Teoria dos Números

1) Números Primos:

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

int main()
{
  int number, raiz_number, isPrime = 1;

  printf("Enter the number\n");
  scanf("%d",&number);

  raiz_number = sqrt(number);

  for(int i = 3; i <= raiz_number; i += 2)
    if (number%i == 0){
      isPrime = 0;
      break;
    }

  if (number == 1 || !isPrime || (number != 2 && number%2 == 0))
    printf("O numero %d, nao eh primo\n", number);
  else
    printf("O numero %d, eh primo\n", number);

  return 0;
}
```

2) MMC e MDC:

```
int mmc(int a, int b) {
	return a * (b / mdc(a, b));
}

int euclidesMDC (int a, int b) {
	if (b == 0) return a;
	else return euclidesMDC(b, a % b);
}
```

3) Sieve of Eratosthenes:

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

int main()
{
  int number, qt_primos, k = 0, raiz_number, isPrime = 1;

  printf("Enter the number\n");
  scanf("%d",&number);

  raiz_number = sqrt(number);
  qt_primos = raiz_number - 1;

  int* primes = (int*) malloc((raiz_number+1)*sizeof(int));
  //Setando todos valores como falso
  for(int j = 0; j <= raiz_number; j++) primes[j] = 0;

  for(int i = 2; i <= raiz_number; i++)
    if(primes[i] == 0)
      for(int j = i*2; j <= raiz_number; j += i)
        if(primes[j] == 0){
          primes[j] = j;
          qt_primos--;
        }

  int* lista_primos = malloc(qt_primos*sizeof(int));
  for(int i = 2; i <= raiz_number; i++){
    if (primes[i] == 0)
      lista_primos[k++] = i;
  }

  for(int i = 0; i < qt_primos; i++)
    if(number % lista_primos[i] == 0){
      isPrime = 0;
      break;
    }

  if (number == 1 || !isPrime)
    printf("O numero %d, nao eh primo\n", number);
  else
    printf("O numero %d, eh primo\n", number);

  return 0;
}
```

4) Fatorial:

```
#include <stdio.h>
int main()
{
    int n, i;
    unsigned long long factorial = 1;

    printf("Enter an integer: ");
    scanf("%d",&n);

    for(i=1; i<=n; ++i)
      factorial *= i;              // factorial =
factorial*i;
    printf("Factorial of %d = %llu", n, factorial);
    return 0;
}
```

5) Fatores Primos:

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

int main(int argc, char const *argv[]) {
  unsigned long long number;
  int potencia = 0;

  printf("Enter an integer: ");
  scanf("%llu",&number);

  printf("Prime Factors\n");
  while(number%2 == 0){
    number = number/2;
    potencia++;
  }
  if (potencia != 0)
    printf("2^%d\n", potencia);

  for (int i = 3; i <= number; i = i+2){
    potencia = 0;
    while (number%i == 0){
      number = number/i;
      potencia++;
    }
    if(potencia != 0)
      printf("%d^%d\n", i,potencia);
  }
  return 0;
}
```

6) Número de Fatores Primos:

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

int main() {
  unsigned long long number;
  int numPF = 0;

  printf("Enter an integer: ");
  scanf("%llu",&number);

  for (int i = 2; i <= number; i = i+2){
    while (number%i == 0){
      number = number/i;
      numPF++;
    }
    if(i == 2)
      i--;
  }

  printf("Numero de Fatores Primos:
%d\n",numPF );
  return 0;
}
```

7) Número de Fatores Primos Distintos:

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

int main() {
  unsigned long long number;
  int potencia = 0, numPF = 0;

  printf("Enter an integer: ");
  scanf("%llu",&number);

  for (int i = 2; i <= number; i = i+2){
    potencia = 0;
    while (number%i == 0){
      number = number/i;
      potencia++;
    }
    if(potencia != 0)
      numPF++;
    if(i == 2)
      i--;
  }

  printf("Numero de Fatores Primos Distintos:
%d\n",numPF );
  return 0;
}
```

8)Soma dos Fatores Primos:

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

