

Laboratório 17

Representação de Ponto Flutuante

1) Escreva uma função “float2int” para converter um número no formato *float* para o formato *int*. Sua função deve fazer a conversão sem usar operações de ponto flutuante. O protótipo da função é:

```
int float2int(float f);
```

Utilize o arquivo abaixo para implementar e a sua função:

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

#define getsig(x)      ((x)>>31 & 1)
#define getexp(x)      ((x)>>23 & 0xff)
#define getfrac(x)      ((x) & 0x7fffff)
#define makefloat(s,e,f) ((s & 1)<<31 | (((e) & 0xff) << 23) | ((f) & 0x7fffff))

typedef union {
    float f;
    unsigned int i;
} U;

int float2int(float f);

int main() {
    float f;
    printf("\n***** float2int *****\n");
    f = 0.0;
    printf(" %+10.4f -> %d\n", f, float2int(f));
    f = 1.0;
    printf(" %+10.4f -> %d\n", f, float2int(f));
    f = -1.0;
    printf(" %+10.4f -> %d\n", f, float2int(f));
    f = 12345.0;
    printf(" %+10.4f -> %d\n", f, float2int(f));
    f = -12345.0;
    printf(" %+10.4f -> %d\n", f, float2int(f));
    f = 1.5;
    printf(" %+10.4f -> %d\n", f, float2int(f));
    f = 2.5;
    printf(" %+10.4f -> %d\n", f, float2int(f));
    f = 2.4;
    printf(" %+10.4f -> %d\n", f, float2int(f));
    f = 0.5;
```

```

printf(" %+10.4f -> %d\n", f, float2int(f));
f = (float) pow(2,31);
printf(" %+10.4f (2^31) = overflow -> %d\n", f, float2int(f));
f = -f;
printf(" %+10.4f (-2^31) -> %d\n", f, float2int(f));
f = (float) pow(2,32);
f = -f;
printf(" %+10.4f (-2^32 = overflow) -> %d\n", f, float2int(f));

return 0;
}

```

Atenção: sua função deve estar preparada para tratar casos de *overflow*, pois nem todo valor *float* pode ser convertido para um inteiro de 32 bits. Em caso de *overflow*, emita uma mensagem de erro, e retorne os valores INT_MAX ou INT_MIN (de acordo com o sinal). Esses valores são definidos no arquivo de cabeçalho limits.h.

2) Escreva uma função “*maxdouble*” que recebe dois valores do tipo *double* e retorna o maior deles. Sua função não deve fazer uma comparação direta entre os valores. Manipule a representação dos valores com operações bit a bit. O protótipo da função é

```
double maxdouble(double a, double b);
```

Utilize o arquivo abaixo como base para implementar e testar sua função:

```

#include <stdio.h>

double maxdouble(double a, double b);

int main() {
    double a,b;

    a = 1.0;
    b = 2.0;
    printf("maior de %f %f = %f\n", a,b,maxdouble(a,b));

    a = -1.0;
    b = -2.0;
    printf("maior de %f %f = %f\n", a,b,maxdouble(a,b));

    a = 1.0;
    b = -2.0;
    printf("maior de %f %f = %f\n", a,b,maxdouble(a,b));

    a = 1.5;
    b = 1.75;
    printf("maior de %f %f = %f\n", a,b,maxdouble(a,b));

    a = -1.5;
    b = -1.75;
    printf("maior de %f %f = %f\n", a,b,maxdouble(a,b));
}

```

```
a = -1.5;
b = -2.5;
printf("maior de %f %f = %f\n", a,b,maxdouble(a,b));

a = 1.5;
b = -1.75;
printf("maior de %f %f = %f\n", a,b,maxdouble(a,b));

return 0;
}
```