

# Programming Assignment 1: Finding Perfect Number CO21BTECH11004

Function **IsPerfectNumber(int n)** checks whether a number is perfect or not. If it is a perfect number then IsPerfectNumber return **1** else return **0**.

In the main function, n and k are read from the input file ("input.txt").

An array is created that would be acting as shared memory between parent and k child processes. Here this array is shared by using POSIX shared memory using mmap.

This array is of length n. Initially all elements of this array are initialized to 0. The  $i^{th}$  index in the array represents  $(i + 1)^{th}$  number and its value represents whether i+1 is perfect or not.

For eg:- arr[0] = 0 , tells that  $0+1 = 1$  :- 1 is not a perfect number.

arr[5] = 1, tells the  $5+1 = 6$  :- 6 is a perfect number

K child processes are created using fork().

The number from 1 to n are divided to k childs by the following distribution:-

Child0 :- k,2k,3k,... (for i from 1 to n :-  $(i\%k == 0)$ )

Child1 :- 1,k+1,2k+1,... (for i from 1 to n :-  $(i\%k == 1)$ )

Hence for  $i^{th}$  child, it has to check on the number from 1 to n which give remainder i when divided by k. (i from 0 to k , excluding k)

Each child process do the following task : -

- Creates a ChildProcess output file.
- Checks for the number in its set whether it is perfect or not.
- If it is perfect then write in its output file "Is a perfect number", otherwise write "Not a perfect number".

- If it is a perfect number then change the value of the corresponding index in the shared array to 1.
- Close the file.
- Exits.

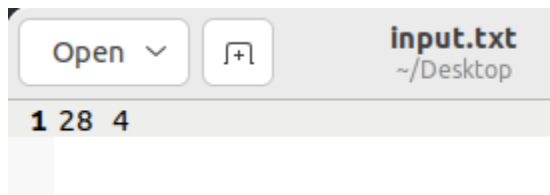
After all the child processes exits, the parent process runs a loop k times executing wait().

Then the parent process creates its output file and write number for which array value is 1 under their corresponding process , it finds corresponding process by checking the mod by k of number.

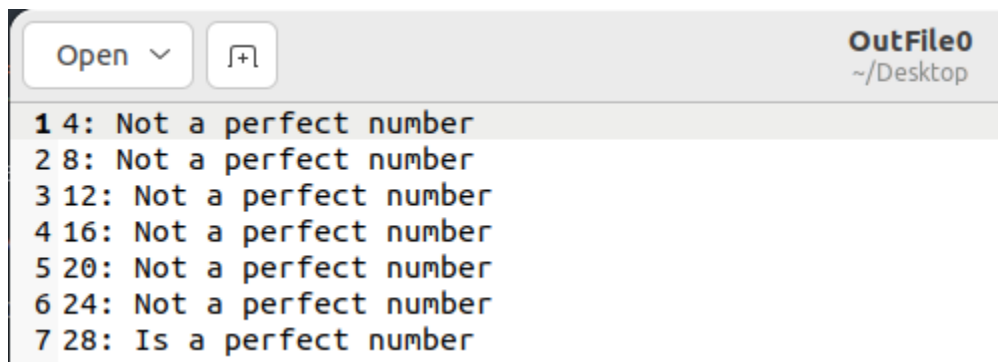
Then unmap the array created by mmap by using munmap.  
Close the input and output file.

For n=28, k=4. Illustration of code is demonstrated.

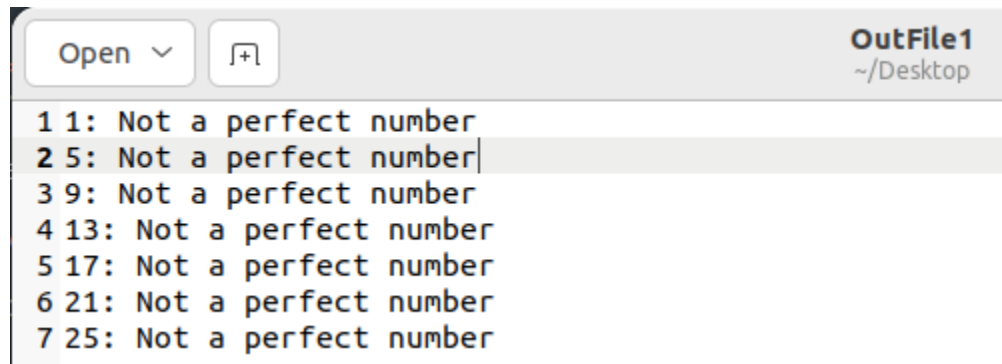
Input file:-



Child Output Files:- 4 Output File (OutFile0, OutFile1, OutFile2, OutFile3)  
OutFile0.txt :-



