

# Conversão de Bases

Introdução a Ciências da Computação

UFSJ



Universidade Federal  
de São João del-Rei

## 1 Grupo

João Antônio Santos Carvalho  
José Antônio Lopes Paiva Júnior  
Rodrigo Rodrigues Silva Camargos  
Paulo Victor Fernandes Sousa

## 2 Objetivo do Trabalho

O objetivo do trabalho é desenvolver um programa que calcule operações básicas de diferentes bases numéricas, ou seja, soma, subtração, multiplicação e divisão de base binária, octal ou hexadecimal.

## 3 Lógica

A lógica pensada para o trabalho foi deixar o usuário selecionar a operação e a base numérica de sua escolha e digitar os números já na base. O programa irá enviar os números para funções auxiliares, as quais irão convertê-los para a base decimal, efetuar o cálculo e convertê-los novamente para a base escolhida, mostrando assim o resultado.

## 4 Execução do Programa

Ao iniciar o programa irá aparecer um menu e o usuário deve selecionar qual a operação que deseja fazer.

```
Qual operação deseja fazer?  
1 - Soma  
2 - Subtração  
3 - Divisão  
4 - Multiplicação
```

Feito isso, aparecerá outro menu, o qual o usuário irá escolher qual base numérica ele fará as operações.

```
Em qual base voce quer calcular?  
1 - Numeros Binarios  
2 - Numeros Octais  
3 - Numeros Hexadecimais
```

Depois de selecionar a operação desejada e a base que irá calcular, deve-se digitar dois números, logo irá aparecer a resposta.

```
Digite um numero:  
Digite outro numero:  
Resposta:
```

## 5 Programação do Código

### 5.1 Bibliotecas

No começo do código declaramos as bibliotecas, apenas as necessárias. As bibliotecas `stdlib.h` e `stdio.h` são as mais comuns, porém inserimos `locale.h` e `string.h`, para permitir respectivamente acentuação normal do português na saída de caracteres, e utilização de funções de string.

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

### 5.2 Menu

Logo depois, há duas funções para os menus, a primeira para escolher qual operação e a segunda para escolher a base numérica. As duas irão retornar para a função principal as opções escolhidas pelo usuário.

```
int menu1(){
    int op;
    printf("Qual operação deseja fazer?\n");
    printf("1 - Soma\n2 - Subtração\n3 - Divisão\n4 - Multiplicação\n");
    scanf("%d",&op);
    return(op);
}

int menu2(){
    int op;
    printf("Em qual base voce quer calcular?\n");
    printf("1 - Numeros Binarios\n2 - Numeros Octais\n3 - Numeros Hexadecimais\n");
    scanf("%d",&op);
    return(op);
}
```

## 5.3 Conversões

### 5.3.1 Para decimal

Segundo a lógica do programa, a função principal irá mandar para as funções auxiliares os números digitados pelo usuário. O programa irá converter tal número da base escolhida para decimal, para efetuar a operação selecionada.

```
int octal(int oct){
    int total = 0, potenc = 1;
    while(oct > 0){
        total += oct % 10 * potenc;
        oct = oct / 10;
        potenc = potenc * 8;
    }
    return total;
}

int binario(int bin){
    int total = 0, potenc = 1;
    while(bin > 0) {
        total += bin % 10 * potenc;
        bin = bin / 10;
        potenc = potenc * 2;
    }
    return total;
}

int hexadecimal(int hexa){
    int total = 0, potenc = 1;
    while(hexa > 0){
        total += hexa % 10 * potenc;
        hexa = hexa / 10;
        potenc = potenc * 16;
    }
}
```

