UNIVERSIDAD DE GRANADA

Metodología de la Programación

Grado de Ingeniería Informática

Práctica 2. Punteros y Memoria Dinámica

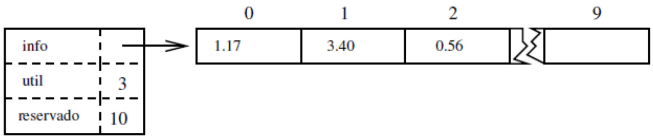
DECSAI

EJERCICIO A

Supongamos la siguiente definición de un tipo de dato:

```
struct vectorSD {
     double *info;
     int util;
     int reservado;
};
```

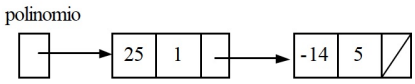
donde `info` es un puntero que mantiene la dirección de una secuencia de reales, `util` indica el número de componentes de la secuencia y `reservado` indica el número de posiciones reservadas de la memoria dinámica para almacenar la secuencia de reales. La siguiente figura muestra un ejemplo de este tipo de representación.



1. Construir un módulo que inicialice una variable de tipo `vectorSD` reservando n casillas de memoria dinámica y ponga el número de componentes usadas a 0.
2. Construir un módulo que añada un elemento en una variable `vectorSD`. Considerar el caso de que la inclusión del nuevo valor sobrepase la reserva de memoria. En este caso, realojar el vector reservando el doble de posiciones.
3. Construir un módulo que realice una copia de una variable `vectorSD` en otra variable del mismo tipo. La copia debe reservar memoria para almacenar sólo las componentes usadas del vector.
4. Construir un módulo que libere la memoria reservada por una variable de tipo `vectorSD`.

EJERCICIO B

Un polinomio en x de un grado arbitrario se puede representar mediante una lista enlazada con punteros, donde cada nodo contiene el coeficiente y el exponente de un término del polinomio, más un puntero al siguiente nodo. Por ejemplo el polinomio: $25x - 14x^5$ se puede representar como:



De esta manera, si consideramos el tipo de dato

```
struct coeficientes{
        int coef;
        int grado;
        coeficientes* sig;
};
```

entonces un dato de tipo `TipoPolinomio` no es más que un puntero al tipo de dato `coeficientes`. Es decir,

```
typedef coeficientes* TipoPolinomio;
```

1. Escribe una función que evalúe un polinomio `p` en un entero `valor`.

   ```
   int evaluar (const TipoPolinomio p, const int valor);
   ```

2. Escribe una función que devuelva el coeficiente del término de grado `i` de un polinomio `p`.

   ```
   int obtener (const TipoPolinomio p, const int i);
   ```

3. Escribe una función que sume 2 polinomios `p1` y `p2`:

   ```
   TipoPolinomio sumar(const TipoPolinomio p1, const TipoPolinomio p2);
   ```

4. Escribe una función que realice la derivada de un polinomio `p`:

   ```
   TipoPolinomio derivada (const TipoPolinomio p);
   ```

Escribir un programa que pida por consola los datos de dos variables de tipo `TipoPolinomio` y un entero para probar la ejecución de las funciones anteriores.