MACM 401/MATH 701/MATH 819
Assignment 2, Spring 2019.

Michael Monagan

Due Monday February 4th at 4pm. Please hand in to Dropoff box 1b outside AQ 4100
Late Penalty: -20% for up to 48 hours late. Zero after that.
For problems involving Maple calculations and Maple programming, you should submit a printout of a Maple worksheet of your Maple session.

Question 1 (15 marks): Univariate Polynomials

Reference section 2.5.

(a) Program the *extended* Euclidean algorithm for $\mathbb{Q}[x]$ in Maple. The input is two non-zero polynomials $a, b \in \mathbb{Q}[x]$. The output is three polynomials (s, t, g) where g is the *monic* gcd of a and b and $sa + tb = g$ holds.

Please print out the values of (r_k, s_k, t_k) that are computed at each division step so that we can observe the exponential growth in the size of the rational coefficients the r_k, s_k, t_k polynomials.

Use the Maple commands `quo(a,b,x)` and/or `rem(a,b,x)` to compute the quotient and remainder of a divided b in $\mathbb{Q}[x]$. Remember, in Maple, you must explicitly expand products of polynomials using the `expand(...)` command.

Execute your Maple code on the following inputs.

```
> a := expand((x+1)*(2*x^4-3*x^3+5*x^2+3*x-1));
> b := expand((x+1)*(7*x^4+5*x^3-2*x^2-x+4));
```

Check that your output satisfies $sa + tb = g$ and check that your result agrees with Maple's `g := gcdex(a,b,x,'s','t');` command.

(b) Consider $a(x) = x^3 - 1$, $b(x) = x^2 + 1$, and $c(x) = x^2$. Apply the algorithm in the proof of Theorem 2.6 (as presented in class) to solve the polynomial diophantine equation $\sigma a + \tau b = c$ for $\sigma, \tau \in \mathbb{Q}[x]$ satisfying $\deg \sigma < \deg b - \deg g$ where g is the monic gcd of a and b. Use Maple's `g := gcdex(a,b,x,'s','t');` command to solve $sa + tb = g$ for $s, t \in \mathbb{Q}[x]$ or your algorithm from part (a) above.

Question 2 (15 marks): Multivariate Polynomial Division

(a) Consider the polynomials

$$A = 6\,y^2x^3 + 2\,x^2y^2 + 5\,yx^2 + 3\,xy^2 + yx + y^2 + x + y \quad \text{and} \quad B = 2\,yx^2 + x + y.$$

Write $A \in \mathbb{Z}[y][x]$ and test if $B|A$ by doing the division in $\mathbb{Z}[y][x]$ by hand. Show your working. If $B|A$ determine the quotient Q of $A \div B$. Check your answer using Maple's `divide` command.

(b) Given two polynomials $A, B \in \mathbb{Z}[x_1, x_2, \ldots, x_n]$ with $B \neq 0$, give pseudo code for the multivariate division algorithm for dividing A by B. The pseudo code should begin like this

Algorithm DIVIDE(A,B)
Inputs $A, B \in \mathbb{Z}[x_1, x_2, \ldots, x_n]$ satisfying $B \neq 0$ and $n \geq 0$.
Output $Q \in \mathbb{Z}[x_1, x_2, \ldots, x_n]$ s.t. $A = BQ$ or FAIL meaning B does not divide A.

I suggest you start with my pseudo code for the division algorithm in $F[x]$ and modify it to work in $D[x_1]$ where $D = \mathbb{Z}[x_2, \ldots, x_n]$. The algorithm will make a recursive call to divide the leading coefficients in D. Because the algorithm is recursive you need a base of the recursion.

