

```

④ void ImprimeEnBase(valor, base) { entero i=0, resultado=0; caracter a[99];
    // función que sólo utiliza como entradas de base 2, 8, 10, 16
    Si (base != 2 ^ base != 8 ^ base != 10 ^ base != 16) {
        mostrar("base errónea");
        retornar;
    }
    Si (base != 16) {
        Mientras (valor > base) hacer {
            resultado += resto(valor, base) * 10i;
            valor = valor / base;
            i++;
        }
        resultado += valor * 10i;
        mostrar(resultado);
        retornar;
    }
}

```



Mientras (valor > base) hacer {

    resultado = resto(valor, base);

    segun (resultado) {

        caso 0, 1, 2, 3, 4, 5, 6, 7, 8, 9:

            a[i] = 'resultado';

            break;

        caso 10:

            a[i] = 'A';

            break;

        caso 11:

            a[i] = 'B';

            break;

        caso 12:

            a[i] = 'C';

            break;

        caso 13:

            a[i] = 'D';

            break;

        caso 14:

            a[i] = 'E';

            break;

        caso 15:

            a[i] = 'F';

            break;

    }

    valor = valor / base;

    i++;

}

a[i+1] = valor;

```
para (resultado = i+1 hasta 0 con paso -1) hacer {  
    mostrar (a[resultado]);
```

```
}
```

```
    retornar;
```

```
}
```

/\* El valor si se quiere pasar a base 16 está restringido.

Luego de terminar el caso en que la base es 16  
me di cuenta que también sirve para las demás  
bases, pero dejo la primer parte de todas maneras  
para ver si está bien pensado \*/