

# Programación y Algoritmos I

## Tarea 1

### Problema 1 [ 2 punto ]

**Demostrar** que la suma de dos enteros con signo está en overflow si y solamente si los dos enteros sumados son de mismo signo y si al realizar la operación de adición, el signo de la representación de la suma es diferente de los dos operandos.

### Problema 2 [ 2 punto ]

Convertir los siguientes números expresados en base 10 en binario Y en hexadecimal. En cada caso, expresar cual sería su **opuesto en signed short**.

- 87
- 1101
- 124
- 257

### Problema 3 [ 2 punto ]

Escribir la representación binaria de 0.1 con 6 dígitos después del punto. Redondear al más cercano.

### Problema 4 [ 2 punto ]

Suponiendo que no se manejan números especiales (sólo números flotantes normalizados), escribir todos los números flotantes que se pueden obtener, en base 2, con 6 bits en total, y 2 para el exponente, es decir los números de la forma:

$$f = \pm(1.m)2^e$$

donde  $m$  está encodificado con 3 bits y  $e$  con 2 bits, a la manera de como se vio en la clase.

### Problema 5 [ 2 punto ]

Consideramos el siguiente programa en C. No necesitas todavía entender todos los detalles.

```
#include <stdio.h>
#include <string.h>
```

```
int main(int argc, char *argv[]){
    // Needs one argument
    if(argc < 2){
        return -1;
    }
```

```
    int i = atoi(argv[1]); // This transforms the second argument passed to the progra
    unsigned char s = i;
```

```

    if(s >= 90){
        printf("Thou_shall_not_pass!\n");
        return -1;
    }

    char src_string[]="Les_sanglots_longs_des_violons_de_l'automne_bercent_mon_coeur_d
    char dest_string[90];
    // Copy i bytes from src_string into dest_string
    memcpy(dest_string, src_string, i);
    // This adds an ending character for having a proper string
    dest_string[i] = '\0';
    printf("%s\n", dest_string);
    return 0;
}

```

Compilar el programa y correrlo con los valores siguientes pasados como argumento: 25, 85, 105, 265. Qué explicaciones le puede dar a lo observado? Explicar en particular donde se produce el bug y por qué.