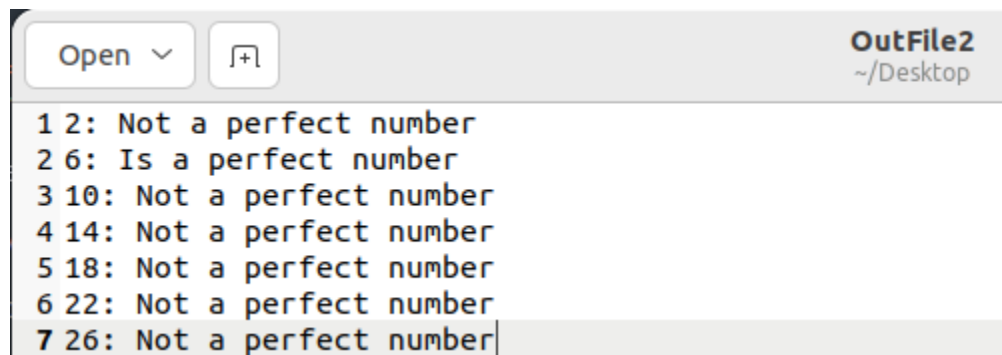
OutFile1.txt :-



The screenshot shows a text editor window with a title bar that includes an 'Open' button with a dropdown arrow, a file icon, and the title 'OutFile1' with the path '~/Desktop'. The text area contains the following lines:

```
1 1: Not a perfect number  
2 5: Not a perfect number  
3 9: Not a perfect number  
4 13: Not a perfect number  
5 17: Not a perfect number  
6 21: Not a perfect number  
7 25: Not a perfect number
```

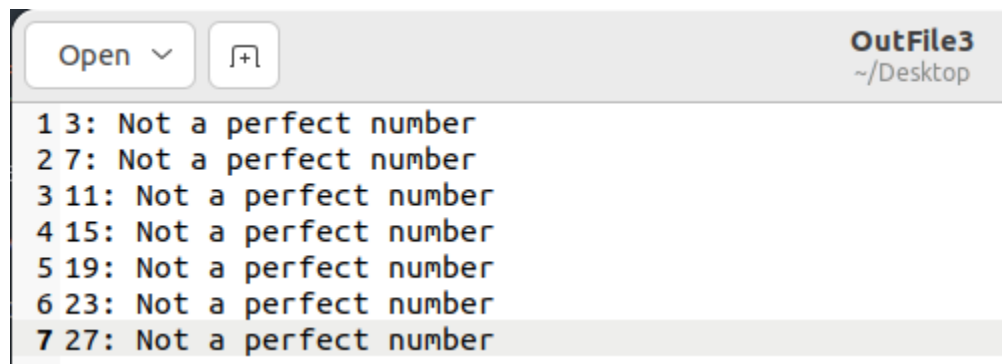
OutFile2.txt :-



The screenshot shows a text editor window with a title bar that includes an 'Open' button with a dropdown arrow, a file icon, and the title 'OutFile2' with the path '~/Desktop'. The text area contains the following lines:

```
1 2: Not a perfect number  
2 6: Is a perfect number  
3 10: Not a perfect number  
4 14: Not a perfect number  
5 18: Not a perfect number  
6 22: Not a perfect number  
7 26: Not a perfect number
```

OutFile3.txt :-



The screenshot shows a text editor window with a title bar that includes an 'Open' button with a dropdown arrow, a file icon, and the title 'OutFile3' with the path '~/Desktop'. The text area contains the following lines:

```
1 3: Not a perfect number  
2 7: Not a perfect number  
3 11: Not a perfect number  
4 15: Not a perfect number  
5 19: Not a perfect number  
6 23: Not a perfect number  
7 27: Not a perfect number
```

Parent Output File:-

OutMain.txt :-

```
Open  ▾  [+]  
OutMain.txt  
~/Desktop  
1 P0 :- 28  
2 P1 :-  
3 P2 :- 6  
4 P3 :-
```

Initially Shared Array :-

```
28 4  
1 :- 0 , 2 :- 0 , 3 :- 0 , 4 :- 0 , 5 :- 0 , 6 :- 0 , 7 :- 0 , 8 :- 0 , 9 :- 0 ,  
10 :- 0 , 11 :- 0 , 12 :- 0 , 13 :- 0 , 14 :- 0 , 15 :- 0 , 16 :- 0 , 17 :- 0 ,  
18 :- 0 , 19 :- 0 , 20 :- 0 , 21 :- 0 , 22 :- 0 , 23 :- 0 , 24 :- 0 , 25 :- 0 ,  
26 :- 0 , 27 :- 0 , 28 :- 0 ,
```

Shared Array after Child process execution is done:-

```
1 :- 0 , 2 :- 0 , 3 :- 0 , 4 :- 0 , 5 :- 0 , 6 :- 1 , 7 :- 0 , 8 :- 0 , 9 :- 0 ,  
10 :- 0 , 11 :- 0 , 12 :- 0 , 13 :- 0 , 14 :- 0 , 15 :- 0 , 16 :- 0 , 17 :- 0 ,  
18 :- 0 , 19 :- 0 , 20 :- 0 , 21 :- 0 , 22 :- 0 , 23 :- 0 , 24 :- 0 , 25 :- 0 ,  
26 :- 0 , 27 :- 0 , 28 :- 1 ,
```