 * add_end añade newp al final de la listp
 */

Node *add_end(Node *listp, Node *newp){
    Node *p;

    if(listp==NULL)
        return newp;
    for(p=listp;p->next!=NULL;p=p->next);
    p->next = newp;
    return listp;
}

/*
 * print: muestra los elementos en listp
 */

void print(Node *listp){
    printf("-->");
    for(;listp!=NULL;listp=listp->next)
        printf("%s:%d-->",listp->name,listp->value);
    printf("NULL\n");
}

```

Se desea que usted:


- a.-** Implemente la función reverse, que invierta la secuencia de los elementos de una lista, de forma iterativa. A continuación se muestra su prototipo:

```

// reverse: invierte una lista donde listp apunta al primer
// elemento retorna un apuntador al primer elemento de la
// lista invertida

```

REVISADO: NOMBRE: JESUS J. LAREZ M. FIRMA: _____ Responsable y/o Profesor de la Cátedra	APROBADO: NOMBRE: MARIA CORA URDANETA FIRMA: _____ Directora de Escuela de Ing. Informática
---	---

	UNIVERSIDAD CATÓLICA ANDRÉS BELLO – GUAYANA Escuela de Ingeniería Informática Manual de Prácticas Laboratorio de Algoritmos y Programación II			FORMA: P-GC-01/6		
				VIGENCIA	REVISIÓN	No.
				31-01-12	1	
				DOCUMENTO		
				VIGENCIA	REVISIÓN	No.
	24-09-12					
CÓDIGO: LIFPAPII03						

```
Node *reverse(Node *listp);
```

- b.-** Implemente la función copy, la cual crea una nueva lista que es la copia de la lista dada, su prototipo se muestra continuación:

```
// copy: copia una lista donde listp apunta al primer elemento
// de la lista a copiar y retorna un apuntador al primer
// elemento de la nueva lista
```

```
Node *copy(Node *listp);
```

- c.-** Coloque todas las funciones (las dadas y las implementadas por usted) en el archivo lista.c cree el archivo de cabecera lista.h con las definiciones y escriba un programa (prueba.c), que haga uso de las funciones previamente definidas y muestre su uso. Compile y muestre los resultados.

Asignación

- Una cadena de caracteres (string) es un vector de caracteres que termina con un carácter '\0', por lo que las funciones que manipulan strings esperan que los mismos terminen en '\0'.

- a.** Dado la siguiente función

```
void strcat (char s[], char t[]) {
    int i, j;

    i = j = 0;
    while (s[i] != '\0') {
        i++;
    }
    while ((s[i++] = t[j++]) != '\0') {
    }
}
```

Reescribala usando apuntadores. ¿Qué ventaja se tienen?

- b.** Explique que hacen las siguientes funciones:

```
int my_strlen(const char *s){
    register const char *p = s;


    while(*p) // *p != '\0'
```

REVISADO:
NOMBRE: JESUS J. LAREZ M.
FIRMA:

Responsable y/o Profesor de la Cátedra

APROBADO:
NOMBRE: MARIA CORA URDANETA
FIRMA:

Directora de Escuela de Ing. Informática

	UNIVERSIDAD CATÓLICA ANDRÉS BELLO – GUAYANA Escuela de Ingeniería Informática Manual de Prácticas Laboratorio de Algoritmos y Programación II			FORMA: P-GC-01/6		
				VIGENCIA	REVISIÓN	No.
				31-01-12	1	
				DOCUMENTO		
				VIGENCIA	REVISIÓN	No.
	24-09-12					
CÓDIGO: LIFPAPII03						

```

        p++;
    return p-s;
}

void my_strcpy(char *d, const char *f){
    while(*d++ = *f++);
}

void my_strcat(char *d, const char *f){
    my_strcpy(d+my_strlen(d), f);
}

char *my_strdump(const char *f){
    char *d;

    d=(char *)malloc(my_strlen(f)+1);
    my_strcpy(d, f);
    return d;
}

int my_substr(const char *s1, const char *s2){
    int i,v;
    const char *p1,*p2;

    v=my_strlen(s2)-my_strlen(s1);
    for(i=0;i<=v;i++)
        for(p1=s1,p2=s2+i;*p1++==*p2++;)
            if(*p1=='\0') return i;
    return -1;
}

```

c. Implemente, usando Lenguaje C, las siguientes funciones:


- a. Que retorne el número de veces que el string s1 está en el string s2

```
int my_nsubstr(const char *s1, const char *s2)
```

- b. Invertir un string, por ejemplo, convierte a "abc" en "cba", trabaja sobre el mismo espacio apuntado por s.

```
void my_strrev(char *s)
```

REVISADO: NOMBRE: JESUS J. LAREZ M. FIRMA: _____ Responsable y/o Profesor de la Cátedra	APROBADO: NOMBRE: MARIA CORA URDANETA FIRMA: _____ Directora de Escuela de Ing. Informática
---	---

	UNIVERSIDAD CATÓLICA ANDRÉS BELLO – GUAYANA Escuela de Ingeniería Informática Manual de Prácticas Laboratorio de Algoritmos y Programación II		FORMA: P-GC-01/6		
			VIGENCIA	REVISION	No.
			31 - 01 - 12	1	
			DOCUMENTO		
			VIGENCIA	REVISION	No.
		CÓDIGO: LIFPAPII03			

2. Apoyándose en las funciones previamente implementadas, para el manejo de listas, se desea que usted implemente las siguientes funciones:

- a. **reverse**, es su versión recursiva. Por ejemplo, dada (A, B, C) retorna la lista (C, B, A)
- b. **concat**, que dado dos lista las concatena. Por ejemplo, dada las listas (A, B, C) y (D, E) la función retorna un apuntador a una nueva lista con los elementos (A, B, C, D, E)

REVISADO:
 NOMBRE: JESUS J. LAREZ M.
 FIRMA:

 Responsable y/o Profesor de la Cátedra

APROBADO:

NOMBRE: MARIA CORA URDANETA
 FIRMA:

 Directora de Escuela de Ing. Informática