```
#include <math.h>

int main() {
  unsigned long long number;
  int sumPF = 0;

  printf("Enter an integer: ");
  scanf("%llu",&number);

  for (int i = 2; i <= number; i = i+2){
    while (number%i == 0){
      number = number/i;
      sumPF += i;
    }
    if(i == 2)
      i--;
  }

  printf("Soma dos Fatores Primos:
%d\n",sumPF );
  return 0;
}
```

9) Número de Divisores:

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

int main() {
  unsigned long long number;
  int potencia = 0, power = 1;

  printf("Enter an integer: ");
  scanf("%llu",&number);

  for (int i = 2; i <= number; i = i+2){
    potencia = 0;
    while (number%i == 0){
      number = number/i;
      potencia++;
    }
    if(potencia != 0)
      power *= (potencia + 1);
    if(i == 2)
      i--;
  }

  printf("Numero de Divisores: %d\n",power);
  return 0;
}
```

10)Soma dos Divisores:

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

int main() {
  unsigned long long number;
  int potencia = 0, power = 1, aux;
  printf("Enter an integer: ");
  scanf("%llu",&number);

  for (int i = 2; i <= number; i = i+2){
    potencia = 0;
    while (number%i == 0){
      number = number/i;
      potencia++;
    }
    if(potencia != 0){
      aux = i;
      for(int p = 0; p < potencia; p++)
        aux *= i;
      power *= ((aux - 1)/(i-1));
    }
    if(i == 2)
      i--;
  }

  printf("Soma de Divisores: %d\n",power);
  return 0;
}
```

11) Exemplo do módulo:

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

#define FOR(i, n) for (__typeof(n)i = 0; i < n;
i++)

const int MOD = 131071;


int main() {
        char a[] = "cadeia_de_100_bits";
        int i = 0;
        int M = 0;
        FOR(i, strlen(a)) {
                M = (M << 1) +  a[i] - '0';
                M %= MOD ;
        }
}
```

12) Equação Diofantina

```
int x,y;
```

```
void ExtendedEuclid(int a, int b) {
        if(b == 0) {
                x = 1;
                y = 0;
                d = a;
                return;
        }
        ExtendedEuclid(b, a%b);
        int x1 = y;
        int y1 = x - (a / b) * y;
        x = x1;
        y = y1;
}

void next(int *x, int *y, int a, int b) {
        // n é qual resultado da “lista” de
possíveis resultados você quer
        *x = x0 + (b/d) * n;
        *y = y - (a/d) * n;
}
```

13) Função Totiente

```
#include <stdio.h>
int phi(int n) {
  int result = n;
  for (int i = 2; i * i <= n; i++) {
    if(n % i == 0) {
      while(n % i == 0)
        n /= i;
      result -= result / i;
    }
  }
  if(n > 1)
    result -= result / n;
  return result;
}
```

14) Mudança de base:

14.1) Qualquer base para decimal:

```
int paraDecimal(char *str, int base)
{
  int tamanhoEntrada = strlen(str);
  int power = 1;
  int num = 0;
  int i;

  for (i = len - 1; i >= 0; i--)
  {
    if (val(str[i]) >= base)
    {
      printf("Invalid Number");
      return -1;
    }

    num += val(str[i]) * power;
    power = power * base;
  }
  return num;
}

int val(char c)
{
  if (c >= '0' && c <= '9')
    return (int)c - '0';
  else
    return (int)c - 'A' + 10;
}
```

14.2) Decimal para qualquer base:

```
char* doDecimal(char res[], int base, int
inputNum)
{
  int index = 0;

  while (inputNum > 0)
  {
    res[index++] = reVal(inputNum % base);
    inputNum /= base;
  }
  res[index] = '\0';

  reverseString(res);

  return res;
}

char reVal(int num)
{
  if (num >= 0 && num <= 9)
    return (char)(num + '0');
  else
    return (char)(num - 10 + 'A');
}

void reverseString(char *str)
{
  int len = strlen(str);
  int i;
  for (i = 0; i < len/2; i++)
  {
    char temp = str[i];
    str[i] = str[len-i-1];
    str[len-i-1] = temp;
  }
}
```