PE FINAL 07/2022

EJ T1

```c
int foo(int n, int m)
{
    int miFoo = 1;

    if (m > 0)
    {
        miFoo = foo(n, m - 1) * foo(n, m - 1);
    }

    return miFoo;
}

int main()
{
    printf("%d", (int)foo(0, 2));

    return 0;
}
```

¿Que retorna la funcion?

RESPUESTA CORRECTA: *1*****

La funcion ***fun*** retorna **1** ya que se encuentra la variable local ***miFoo*** y se le declara un valor de 1; cada vez que se llama a la recursiva, el valor de ***miFoo*** se vuelve a establecer como 1.

EJ T2

```c
int main()
{
    char *s[5];
    int i = 0;
    int *x = NULL;
    char c = 2;

    s[i] = &c;

    for(i = 0; i < 5; i++)
    {
        s[i] = &c;
    }

    printf("(%d, %d)", (c == '2'), (*s != &c));
```

```
    return 0;
}
```

¿Que retorna la funcion?

RESPUESTA CORRECTA: ***(0,0)***

EJ T3

```
int main()
{
    char *s = NULL;
    int i;
    char *x = (char*)malloc(sizeof(char));

    for (i = 0; i < 4; i++)
    {
        *(s + i) = 'a' + i;
    }

    s[i] = '\0';

    printf("%s", s);

    return 0;
}
```

¿Que retorna la funcion?

RESPUESTA CORRECTA: ***Error***

Nunca se inicializo la variable ***s***, por lo que al intentar acceder a ella, se produce un error.

EJ T4

```
char foo(unsigned int w)
{
    unsigned int x = 0;
    int i = 0;
    char ar[20] = "domingo";
    x = x | ar[i];
    for(i = 1; ar[i] != '\0'; i++)
    {
        x = x << 8;
        x = x | ar[i];
    }
```

```
    return (x&(255<<w))>>w;
}

int main()
{
    printf("%c", foo(24));
    return 0;
}
```

RESPUESTA CORRECTA: *i*

La funcion ***foo*** retorna el caracter que se encuentra en la posicion ***w*** del arreglo ***ar***. En este caso, ***w*** es 24, por lo que se retorna el caracter ***i***.

EJ T5

```
void sum(int *arr)
{
    *arr = *arr + 3;
}

void resta(int *a)
{
    a = a - 1;
}

int main()
{
    int a[4] = {3, 2, 13, 4};
    int num = 3;
    sum(a);
    resta(&num);
    printf("%d", *(a + num));

    return 0;
}
```

RESPUESTA CORRECTA: *4*

El arreglo ***a*** queda con los valores {6, 5, 16, 7}, y el valor de ***num*** queda en 2. Al imprimir el valor de ***a*** en la posicion ***num***, se obtiene el valor 4. La funcion ***resta*** no modifica el valor de ***num***, ya que no se le pasa la direccion de memoria de ***num***, sino que se le pasa el valor de ***num***. Por lo tanto, la funcion resta no modifica el valor de ***num***.

EJ T6

```c
struct s_x
{
    char c[20];
};
typedef struct s_x *t_x;

struct s_dato
{
    int a[4];
    char c[20];
    int b;
    t_x x;
};
typedef struct s_dato t_dato;

int main()
{
    t_dato d;
    printf("%d\n", (int)sizeof(d));

    return 0;
}
```

RESPUESTA CORRECTA: *44*

El tamaño de la estructura es de 44 bytes, ya que el tamaño de un puntero es de 4 bytes, y el tamaño de un arreglo de 4 enteros es de 16 bytes.

EJ T7

```c
struct s_dato{
    char c;
    int n;
};
typedef struct s_dato*t_dato;

int main()
{
    struct s_dato *p = malloc(sizeof(t_dato));
    t_dato*p2 = &p;
    (*p).n = 11;
    (*p2)->n = 11;
    printf("%d",((*p).n) + (*p2)->n);
    return 0;
}
```

RESPUESTA CORRECTA: *22*

EJ T8 -> PREORDEN ARBOL

Ejemplo:

```
struct s_nodo
{
    int dato;
    struct s_nodo *izq;
    struct s_nodo *der;
};

typedef struct s_nodo *t_nodo;

void preorden(t_nodo arbol)
{
    if(arbol != NULL)
    {
        printf("%d", arbol->dato);
        preorden(arbol->izq);
        preorden(arbol->der);
    }
}
```

EJ T9 -> INSERTAR EN PILA

Ejemplo:

```
struct s_nodo
{
    int dato;
    struct s_nodo *sig;
};

typedef struct s_nodo *t_nodo;

void insertar(t_nodo *pila, int dato)
{
    t_nodo nuevo = malloc(sizeof(struct s_nodo));
    nuevo->dato = dato;
    nuevo->sig = *pila;
    *pila = nuevo;
}
```

EJ T10 -> CREAR UN ARCHIVO ARCH.TXT

```c
int main()
{
    int c,r;
    FILE *archT = fopen("arch.txt","w");
    fprintf(archT, "%s", "Hola");
    fprintf(archT, "%s"," Mundo");
    fclose(archT);

    FILE *archB = fopen("arch.txt", "rb");
    r = fread(&c, 1, 1, archB);

    while(!feof(archB))
    {
        printf("%c", c);
        r = fread(&c, 1, 1, archB);
    }

    fclose(archB);

    return 0;
}
```

Retorna: ***Hola Mundo***