### 5.3.2 Para a base selecionada

Depois de efetuada a operação em decimal, o resultado irá ser convertido para a base escolhida previamente.

```
void hexdec(long long decimal)
{
    char VALORHEX[] = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F'};
    long long tdec;
    char hex[65];
    int i, y;
    tdec = decimal;
    i = 0;
    while(tdec != 0){
        y = tdec % 16;
        hex[i] = VALORHEX[y];
        tdec /= 16;
        i++;
    }
    hex[i] = '\0';
    printf("Resposta:\n%s", hex);
}

void bindec(int num){
    int i, bin[7];
    for (i = 7; i >= 0; i--) {
        if (num % 2 == 0) {
            bin[i] = 0;
            num = num / 2;
        }
        else {
            bin[i] = 1;
            num = num / 2;
        }
    }
    printf("Resposta:\n");
    for (i = 0; i <= 7; i++) {
        printf("%d", bin[i]);
    }
}

void octdec(int n) {
    int resultado[15] = {};
    int i = 0, y = 0;
    if(n >= 8) {
        printf("Resposta:\n");
        while(n >= 8) {
            resultado[i++] = n % 8;
            n = n / 8;
        }
        resultado[i] = n;
        for(y = i; y >= 0; y--) {
            printf("%d", resultado[y]);
        }
        printf("\n");
    }
    else {
        printf("Resposta:\n%d\n", n);
    }
}
```

## 5.4 Função Principal

Iniciamos a função principal selecionando a língua portuguesa (usando a função `locale.h`), declaramos as variáveis, sendo que `op1` e `op2` receberão seus valores das funções dos menus.

```
int main(){
    setlocale(LC_ALL, "Portuguese");
    int op1, op2,num1,num2;
    op1=menu1();
    op2=menu2();
```

De acordo com esses valores, iniciará os cálculos do programa.

```
//Divisão
if(op1==3){
    printf("Digite um numero: ");
    scanf("%d",&num1);
    printf("Digite outro numero: ");
    scanf("%d",&num2);
    if(op2==1){
        bindec(binario(num1)/binario(num2));
    }
    if(op2==2){
        octdec(octal(num1)/octal(num2));
    }
    if(op2==3){
        hexdec(hexadecimal(num1)/hexadecimal(num2));
    }
}

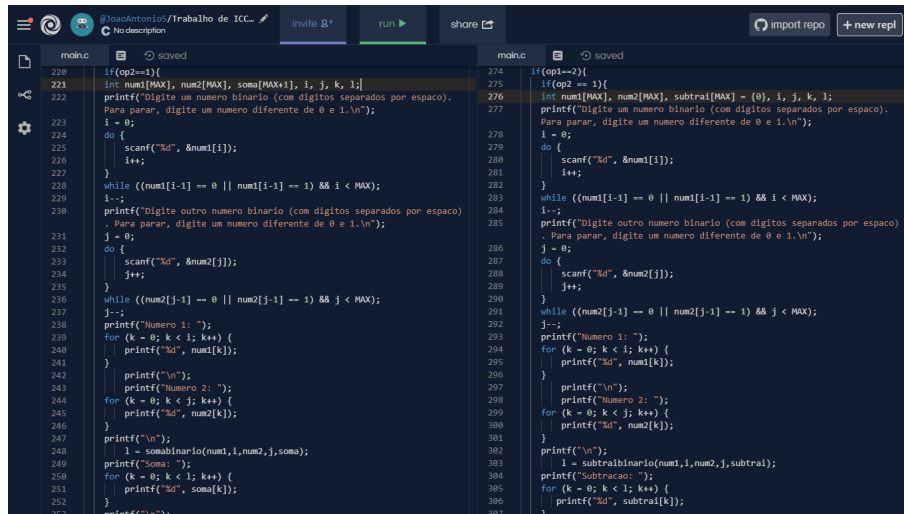
//Multiplicação
if(op1==4){
    printf("Digite um numero: ");
    scanf("%d",&num1);
    printf("Digite outro numero: ");
    scanf("%d",&num2);
    if(op2==1){
        bindec(binario(num1)*binario(num2));
    }
    if(op2==2){
        octdec(octal(num1)*octal(num2));
    }
    if(op2==3){
        hexdec(hexadecimal(num1)*hexadecimal(num2));
    }
}
```

```
//SOMA
if(op1==1){
    printf("Digite um numero: ");
    scanf("%d",&num1);
    printf("Digite outro numero: ");
    scanf("%d",&num2);
    if(op2==1){
        bindec(binario(num1)+binario(num2));
    }
    if(op2==2){
        octdec(octal(num1)+octal(num2));
    }
    if(op2==3){
        hexdec(hexadecimal(num1)+hexadecimal(num2));
    }
}

//Subtração
if(op1==2){
    printf("Digite um numero: ");
    scanf("%d",&num1);
    printf("Digite outro numero: ");
    scanf("%d",&num2);
    if(op2==1){
        bindec(binario(num1)-binario(num2));
    }
    if(op2==2){
        octdec(octal(num1)-octal(num2));
    }
    if(op2==3){
        hexdec(hexadecimal(num1)-hexadecimal(num2));
    }
}
```

