

Arquitetura de Computadores 2017/18

Ficha 2

Tópicos: Representação de reais (IEEE 754). Programação em C.

Exercícios sobre bases e representação de dados

Observação: O objectivo desta secção é recordar as bases numéricas e a representação binária.

1. Considere a representação de inteiros e em vírgula flutuante IEEE 754 de precisão simples. Indique os valores representados pelos seguintes padrões de 32 bits:

binário	int. s/sinal dec	int c/sinal dec	FP/IEEE dec
01000001010110000000000000000000			
11000001101010000000000000000000			

2. Admita que numa variável do tipo float coloca o valor 0,1. (considere a norma IEEE 754)
- Qual será a representação binária desse valor em memória? *Se achar que está a obter muitas casas decimais ou se a sua calculadora deixar de ter precisão suficiente, passe à alínea seguinte.*
 - Comprove escrevendo um programa Java que escreve os bits da variável. Use o método `Float.floatToIntBits()` para forçar a interpretação dos bits no float como int e depois o `printBin()` da ficha 1 (em alternativa pode usar o método `toBinaryString` mas lembre-se que os zeros mais significativos não vão aparecer).
 - Complete o programa anterior para afixar no ecrã o valor inteiro representado no expoente.
 - Verifique o número real realmente representado e o erro cometido imprimindo 0.1 com 20 casas decimais: `System.out.printf("%.20f\n", 0.1f);`
3. Crie um programa C que verifica se um dado número inteiro é par usando operadores de lógica binária. Para ler o inteiro do teclado para uma variável, use a função `scanf` do `stdio.h` da seguinte forma:
- ```
scanf("%d", &num);
```
- onde `num` é uma variável inteira declarada da seguinte forma: `int num;`
4. Altere o programa anterior de modo a avaliar agora uma sequência de inteiros introduzida pelo utilizador, guardando num vetor de inteiros todos os que forem pares (admita um limite de 100 inteiros). A sequência é terminada quando o utilizador inserir o valor 0. No fim, escreva no ecrã todos os valores guardados no vetor.
5. Programe em C uma função, que inverte o bit 5 de um byte (`unsigned char`), mantendo os restantes bits inalterados. *Sugestão: implemente a função de acordo com o protótipo seguinte e utilize o operador 'ou exclusivo' (^):*
- ```
unsigned char inverteBit5( unsigned char b )
```

Verifique com um programa, que esta função pode ser usada para passar letras maiúsculas a minúsculas e também para o contrário (veja uma tabela ASCII e compare a representação binária dos códigos das letras maiúsculas e minúsculas).

6. Crie um programa C que recebe uma sequência de valores inteiros e, para cada valor, verifica se é ou não um número primo, afixando o resultado no ecrã. O seu programa tem de implementar e usar a função em baixo, que retorna 1 (*true*) se o número for primo, 0 (*false*) se não for.

```
int ehPrimo(int num);
```