

Lista de exercícios 01 - Matemática e Física para Jogos

Prof. Jose Gilvan Rodrigues Maia

Aluno: Gabriel Silva Ribeiro
Matrícula: 547879

Questão 1 – Realize mudanças de base para os seguintes decimais

(a) 173 para a base 2

Solução:

$$\begin{array}{r}
 173 \mid 2 \\
 172 \mid 86 \mid 2 \\
 \hline 1 \mid 86 \mid 43 \mid 2 \\
 \hline 0 \mid 42 \mid 21 \mid 2 \\
 \hline 1 \mid 20 \mid 10 \mid 2 \\
 \hline 1 \mid 10 \mid 5 \mid 2 \\
 \hline 0 \mid 4 \mid 2 \mid 2 \\
 \hline 1 \mid 2 \mid 2 \\
 \hline 0 \mid 1
 \end{array}$$

←

$$173 = 10101101_2$$

(b) 173 para a base 3

Solução:

$$\begin{array}{r}
 173 \mid 3 \\
 171 \mid 57 \mid 3 \\
 \hline 2 \mid 57 \mid 19 \mid 3 \\
 \hline 0 \mid 18 \mid 6 \mid 3 \\
 \hline 1 \mid 6 \mid 2 \\
 \hline 0
 \end{array}$$

←

$$173 = 20102_3$$

(c) 213 para a base 2

Solução:

$$\begin{array}{r}
 213 \mid 2 \\
 212 \mid 106 \mid 2 \\
 \hline 1 \mid 106 \mid 53 \mid 2 \\
 \hline 0 \mid 52 \mid 26 \mid 2 \\
 \hline 1 \mid 26 \mid 13 \mid 2 \\
 \hline 0 \mid 12 \mid 6 \mid 2 \\
 \hline 1 \mid 6 \mid 3 \mid 2 \\
 \hline 0 \mid 2 \mid 1 \\
 \hline 1
 \end{array}$$

←

$$213 = 11010101_2$$

- (d) 213 para a base 4

Solução:

$$\begin{array}{r}
 213 \quad \begin{array}{|l} 4 \\ \hline \end{array} \\
 \hline
 212 \quad \begin{array}{|l} 53 \quad \begin{array}{|l} 4 \\ \hline \end{array} \\ \hline \end{array} \\
 \hline
 \textcolor{red}{1} \quad \begin{array}{|l} 52 \quad \begin{array}{|l} 13 \quad \begin{array}{|l} 4 \\ \hline \end{array} \\ \hline \end{array} \\ \hline \end{array} \\
 \hline
 \quad \begin{array}{|l} \textcolor{red}{1} \quad \begin{array}{|l} 12 \quad \begin{array}{|l} \textcolor{red}{3} \\ \hline \end{array} \\ \hline \end{array} \\ \hline \end{array} \\
 \hline
 \quad \quad \begin{array}{|l} \textcolor{red}{1} \\ \hline \end{array} \\
 \hline
 \end{array}$$

←

$$213 = 3111_4$$

Também, de forma alternativa, como $4 = 2^2$:

$$213_{10} = 11010101_2 = \boxed{11} \boxed{01} \boxed{01} \boxed{01} = \boxed{3} \boxed{1} \boxed{1} \boxed{1} = \boxed{3111_4}$$

- (e) 213 para a hexadecimal

Solução:

$$213_{10} = 11010101_2 = \boxed{1101} \boxed{0101} = \boxed{D} \boxed{5} = D5_{16}$$

Questão 2 – Resolva as seguintes questões:

- (a) Calcule o resultado de $010101_2 + 001101_2$. Confira o resultado na base decimal.

Solução:

$$\begin{array}{r}
 \textcolor{red}{111} \textcolor{red}{1} \\
 010101_2 \rightarrow 0 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 0 + 16 + 4 + 0 + 4 + 1 = 21_{10} \\
 001101_2 \rightarrow 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 0 + 0 + 8 + 4 + 0 + 1 = 13_{10} \\
 \hline
 \boxed{100010_2} \rightarrow 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 34_{10}
 \end{array}$$

$$21 + 13 = 34 \quad \text{OK!}$$

- (b) Tome o resultado dos itens (a) e (c) da questão anterior. Subtraia o primeiro do segundo.

Solução:

$$(c) - (a) \rightarrow 2000_3 - 100010_2 = 54_{10} - 34_{10} = 20_{10}$$

$$\boxed{20_{10} = 10100_2 = 202_3}$$

- (c) Quanto é $1102_3 + 0121_3$?

Solução:

$$\begin{array}{r}
 \textcolor{red}{111} \\
 1102_3 \\
 0121_3 \\
 \hline
 \boxed{2000_3}
 \end{array}$$

- (d) Qual é o resultado do item anterior na base decimal?

