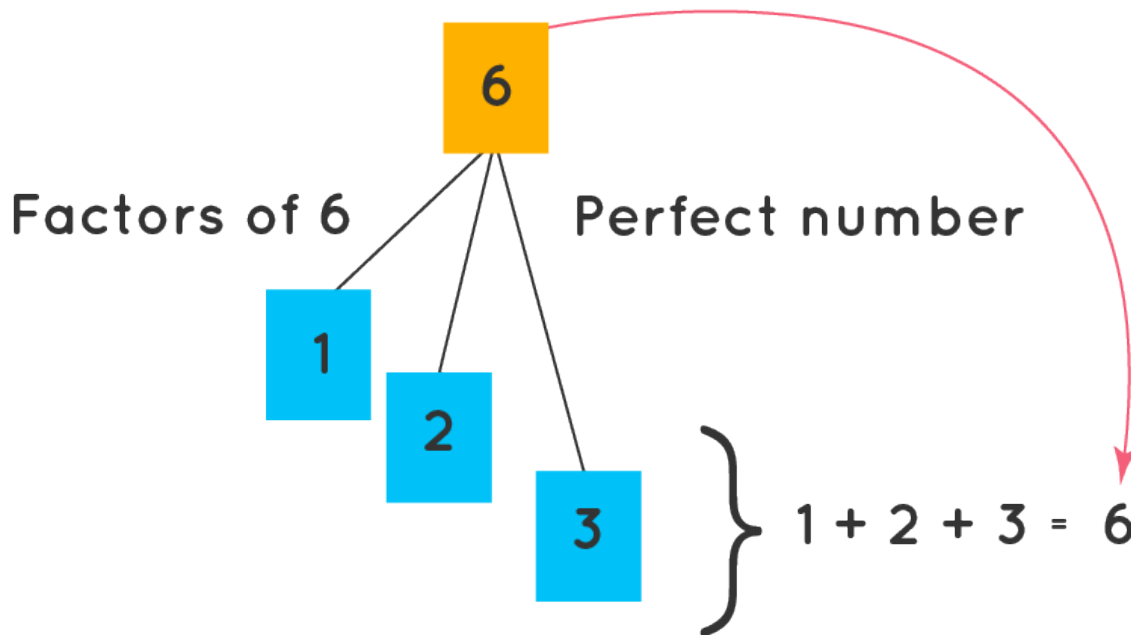


Program 08: perfect numbers

Perfect Number



Write a program that asks the user for an integer and then determines if that integer is perfect. A perfect number is one where its proper divisors sum to the number itself. The proper divisors of an integer are those less than the integer itself. Here is one run of the program:

```
Enter number: 6
6 is perfect!
```

Here is another run:

```
Enter number: 3975
3975 is not perfect
```

And another run:

```
Enter number: 33550336
33550336 is perfect!
```

Calculate using integer arithmetic. Assume that the integer is positive and fits into 32 bits. Your program will ask the user for N. Initialize a sum to zero, then generate trial divisors from one up to a limit in a loop. If a trial divisor divides N with a zero remainder, add that divisor to the sum. Of course, the largest possible proper divisor of N is N/2. If, after looping, the sum is equal to N then N is perfect.

Start your source file with comments that describe it:

```
## CS 254 Program 8
##
## Determine if user-entered number is a perfect number.
##
## Programmer:
## Date:
##
## Register use table:
```

Use instructions discussed in the notes up through chapter 24.

Set SPIM options to the following:

```
OFF Bare Machine
OFF Enable Delayed Branches
OFF Enable Mapped IO
ON  Accept pseudo instructions
ON  Load Exception Handler
OFF Enable Delayed Loads
```

Turn in: Your source code file, nicely commented and formatted to professional standards. Be sure that the columns for mnemonics, operands and comments are straight. Remove all tabs from the source file.

Nearly every line of code will have a comment that explains what it does in terms of the problem. Use the Blackboard assignments tool to turn in your file.

As always, play awith the problem with paper and pencil. Then program a stub that does input and output. Then write and debug the main loop. Then insert the body of the loop.