

TRABAJO PRÁCTICO N° 7

Ejercicios 5

Ejercicio 1

¿Qué se emite?

```
int *h, *k;
h = (int) malloc(sizeof(int));
k = (int) malloc(sizeof(int));
*h = 10
k = 4;
h = k;
*k = 6;
printf("%d\n", *h );
```

Ejercicio 2

¿Qué se emite?

```
int *h, *k;
h = (int) malloc(sizeof(int));
k = (int) malloc(sizeof(int));
*h = 4;
*k = 6;
*h = *k;
*k = 5;
printf("%d\n", *k );
```

Ejercicio 3

¿Cual es el valor de h ?

```
int main( )
{
    int vec[4] = { 1, 2, 3}
    int h, *p;
    p = vec+2;
    p--;
    h = *p + *(p+1) + *(vec+1) + p[2];
    printf("%d\n", h);
    return 0;
}
```

Ejercicio 4

Explicar que sucede en cada línea e este código

```
int *h, i ,l;
h = (int*)malloc( 3* sizeof(int));
h[0] = 1;
h[1] = 2;
h[2] = 3;
for(i =0 ;i < 3 ; i + +)
    printf ("%d\n", h[i]);
free(h); //¿h ahora está nulo?
```

Ejercicio 5

Partiendo del siguiente código:

```
typedef struct s
{
    Double dato;
    Struct s *siguiente;
}Elemento;

Void insertar (Elemento **lista, Elemento *e)
{
    Elemento *p;

    //código pedido

    If (*lista != NULL)
    {
        P = *lista;
        While (p->siguiente != NULL)
            p = p->siguiente;
        p->siguiente = e;
    }
    Else
        *lista = e;
}
```

Qué código tenemos que escribir en el lugar del comentario para insertar el elemento apuntado por e, suponiendo que esta perfectamente iniciado?

- a) e->siguiente = lista;
- b) e->siguiente = *lista;
- c) e->siguiente = **lista;
- d) e->siguiente = NULL;

Ejercicio 6

Partiendo del siguiente código:

```
typedef struct s
{
    Double dato;
    Struct s *siguiente;
}Elemento;

Void mostrar (Elemento *p)
{
    If (p != NULL)
    {
        Mostrar (p->siguiente);
        Printf("%d ", p->dato);
    }
}
```

Que visualizará la función mostrar cuando reciba como argumento una lista con los valores 1, 3, 5, 7, 9 (el primer elemento es el 1)?

- a) 1 5 3 9 7
- b) 1 3 5 7 9
- c) 9 7 5 3 1
- d) 7 9 5 1 3

Ejercicio 7

Ingresar palabras y almacenarlas en arrays dinámicos para que sea eficiente el uso del espacio de memoria. Las direcciones de las mismas se almacenarán en un vector. Luego, ordenar alfabéticamente las mismas realizando el intercambio correspondiente de punteros, y emitir la secuencia de palabras ordenada. Resolver utilizando una estructura.

Ejercicio 8

Definir un tipo de datos que represente a una persona, siendo sus campos: el nombre (char *), el apellido (char *), el DNI (char [8]) y la edad (int). Llamar PERSONA a este tipo de datos. Escribir un programa que pida por pantalla esos datos y rellene una variable de tipo PERSONA. Prestar atención al hecho de que los miembros nombre y apellido deben alocarse dinámicamente, y liberarse posteriormente. Escribir una función auxiliar de prototipo void LiberarPersona(PERSONA); que libere dichos punteros..

Ejercicio 9

Modificar el programa del ejercicio 8 para que una vez que ha leído todos los datos, busque mediante un bucle si hay dos personas que tengan la misma edad. Repetir lo mismo para buscar si hay dos personas con el mismo nombre, apellido o DNI. Repetir lo mismo para buscar si hay dos personas que tienen el mismo amigo.

Ejercicio 10

Escribir un programa que imprima una lista de amigos guardados en una agenda la cual esta implementada en una lista ligada doblemente enlazada y los datos de los amigos en una estructura. La impresión debe ser solicitada por un menú, que además permite las opciones de inserción de datos, eliminación y búsqueda en la agenda.