Solução:

$$2000_3 \rightarrow 2 \cdot 3^3 + 0 \cdot 3^2 + 0 \cdot 3^1 + 0 \cdot 3^0 = 2 \cdot 27 + 0 + 0 + 0 = \boxed{54_{10}}$$

Questão 3 – Responda às seguintes questões, usando suas próprias palavras:

1. O que é a representação binária por complemento de 2? Para que serve?

Solução: Representação binária por complemento de 2 nada mais é do que uma maneira de se trabalhar com números binários negativos. Com o complemento, é possível realizar operações de adições e subtrações com números positivos e negativos. O algoritmo consiste em inverter cada bit do número ao seu oposto, e então, ao final soma-se 1.

2. O que é *overflow*? O que é *underflow*? Como esses fenômenos podem interferir na vida de um programador de jogos?

Solução: Overflow é um fenômeno de bug que ocorre quando tentamos representar um número cujo valor ultrapassa a quantidade de memória disponível (muito grande para ser representado). Com o underflow, o caso é o contrário, ocorre uma falta de memória para poder representar certo valor (muito pequeno para ser representado). Esses fenômenos impactam bastante na dinâmica de jogos, por exemplo, quando queremos modelar um número de fases de um jogo arcade, pode acontecer do jogador jogar todas as fases armazenadas, e assim o console irá acessar endereços de fases inexistente (ocorre um overflow).

Questão 4 – Crie um programa que receba (1) uma base numérica $b > 1$ e um inteiro não-negativo i_b na base **b** e imprima o valor de **i** na base decimal.

Solução: Escrito em C++

```
#include <iostream>
#include <cmath>

using namespace std;

int main(){
    int b, digit, e, dec = 0;
    string i;
    cout << "Digite a base numérica(b): ";
    cin >> b;
    cout << "Digite o inteiro não-negativo a ser convertido: ";
    cin >> i;
    e = i.size() - 1;
    for(int l = 0; l < i.size() ; l++){
        if(i[l] >= 'A'){
            digit = (int) i[l] - 'A' + 10;
        }else{
            digit = (int) i[l] - '0';
        }
        dec += digit * pow(b, e);
        e--;
    }
    cout << "O número " << i << " na base " << b << " é igual a ";
    cout << dec << " na base 10.\n";
    return 0;
}
```

```
}
```

Link para o GitHub com o código devidamente comentado → [aqui](#)

Questão 5 – Escreva um programa que converta um número decimal d para uma base numérica $b > 0$.

Solução: Escrito em C++

```
#include <iostream>
#include <cmath>

using namespace std;

int main(){
    int d, b, q, n[500], i = 0;
    cout << "Digite o número decimal a ser convertido (d): ";
    cin >> d;
    cout << "Digite a base para qual deseja-se converter(b): ";
    cin >> b;
    q = d;
    while(q >= b){
        n[i] = q % b;
        q /= b;
        i++;
    }
    n[i] = q;
    cout << "O equivalente de " << d << " na base " << b << ", é ";
    for(int j = i; j >= 0; j--){
        if(n[j] >= 10){
            cout << (char) (n[j] + 55);
        }else{
            cout << n[j];
        }
    }
    cout << ".\n";
    return 0;
}
```

Link para o GitHub com o código devidamente comentado → [aqui₂](#)

Questão 6 – Com base nos programas anteriores, escreva um programa que converta números entre duas bases numéricas.

Solução: Escrito em C++ (claro), uma leve modificação foi feita sob o código da questão 4 com algoritmos provenientes da questão 5.

```
#include <iostream>
#include <cmath>

using namespace std;
```

```

int main(){
    int b1, b2, q, digit, e, dec = 0, j = 0, n[500];
    string i;
    cout << "Digite a base n mica do n mero a ser inserido (b1): ";
    cin >> b1;
    cout << "Digite o inteiro n o-negativo a ser convertido: ";
    cin >> i;
    cout << "Digite a base para qual deseja-se converter(b2): ";
    cin >> b2;
    e = i.size() - 1;
    for(int l = 0; l < i.size() ; l++){
        if(i[l] >= 'A'){
            digit = (int) i[l] - 'A' + 10;
        }else{
            digit = (int) i[l] - '0';
        }
        dec += digit * pow(b1, e);
        e--;
    }
    q = dec;
    while(q >= b2){
        n[j] = q % b2;
        q /= b2;
        j++;
    }
    n[j] = q;
    cout << "O equivalente de " << i << "(base " << b1 << ") na base " << b2;
    cout << ",   ";
    for(int k = j; k >= 0; k--){
        if(n[k] >= 10){
            cout << (char) (n[k] + 55);
        }else{
            cout << n[k];
        }
    }
    cout << ".\n";
    return 0;
}

```

Link para o GitHub com o c digo → [aqui](#)₃