



```

        if (n==*inicio){
            n->sgte=m->sgte;
            m->sgte=n;
            m=*inicio;
        }

    }
    if (m-n>1){
        N->sgte=M->sgte;
        antM->sgte=n;
        m->sgte=postN;
        if (n>1)
            antN->sgte=m;
        else
            *inicio=m;
    }
    return 1;
}
else
    return 0;
else
    return 0;
else
    return 1;
}

```

2. (30 ptos.) Cuando un número es expresado en binario su  $k$ -ésimo dígito representa un múltiplo de  $2^k$ . Por ejemplo:

$$10011_2 = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 19$$

Por otro lado, cuando un número es expresado en base  $sb$  su  $k$ -ésimo dígito representa un múltiplo de  $2^{k+1} - 1$ . Los números se componen únicamente de 0s y 1s excepto el dígito menos significativo distinto de 0 el cual puede ser un 2. Por ejemplo:

$$10120_{sb} = 1 \times (2^5 - 1) + 0 \times (2^4 - 1) + 1 \times (2^3 - 1) + 2 \times (2^2 - 1) + 0 \times (2^1 - 1) = 44$$

En un archivo de texto, llamado "in.txt", se encuentran almacenados un conjunto de números (uno por línea). Escriba un programa en C basado en funciones que para cada número almacenado en el archivo y que se encuentre expresado en base  $sb$ , entregue su equivalente en decimal. La lectura de datos desde el archivo se detiene al leer un 0. Restricción: no se permite el uso de arreglos.

- (a) uso de funciones (5 ptos)
- (b) lectura archivo (5 ptos)
- (b) solución (20 ptos)

```

#include <stdio.h>

main(){
    FILE *fp;
    int numero;

    fp=fopen("in.txt","r");

    while (1){
        fscanf(fp,"%d \n",&numero);
        if (numero!=0){
            if (testBaseSB(numero)){
                decimal=transfSBaDecimal(numero);
                printf("%d(base sb)=%d(decimal)\n",numero,decimal);
            }
        }
    }
}

```

```
        }
    else
        break;
}
fclose(fp);
}
```

```

int testBaseSB(int n){
    int uno=0,dos=0;
    while (n>0){
        digito=n%10;
        switch (digito){
            case 0:
                break;
            case 1:
                uno = 1;
                break;
            case 2:
                dos++;
                if (dos>1) return 0;
                if (uno) return 0;
                break;
            default:
                return 0;
        }
        n/=10;
    }
    return 1;
}

int transfSBaDecimal(int n){
    int decimal,digito;
    int k=0;

    while (n>0){
        digito=n%10;
        decimal += digito*(pow(2,k+1)-1);
        n/=10;
        k++;
    }
    return decimal;
}

```