2 Prime Numbers and Factorisation

Question 1: The source and header files are in Appendix B (files _____). A simple modification to speed up this test is to divide n by every natural number less than $\sqrt{n}$ (rather than n). (PROOF!)(ALSO ONLY TEST EVEN NUMBERS...) Thus number of trial divisions is reduced. The output of my program when the prime 17 is entered is:

```
Enter n:17
m=8, r=8.500000
m=5, r=5.666667
m=4, r=4.250000
Is Prime.
```

The output when the composite number 25 is entered is:

```
Enter n:25
m=12, r=12.500000
m=8, r=8.333333
m=6, r=6.250000
m=5, r=5.000000
Not prime.
```

Question 2: The source and header files are in Appendix B (files _____). The text file that is created by P2_1question2.c is in Appendix A (printout ___). My results are an example of the Prime Number Theorem since the quantity $\frac{\pi(n)}{\left(\frac{n}{log(n)}\right)}$ approaches 1 as n increases.

Question 3: The source file is in Appendix B (file ___). The output when the number 378 is entered is:

```
Enter n:378
Prime factors: 2 3 3 3 7
```

Let $n = p_1 p_2 ... p_k$, where $p_i \leq p_{i+1}$. My algorithm makes at most $\sqrt{n} - 1 + k$ trial divisions since it divides by every natural number less than $\sqrt{n}$ (except for 1) and does another division for every factor that it finds (it only moves onto the next factor when all smaller factors have been removed). Each trial division has a constant (independent of n) number of arithmetic operations. In addition, it is possible to create an upper bound for k in terms of n:

$n = p_1 p_2 ... p_k \Rightarrow 2^k \leq n \Rightarrow k log(2) \leq log(n) \Rightarrow k \leq \frac{log(n)}{log(2)}$.

This means that the complexity of my algorithm is $\mathcal{O}(\sqrt{n} + log(n)) = \mathcal{O}(\sqrt{n})$.

3 Linear Congruences

Question 4: The source file is in Appendix B (file ___). The full printouts of P2_1question4.c are in Appendix A (printouts ___). Table 1 shows the pairs of numbers and the corresponding highest common factors and linear combinations.

a	b	HCF = Linear Combination of a and b
2 028 532 369	1 926 605 803	5503 = 2028532369*(104433) + 1926605803*(-109958)
1 853 573 797	1 816 883 927	7823 = 1853573797*(92652) + 1816883927*(-94523)
1 992 502 829	1 750 170 271	7621 = 1992502829*(62638) + 1750170271*(-71311)
1 672 295 039	821 350 697	3821 = 1672295039*(21010) + 821350697*(-42777)

Table 1

Question 5:

Lemma. $ax \equiv ay\,(mod\,(m)) \Rightarrow x \equiv y\left(mod\left(\frac{m}{(a,m)}\right)\right)$.

Proof. $ax \equiv ay\,(mod\,(m)) \Rightarrow ax - ay = km \Rightarrow \frac{a}{(a,m)}(x-y) = \frac{m}{(a,m)}k$.

Now, $\left(\frac{a}{(a.m)}, \frac{m}{(a,m)}\right) = 1$, so $\frac{m}{(a,m)} \mid (x-y)$. Thus $x \equiv y\left(mod\left(\frac{m}{(a,m)}\right)\right)$. □

Method: First, use the Euclidean algorithm to find $(a,m) := HCF\,(a,m)$. Now, $(a,m) \nmid b \Rightarrow (a,m) \mid ax + mz = b\perp$. So there is no solution unless $(a,m) \mid b$.

By the above Lemma,

$ax \equiv b\,(mod\,(m)) \Rightarrow \frac{a}{(a,m)}x \equiv \frac{b}{(a,m)}\left(mod\left(\frac{m}{(a,m)}\right)\right)$.

However $\left(\frac{a}{(a.m)}, \frac{m}{(a,m)}\right) = 1$, so by the Euclidean algorithm,

$(\exists\alpha,\beta)\,\alpha\frac{a}{(a,m)} + \beta\frac{m}{(a,m)} = 1 \Longrightarrow (\exists\alpha)\,\alpha\frac{a}{(a,m)} \equiv 1\left(mod\left(\frac{m}{(a,m)}\right)\right)$.

