COP4342 - 2007 Fall

Assignment 6

Fermat's method using GMP

Objectives: Learn how to use the GMP math package.

Instructions: Your assignment is modify a C program called `fermat.c` so that it computes a non-trivial factorization (if it exists) of an arbitrarily large composite odd number using Fermat's method. You are given a program which only computes factorizations for small odd numbers where the components stay under MAXINT. You are to modify this program so that it uses the GMP library to compute the same factorization for arbitrarily large inputs.

To review, Fermat's method is simply based on the observation that an odd number N can be factored simply by observing that

$$N = (x - y)(x + y)$$

$$N = x^2 - y^2$$

$$x^2 - N = y^2$$

This suggests that you all might have to do in an algorithm is set some variable x to the first square above N, and then iterate over increasing squares until you find an $x^2 - N$ that is a perfect square:

```
subroutine factor
{
  while(!perfect_square(x^2 - N))
  {
    // try next
    x = x + 1
  }
}
return(x + sqrt(x^2 - N), x - sqrt(x^2 - N))
```

Here's a small C program that does this:

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <values.h>
int main(int argc, char **argv)
{
  int N = 0;
  if(argc > 1)
    {
      N = atoi(argv[1]);
    }
  else
    {
      printf("Please specify a number to factor...\n");
      exit(1);
    }
  if(N <= 0)
    {
      printf("'%s' doesn't appear to be a number greater than zero...\n",argv[1]);
      exit(1);
    }
  int base,x,xsqr;
  int y,ysqr;
  base = sqrt(N);
  // check for exact square root
  if(base * base == N)
    {
      printf("Factorization: %d = %d * %d\n",N,base,base);
      exit(0);
    }
  // ... else do Fermat's method
  int count = 0;
  for(x = base+1; (float)x * (float)x < (float)MAXINT && (count < (2 * N)); x++)
    {
      xsqr = x * x;
      ysqr = xsqr - N;
      y = (int)sqrt((double)ysqr);
      if(y * y == ysqr)
        {
          // both x+y and x-y should be a factor of N
          int factor1, factor2;

          // check x+y
          factor1 = x+y;
          if((factor1 != N) && (factor1 != 1) && (N % factor1 == 0))
            {
              factor2 = N / factor1;
              printf("Factorization: %d = %d * %d\n",N,factor1,factor2);
              exit(0);
            }
        }
      count++;
    }
  printf("No non-trivial factorization found for %d\n",N);
}
```

Your task is to convert this to a program that uses GMP for all of its calculations, and should be able to factor such numbers as

```
1 69036 16637 97588 93752 56792 74360 99199 38989 63263 20951
```

(which looks formidable, but is actually quite amenable to factorization methods based on Fermat's algorithm.)

Submission: Submit the `fermat.c` C program as an attachment in an e-mail message to `cop4342@cs.fsu.edu` by 11:59pm on Monday, November 26th.