Ejercicio 11

Se tiene una lista ligada cuyos nodos contienen un campo de información con un string y un campo con un puntero al siguiente nodo. Escribir los siguientes módulos:

- Una función que recorra la lista emitiendo el contenido de los campos info.
- Una función que reciba la lista y un dato(string) e inserte en la lista un nuevo nodo al final con dicho string.
- Idem b pero insertando al principio.
- Idem b pero insertando en orden.
- Escribir una función que reciba la lista y retorne un entero indicando cuantos nodos tiene.

Ejercicio 12

Considerar una lista ligada cuyos nodos contienen un campo de información con un char, y un campo con el puntero al siguiente nodo. Escribir los siguientes módulos :

- Función que reciba el puntero de entrada a la lista y un entero y retorne un boolean indicando si el dato esta en la lista
- Función que reciba los punteros de entrada a dos listas y retorne un boolean indicando si ambas listas son exactamente iguales(el mismo contenido, en el mismo orden.
- Función que reciba los punteros de entrada a dos listas y retorne un boolean indicando si la segunda lista contiene a la primera.
- Función que reciba los punteros de entrada a dos listas y retorne el puntero de entrada a una nueva lista generada por el procedimiento, conteniendo la intersección de las listas anteriores.
- Función que reciba una lista y la retorne "al revés"(invirtiendo el orden de los nodos)
- Función que reciba el puntero de entrada a una lista y retorne un boolean indicando si la lista es palindrómica.

Ejercicio 13

Se tiene una lista ligada con nodos formados por dos campos de información un entero y un carácter , y un campo con el puntero al siguiente nodo. Se desean los siguientes módulos:

- función que borre el primer nodo de la lista.
- función que borre el último nodo de la lista.
- función que reciba el puntero de entrada a la lista y un entero y borre el nodo que contenga ese entero.
- Utilizar la estructura dinámica lista para implementar a) una pila; b) una cola.

Ejercicio 14

Escribir una aplicación para que utilizando una pila, simule una calculadora capaz de realizar las operaciones de +, -, * y /. La mayoría de las calculadoras aceptan la notación infija y unas pocas la notación postfija. En estas ultimas, para sumar 10 y 20 introduciríamos primero 10, después 20 y por ultimo el +. Cuando se introducen los operandos, se colocan en una pila y cuando se introduce el operador, se sacan dos operandos de la pila, se calcula el resultado y se introduce en la pila. La ventaja de la notación postfija es que expresiones complejas pueden evaluarse fácilmente sin mucho código. La calculadora del ejemplo propuesto utilizara notación postfija.

De forma resumida, el programa realizará las siguientes operaciones:

- Leerá un dato, operando u operador, y lo almacenara en la variable oper.
- Analizara oper; si se trata de un operando lo meterá en la pila y si se trata de un operador sacara los dos últimos operandos de la pila, realizara la operación indicada por dicho operador y meterá el resultado en la pila para poder utilizarlo como operando en una posible siguiente operación.

Ejercicio 15

Escribir un módulo que reciba los punteros de entrada a dos listas del mismo tipo y retorne un número indicando cuantos nodos de la primera estan repetidos en la segunda. Las listas no estan ordenadas.

Webgrafía y Licencia

- ◆ Textos tomados, corregidos y modificados de diferentes páginas de Internet, tutoriales y documentos.
- ◆ Este documento se encuentra bajo Licencia Creative Commons Attribution – NonCommercial - ShareAlike 4.0 Internacional (CC BY-NC-SA 4.0), por la cual se permite su exhibición, distribución, copia y posibilita hacer obras derivadas a partir de la misma, siempre y cuando se cite la autoría del **Prof. Matías García** y sólo podrá distribuir la obra derivada resultante bajo una licencia idéntica a ésta.
- ◆ Autor:

Matías E. García

Prof. & Tec. en Informática Aplicada
www.profmatiasgarcia.com.ar
info@profmatiasgarcia.com.ar

 **creative commons**

