

Examen 1

Computación 3

Carlos López Roa

19 de septiembre de 2011

1.
 - ++n;
 - n++;
 - n+=1;
 - n=n+1;
2.
 - long int a=7865;
 - long a=7865;
3.
 - F
 - V
 - V
4.
 - Asumiendo \& como &
00000001
(1)
 - 01110111
(119)
 - 01110110
(118)
 - 10101000
(168)

5. else no lleva ;

```
6. #include <stdio.h>\\
#include <math.h>
#define max 126
main(){
    int i;
    for (i=32; i<max; ++i) {
        if ((i%16)==0) {
            putchar(i);
            printf("\n");}
        else
            putchar(i);
    }
    printf("\n");
}
```

7. Este programa ingresa un *unsigned integer*, a dicho integer se le aplica la máscara $1 \ll 15$ que es 1000000000000000 de forma iterativa, en cada iteración al integer se le aplica *integer* $\ll= 1$ que es *integer* = *integer* $\ll 1$ o mejor dicho se recorre a la izquierda en una posición. Se realizan 16 iteraciones. Si sucede que al aplicar la máscara $1 \ll 15$ el resultado es cero entonces se le suma uno a una variable llamada total. Al final la función regresa 1 si el total es divisible entero entre 2 y 0 en otra circunstancia.

En resumen la función aplica una máscara 16 veces, en cada ocasión recorre la entrada a la izquierda

en una posición. Si el resultado de la máscara es igual a cero, incrementa en uno una variable. Si esa variable es divisible entre 2 regresa 1, regresa 0 *else*.

8. ■ Como el argumento de putchar es '1' y '0' la salida es 1 y 0. El uso de putchar es análogo al de printf, sólo sirve para imprimir los valores que se piden.

```
■ /* Printing an unsigned integer in bits */
#include <stdio.h>
#include <math.h>

main() {
    unsigned x;
    void displayBits(unsigned);
    printf("Enter an unsigned integer\n");
    scanf("%u",&x);
    displayBits(x);
    return 0;
}

void displayBits(unsigned value)
{
    unsigned c, displayMask = 1 << 31;
    printf("%10u = ",value);
    for(c=1; c <= 32; c++)
    {
        putchar(value & displayMask ? '1' : '0');
        value <<=1;
        if(c%16 == 0)
            putchar(' ');
    }
    putchar('\n');
}
```

```
9. #include <stdio.h>
#include <math.h>
int power2a(int,int);
void displayBits(unsigned);
main() {
    unsigned x,b;
    printf("Enter an unsigned integer\n");
    scanf("%u",&x);
    printf("Enter a integer power to raise \n");
    scanf("%u",&b);
    printf("Resultado %d\n", power2a(x,b));
    printf("En binario \n");
    displayBits(power2a(x,b));
    return 0;
}

int power2a(int number,int pow){
    int r;
    r=number<<(pow-1);
    return r;
}

void displayBits(unsigned value)
{
    unsigned c, displayMask = 1 << 15;
    printf("%10u = ",value);
    for (c=1; c <= 16; c++)
    {
        putchar(value & displayMask ? '1' : '0');
        value <<=1;
        if (c%8 == 0)
```

```
        putchar(' ');  
    }  
    putchar('\n');  
}
```

Este programa usa $(pow - 1)$ y no pow para que $n^1 = n$ y no a $2n$