

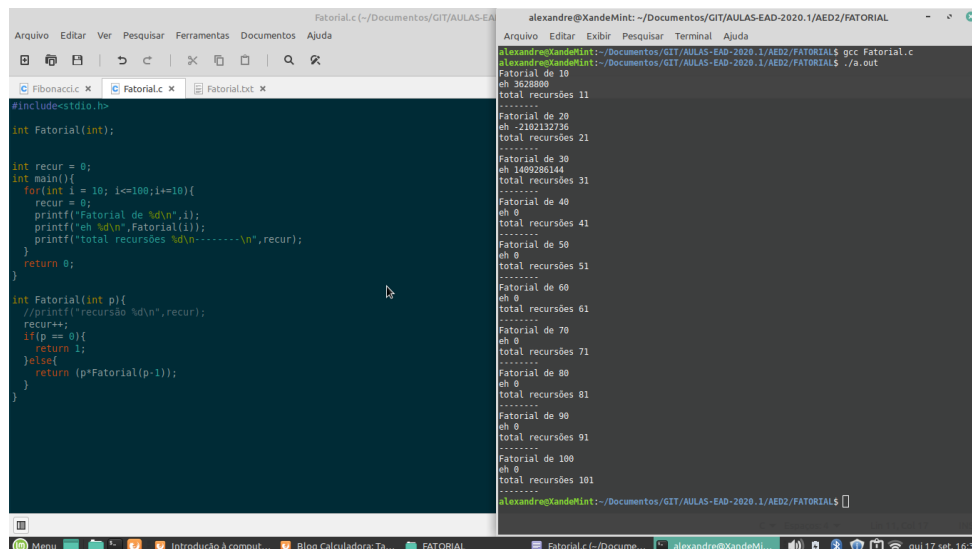
Relatório Fatorial Recursivo

Inicialmente eu criei o código com um for que itera de 10 em 10, indo de 10 a 100. Mas ao executar o código notei algumas coisas.

Primeiro que a quantidade de recursões é igual ao valor inserido mais um ($N+1$).

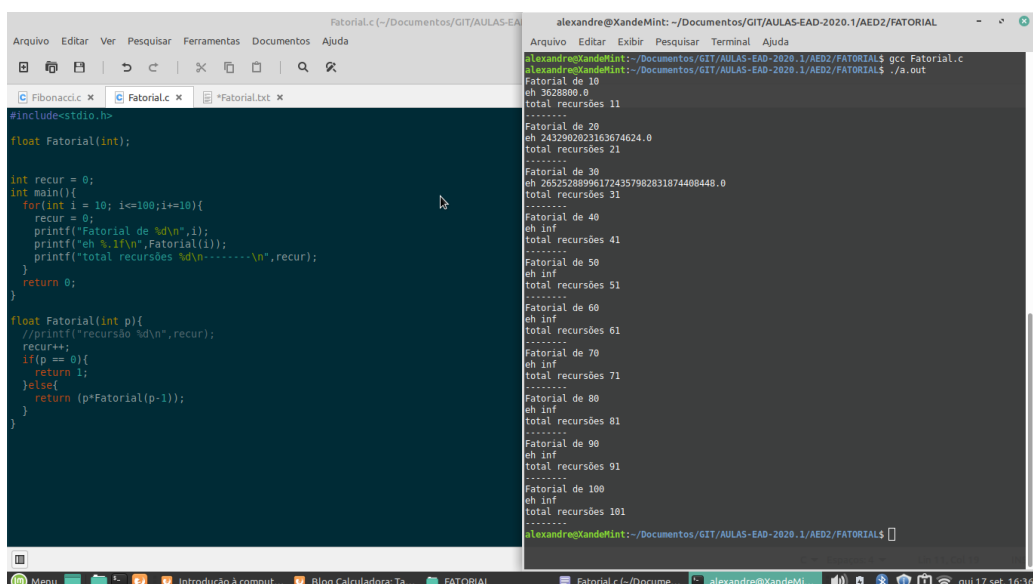
Segundo que o resultado começava a apresentar anormalidade a partir do fatorial de 40.

Checando o código percebi que isso ocorria devido ao fato de ter printado em formato inteiro a resposta.



```
Fatorial.c (~/.Documents/GIT/AULAS-EAD) alexandre@XandeMint: ~/Documents/GIT/AULAS-EAD-2020.1/AED2/FATORIAL
Arquivo Editar Ver Pesquisar Ferramentas Documentos Ajuda
Fibonacci x Fatorial.c x Fatorial.txt x
#include<stdio.h>
int Fatorial(int);
int recur = 0;
int main(){
    for(int i = 10; i<=100;i+=10){
        recur = 0;
        printf("Fatorial de %d\n",i);
        printf("eh %d\n",Fatorial(i));
        printf("total recursões %d\n-----\n",recur);
    }
    return 0;
}
int Fatorial(int p){
    //printf("recursão %d\n",recur);
    recur++;
    if(p == 0){
        return 1;
    }else{
        return (p*Fatorial(p-1));
    }
}
alexandre@XandeMint:~/Documents/GIT/AULAS-EAD-2020.1/AED2/FATORIAL$ gcc Fatorial.c
alexandre@XandeMint:~/Documents/GIT/AULAS-EAD-2020.1/AED2/FATORIAL$ ./a.out
Fatorial de 10
eh 3628800
total recursões 11
-----
Fatorial de 20
eh -2102132736
total recursões 21
-----
Fatorial de 30
eh 1409286144
total recursões 31
-----
Fatorial de 40
eh 0
total recursões 41
-----
Fatorial de 50
eh 0
total recursões 51
-----
Fatorial de 60
eh 0
total recursões 61
-----
Fatorial de 70
eh 0
total recursões 71
-----
Fatorial de 80
eh 0
total recursões 81
-----
Fatorial de 90
eh 0
total recursões 91
-----
Fatorial de 100
eh 0
total recursões 101
-----
alexandre@XandeMint:~/Documents/GIT/AULAS-EAD-2020.1/AED2/FATORIAL$
```