### 5.4.1 Excessão

A excessão do código foi as operações adição e subtração de números binários, que foi feito sem conversão.



```
220 if(op2==1){
221     int num1[MAX], num2[MAX], soma[MAX+1], i, j, k, l;
222     printf("Digite um numero binario (com digitos separados por espaco).");
223     Para parar, digite um numero diferente de 0 e 1.\n");
224     i = 0;
225     do {
226         scanf("%d", &num1[i]);
227         i++;
228     } while ((num1[i-1] == 0 || num1[i-1] == 1) && i < MAX);
229     i--;
230     printf("Digite outro numero binario (com digitos separados por espaco).");
231     Para parar, digite um numero diferente de 0 e 1.\n");
232     j = 0;
233     do {
234         scanf("%d", &num2[j]);
235         j++;
236     } while ((num2[j-1] == 0 || num2[j-1] == 1) && j < MAX);
237     j--;
238     printf("Numero 1: ");
239     for (k = 0; k < i; k++) {
240         printf("%d", num1[k]);
241     }
242     printf("\n");
243     printf("Numero 2: ");
244     for (k = 0; k < j; k++) {
245         printf("%d", num2[k]);
246     }
247     printf("\n");
248     l = somabinario(num1,i,num2,j,soma);
249     printf("Soma: ");
250     for (k = 0; k < l; k++) {
251         printf("%d", soma[k]);
252     }
253     printf("\n");
274 if(op1==2){
275     int num1[MAX], num2[MAX], subtrai[MAX] = {0}, i, j, k, l;
276     printf("Digite um numero binario (com digitos separados por espaco).");
277     Para parar, digite um numero diferente de 0 e 1.\n");
278     i = 0;
279     do {
280         scanf("%d", &num1[i]);
281         i++;
282     } while ((num1[i-1] == 0 || num1[i-1] == 1) && i < MAX);
283     i--;
284     printf("Digite outro numero binario (com digitos separados por espaco).");
285     Para parar, digite um numero diferente de 0 e 1.\n");
286     j = 0;
287     do {
288         scanf("%d", &num2[j]);
289         j++;
290     } while ((num2[j-1] == 0 || num2[j-1] == 1) && j < MAX);
291     j--;
292     printf("Numero 1: ");
293     for (k = 0; k < i; k++) {
294         printf("%d", num1[k]);
295     }
296     printf("\n");
297     printf("Numero 2: ");
298     for (k = 0; k < j; k++) {
299         printf("%d", num2[k]);
300     }
301     printf("\n");
302     l = subtrabinario(num1,i,num2,j,subtrai);
303     printf("Subtracao: ");
304     for (k = 0; k < l; k++) {
305         printf("%d", subtrai[k]);
306     }
307     printf("\n");
```

## 6 Conclusão

Mesmo reunindo várias vezes o grupo não conseguiu resolver o trabalho conforme foi pedido (sem conversão de números), porém o programa funciona do mesmo jeito. Com excessão da soma e subtração de números binários, que com ajuda do monitor conseguimos fazer.

Se caso as operações não funcionarem no seu compilador instalado tente em um online, recomendamos esse: <https://repl.it/>