That is, $x \equiv \alpha\frac{b}{(a,m)}\left(mod\left(\frac{m}{(a,m)}\right)\right) \Rightarrow x \equiv \alpha\frac{b}{(a,m)} + c\frac{m}{(a,m)}\,(mod\,(m))$, where $c = 0, 1...\,((a,m)-1)$. That is there are (a,m) solutions.

Question 6: The source file and header file are in Appendix B (files ___). The full printouts of P2_1question6.c are in Appendix A (printouts ___). Table 2 shows the congruences and their solutions:

Congruence	Solution
$38449x \equiv 10910\,(mod\,(32063))$	$x = 72$
$23151x \equiv 7617(mod\,(7717))$	No solutions.
$34387x \equiv 6777(mod\,(19829))$	$x = 10 + 79c\,(c = 0, 1...250)$

Table 2

There are no solutions in the second example since $23151 \equiv 0\,(mod\,(7717))$ but 7617 is not.

4 Finding Inverses and Breaking RSA

Question 7: Calculating d involves using the Euclidean algorithm on $\phi(n)$ and e (which is w.l.o.g.$< \phi(n)$). For all numbers involved to be $< 2^{13}$, it is sufficient that $\phi(n) < 2^{13} \Rightarrow (p-1)(q-1) < 2^{13}$. That is, $max\{p,q\} < 2^{\frac{13}{2}} + 1$ is a condition that would guarantee that $\phi(n) < 2^{13}$. (C.F. question 10)

Question 8: To find d from n, my algorithm first factorizes $n = pq$. Then it uses the Euclidean algorithm on $\phi(n)$ and e, to find d.

Lemma. *Let the i^{th} equation in the Eulcidean algorithm be $r_i = q_i r_{i+1} + r_{i+2} \Rightarrow r_{i+2} \equiv r_i \,(mod\,(r_{i+1}))$. Then $r_{i+2} < \frac{1}{2} r_i$.*

Proof. If $r_{i+1} < \frac{1}{2} r_i$, then the result follows from the fact that the r_i are strictly decreasing. If $\frac{1}{2} r_i \leq r_{i+1}$, then $r_{i+2} \equiv r_i \,(mod\,(r_{i+1})) \Rightarrow r_{i+2} = r_i - r_{i+1} < \frac{1}{2} r_i$. □

Thus the number of steps has an upper bound that is $\frac{4*log(\phi(n))}{log(2)}$, since $\phi(n) = r_0$ in the above Lemma. So the contribution of the Euclidean algorithm when calculating the number of arithmetic operations needed to find d is negligable. The number of arithmetic operations in the factorization stage is less than $2\sqrt{n}$. For example, given that none of the numbers can exeed 2^{31} in my implementation, a rough order of magnitude would be $\left(2\sqrt{2^{31}} \approx 92681 \approx\right) 10^5$.

Question 9: The source and header files are in Appendix B (files ____). The full printouts are in Appendix A (printouts ____). Table 3 shows the encryption key and the value of d in the computed decryption key (n,d):

Encryption Key: (n,e)	d
(1 805 760 301, 50 512 913)	1175911973
(1 659 317 047, 23 816 383)	361727455
(1 764 053 131, 99 833)	518520281
(145 153 111, 14 655 761)	76350225
(1 955 520 821, 5 537 969)	1782936689
(1 723 466 867, 166 907 407)	1047592543

Table 3

Question 10: The source and header files are in Appendix B (files ____). A test output P2_1question10.c is in Appendix A (printouts ____). Working modulo n, n-1 is the largest number involved. However, my algorithm raises c to the d^{th} power by repeated multiplication. Thus the largest number actually involved is $(n-1)^2$, which by previous discussion must be below 2^{13}.

Question 11: The source and header files are in Appendix B (files ___). The output of P2_1final.c (when the file P2_1message(encoded) is the encrypted message) is:

> alone and warming his five wits the white owl in the belfry sits