Então para fins de teste, decidi usar o tipo float com 1 casa decimal, mesmo sabendo que ele enfrentaria o mesmo problema.



```
Fatorial.c (~/.Documents/GIT/AULAS-EAD) alexandre@XandeMint: ~/Documents/GIT/AULAS-EAD-2020.1/AED2/FATORIAL
Arquivo Editar Ver Pesquisar Ferramentas Documentos Ajuda
Fibonacci x Fatorial.c x Fatorial.txt x
#include<stdio.h>
float Fatorial(int);
int recur = 0;
int main(){
    for(int i = 10; i<=100;i+=10){
        recur = 0;
        printf("Fatorial de %d\n",i);
        printf("eh %.1f\n",Fatorial(i));
        printf("total recursões %d\n-----\n",recur);
    }
    return 0;
}
float Fatorial(int p){
    //printf("recursão %d\n",recur);
    recur++;
    if(p == 0){
        return 1;
    }else{
        return (p*Fatorial(p-1));
    }
}
alexandre@XandeMint:~/Documents/GIT/AULAS-EAD-2020.1/AED2/FATORIAL$ gcc Fatorial.c
alexandre@XandeMint:~/Documents/GIT/AULAS-EAD-2020.1/AED2/FATORIAL$ ./a.out
Fatorial de 10
eh 3628800.0
total recursões 11
-----
Fatorial de 20
eh 2429202322163674624.0
total recursões 21
-----
Fatorial de 30
eh 265252889961724357982831874408448.0
total recursões 31
-----
Fatorial de 40
eh inf
total recursões 41
-----
Fatorial de 50
eh inf
total recursões 51
-----
Fatorial de 60
eh inf
total recursões 61
-----
Fatorial de 70
eh inf
total recursões 71
-----
Fatorial de 80
eh inf
total recursões 81
-----
Fatorial de 90
eh inf
total recursões 91
-----
Fatorial de 100
eh inf
total recursões 101
-----
alexandre@XandeMint:~/Documents/GIT/AULAS-EAD-2020.1/AED2/FATORIAL$
```

A solução foi usar o tipo double, assim consegui obter o resultado esperado. Pois o valor fatorial aumenta consideravelmente, que os tipos int e float são incapazes de armazenar.

```

#include<stdio.h>

double Fatorial(int);

int recur = 0;
int main(){
    for(int i = 10; i<=100;i+=10){
        recur = 0;
        printf("Fatorial de %d\n",i);
        printf("eh %.1lf\n",Fatorial(i));
        printf("total recursões %d\n-----\n",recur);
    }
    return 0;
}

double Fatorial(int p){
    //printf("recursão %d\n",recur);
    recur++;
    if(p == 0){
        return 1;
    }else{
        return (p*Fatorial(p-1));
    }
}
  
```

```

Fatorial de 10
eh 3628800.0
total recursões 11
-----
Fatorial de 20
eh 2432902008176640000.0
total recursões 21
-----
Fatorial de 30
eh 265252859812191032188804708045312.0
total recursões 31
-----
Fatorial de 40
eh 815915283247897683795548521301193790359984930816.0
total recursões 41
-----
Fatorial de 50
eh 3041409320171337557636966406747986832057064836514787179557289984.0
total recursões 51
-----
Fatorial de 60
eh 8329927112741391580056396102959641077457945541076708813599085350531187384917164032.0
total recursões 61
-----
Fatorial de 70
eh 1197857166996989026992585446055884022526702920952930327894441987121439652486137449869147396
6836492048.0
total recursões 71
-----
Fatorial de 80
eh 715694570462637788320734849864155169245142782150663022833152440197864351902213150585239848
4428816675798776564959674368.0
total recursões 81
-----
Fatorial de 90
eh 148571596440176068859812644465839046848504385338494631194665214503809929244428884407664620
677918947681692538663274334960161140774153486336.0
total recursões 91
-----
Fatorial de 100
eh 9332621544394410218832560610857526724094425485496057150916691040040799506424293714863269403
04505128980429892969447489825873204311236641477561877016501813248.0
total recursões 101
  
```

Usando o site <http://blogcalculadora.blogspot.com/2012/08/tabuada-tabela-de-fatorial-de-1-100.html> e a calculadora eu pude checar se os valores obtidos estavam corretos subtraindo os valores obtidos dos valores fornecidos no site.

The screenshot shows a web browser displaying a table of factorials from 1 to 100. A calculator window is open over the table, showing the calculation of 9332621544394410... minus 0, resulting in 0. The table lists the factorial value and the number of recursive calls for each number from 1 to 100.

n	Fatorial de n	total recursões
10	3628800.0	11
20	2432902008176640000.0	21
30	265252859812191032188804708045312.0	31
40	815915283247897683795548521301193790359984930816.0	41
50	3041409320171337557636966406747986832057064836514787179557289984.0	51
60	8329927112741391580056396102959641077457945541076708813599085350531187384917164032.0	61
70	11978571669969890269925854460558840225267029209529303278944419871214396524861374498691473966836492048.0	71
80	7156945704626377883207348498641551692451427821506630228331524401978643519022131505852398484428816675798776564959674368.0	81
90	148571596440176068859812644465839046848504385338494631194665214503809929244428884407664620677918947681692538663274334960161140774153486336.0	91
100	933262154439441021883256061085752672409442548549605715091669104004079950642429371486326940304505128980429892969447489825873204311236641477561877016501813248.0	101