Question 3 (15 marks): The Primitive Euclidean Algorithm

Reference section 2.7

(a) Calculate the content and primitive part of the following polynomial $a \in \mathbf{Z}[x, y]$, first as a polynomial in $\mathbb{Z}[y][x]$ and then as a polynomial in $\mathbb{Z}[x][y]$, i.e., first with x the main variable then with y the main variable. Use the Maple command `gcd` to calculate the GCD of the coefficients. The `coeff` command will be useful.

```
> a := expand( (x^4-3*x^3*y-x^2-y)*(8*x-4*y+12)*(2*y^2-2) );
```

(b) By hand, calculate the pseudo-remainder $\tilde{r}$ and the pseudo-quotient $\tilde{q}$ of the polynomials $a(x)$ divided by $b(x)$ below where $a, b \in \mathbf{Z}[y][x]$.

```
> a := 3*x^3+(y+1)*x;
> b := (2*y)*x^2+2*x+y;
```

Now compute $\tilde{r}$ and $\tilde{q}$ using Maple's `prem` command to check your work.

(c) Given the following polynomials $a, b \in \mathbf{Z}[x, y]$, calculate the GCD$(a, b)$ using the primitive Euclidean algorithm with x the main variable.

```
> a := expand( (x^4-3*x^3*y-x^2-y)*(2*x-y+3)*(8*y^2-8) );
> b := expand( (x^3*y^2+x^3+x^2+3*x+y)*(2*x-y+3)*(12*y^3-12) );
```

You may use the Maple command `prem`, `gcd` and `divide` for the intermediate calculations.

Question 4 (20 marks): Chinese Remaindering and Interpolation

Reference section 5.3, 5.6 and 5.7

(a) By hand, find $0 \le u < M$ where $M = 5 \times 7 \times 9$ such that

$$u \equiv 3 \bmod 5, \quad u \equiv 1 \bmod 7, \quad \text{and} \quad u \equiv 3 \bmod 9$$

using the "mixed radix representation" for u and also the "Lagrange representation" for u.

(b) By hand, using Newton's method, find $f(x) \in \mathbb{Z}_5[x]$ such that $f(0) = 1, f(1) = 3, f(2) = 4$ such that $\deg_x f < 3$.

(c) Let $a = (9y - 7)x + (5y^2 + 12)$ and $b = (13y + 23)x^2 + (21y - 11)x + (11y - 13)$ be polynomials in $\mathbb{Z}[y][x]$. Compute the product $a \times b$ using modular homomorphisms ϕ_{p_i} then evaluation homomorphisms $\phi_{y=\beta_j}$ and $\phi_{x=\alpha_k}$ so that you end up multiplying in $\mathbb{Z}_p$. The Maple command `Eval(a,x=2) mod p` can be used to evaluate the polynomial $a(x, y)$ at $x = 2$ modulo p. Then use polynomial interpolation and Chinese remaindering to reconstruct the product in $\mathbb{Z}[y][x]$.

First determine how many primes you need and put them in a list. Use $P = [23, 29, 31, 39, ...]$. Then determine how many evaluation points for x and y you need. Use $x = 0, 1, 2, \ldots$ and $y = 0, 1, 2, \ldots$.

The Maple command for interpolation modulo p is `Interp(...)  mod p;`
The Maple command for Chinese remaindering is `chrem(...);`
The Maple command for putting the coefficients of a polynomial a in the symmetric range for $\mathbb{Z}_m$ is `mods(a,m);`

Question 5 (10 marks): Recurrence Relations

(a) Solve the recurrence relation $T(n) = T(n - 1) + 2n$ with initial value $T(1) = 1$.

(b) Below is a recursive Maple procedure that has as inputs an array A with $n > 0$ entries. Let $T(n)$ be the number of times the comparsion `A[i] = x` is executed. Write down a recurrence relation for $T(n)$ and a suitable initial value. You do not need to know what the procedure is doing to answer this question.

```
f := proc(A::Array,n::posint)
local i,x,flag1,flag2;
   if n=1 then return false; fi;
   flag1 := false;
   x := A[n];
   for i from 1 to n-1 do if A[i] = x then flag1 := true fi; od;
   flag2 := f(A,n-1);
   return flag1 or flag2;
end;
```