Elementi di teoria dei numeri
(liberamente tratte da Davenport, Aritmetica superiore, Zanichelli e da Delahaye, Stupefacenti numeri primi, Ghisetti e Corvi)

Le attività che dovete svolgere, completando questo file, sono segnate in rosso, così come i commenti alle vostre risposte (in verve quelle di Andrea e in blu quelle di Luca). Le parti evidenziate in giallo sono quelle che mettiamo maggiormente in evidenza per il prof. Impedovo, in modo che, se ne avrà il tempo e le riterrà interessanti di sviluppo, possa darci qualche indicazione in più rispetto a quelle da noi date. Le prime sette pagine (funzione indicatrice di Eulero inclusa) sono da svolgere entro il 10 Marzo. Le altre sono da completare entro il 26 Marzo. A mio avviso è opportuno un impegno piuttosto costante, anche se non è certo necessario lavorarci tutti i giorni. Vi suggerisco di iniziare al più presto e di chiedermi eventuali chiarimenti man mano che proseguite nello studio, senza attendere troppo. Poiché siete solo in due (Luigi sarà solo uno spettatore della classe virtuale) penso che la modalità di lavoro più agile sia quella della comunicazione in posta elettronica. I messaggi che ci scambieremo vanno inviati a tutti e tre (Domingo Paola, Luca Nucifora e Andrea Martinelli). Se avete piacere di svolgere parti in comune potete farlo, come preferite. Vi darò user id e password della classe virtuale quando passerete il lavoro che avrete svolto alla IIID e loro vi passeranno la parte sulle matrici. Dal 26 Marzo al 20 Aprile lavoreremo sul file che vi passeranno gli studenti di IIID (si tratterà solo di ripassare cose che già conoscete) e su alcuni semplici sistemi crittografici, in modo da prepararvi alle lezioni del 22 Aprile e del 6 Maggio (tutta la mattina dei due sabati) con il prof. Impedovo.

Lo scopo dell'aritmetica superiore è quello di indagare proprietà generali dei numeri interi e formulare proposizioni che le esprimano.
Esempi di tali proposizioni:
P1. Ogni numero naturale può essere fattorizzato in un unico modo (a meno dell'ordine dei fattori) in numeri primi.
P2. Esistono infiniti numeri primi.
P3. n e $n-1$ sono primi fra loro.
Provate a dimostrare P2. e P3.

(Andrea Martinelli) Dimostrazione P2
Per il teorema fondamentale del calcolo ogni numero naturale può essere fattorizzato in un unico modo. Utilizziamo una dimostrazione per assurdo, assumendo che esistano solo n primi $p_1, p_2, \ldots. p_n$. Consideriamo ora il numero a costituito dal prodotto di tutti i numeri primi:
$a=p_1 * p_2 * p_3 * ... * p_n$.
Consideriamo ora il numero b tale che $b = a + 1$.
A questo punto o b è primo o, per il teorema fondamentale del calcolo, è divisibile per un numero primo diverso da $p_1, p_2, \ldots. p_n$: Infatti b diviso per p_1 dà come resto 1; b diviso per p_2 dà come resto 1; …. ; b diviso per p_n dà come resto 1.
In ogni caso ci troviamo di fronte ad un nuovo numero primo, per cui dobbiamo rifiutare l'ipotesi che i numeri primi fossero solo $p_1, p_2, \ldots. p_n$.

(Andrea) Dimostrazione P3
Due numeri si dicono primi fra loro se non possiedono fattori primi in comune. Per il teorema fondamentale dell'aritmetica, $n-1$ può scomporsi in fattori primi:
$n - 1 = p_1 * p_2 * p_3 * ... * p_n$.
Quindi $n = p_1 * p_2 * p_3 * ... * p_n + 1$
Quindi nessuno dei fattori che compariva nella scomposizione del numero $n-1$ comparirà nella scomposizione di n (stesso ragionamento applicato nella precedente dimostrazione). Concludiamo quindi che i due numeri sono fra loro primi.

Il passaggio dall'aritmetica della scuola elementare all'aritmetica superiore consiste essenzialmente nel passaggio dalle osservazioni e sperimentazioni alle dimostrazioni. Qui daremo per scontate le caratteristiche principali degli insiemi dei numeri naturali e interi che avete studiato sul vostro libro di testo.

Numeri primi
Si dice primo ogni numero maggiore di 1 divisibile solo per 1 e per se stesso. P1, P2 e P3 sono tre proposizioni riguardanti i numeri primi.
Come fareste a costruire un programma che consenta di fattorizzare un numero? (Vi si chiede solo l'idea, non la costruzione di un programma vero e proprio)

(Andrea) Dato un numero naturale qualsiasi *n* maggiore di 2, si può impostare un ciclo che continui a dividere *n* per tutti i numeri naturali maggiori di 2 e minori della radice quadrata di *n* + 1. Se il resto di una di queste operazioni è uguale a zero il numero non è primo e si esce dal ciclo (è necessario quindi un ciclo ed una verifica della condizione resto=0). Se il ciclo viene completato senza che sia trovato un numero naturale in grado di dividere *n* con resto 0, allora *n* è primo. In prima abbiamo scritto un programma di questo tipo e quindi ne allego il "sorgente"....

```
:primo2(n)
:prgm
:if n=1 Then
:Disp "non è considerato primo"
:endif
:if n≠1 then
:if n=2 Then
:Disp "primo"
:endif
:if n≠2 then
:int (√(n))+1 → m
:2 → i
:While mod(n,i)≠0 and i<m+1
:i+1→i
:EndWhile
:if i=m+1 Them
:disp "primo"
:Else
:disp "non primo"
:endif
:endif
:endif
:endprogm
```

Prof: naturalmente questo programma non consente di fattorizzare un numero, ma di dire solo se è primo oppure no.
Avevo perso di vista la consegna iniziale (brutta cosa, lo so!). Direi che a questo punto le cose si complicano un po'…. Sinceramente io non saprei come fare (se si disponesse di una lista di numeri primi, si potrebbe provare a dividere il numero n assegnato per ciascuno dei numeri primi… peccato che questi siano infiniti!)…. Passo la palla a Luca….

Il problema è che non solo (ovviamente) non si possiede di una lista di tutti i numeri primi, ma fare liste (molto numerose) non è efficiente dal punto di vista computazionale, nel senso che per avere una lista di numeri primi bisogna considerare ogni numero e vedere se è primo; in caso affermativo metterlo in lista, in caso negativo scartarlo … Questo vorrebbe dire far girare il programma sopra scritto per ogni numero considerato e preparare un vettore che si riempie, via via, solo quando il numero è primo. Questo metodo, per quanto possa essere migliorato, dà un'idea del perché non ha molto senso utilizzare liste di numeri primi per scomporre in fattori un numero: è molto meglio dividere quel numero per tutti i numeri minori della parte intera della radice quadrata (+1) e, via, via, per ciascun fattore trovato:

a) se è primo inserirlo in una lista (quella dei suoi fattori primi)
b) se non è primo, richiamare il programma di scomposizione.

Luca, se lo ritieni utile puoi aggiungere qualcosa, ma mi sembra che tutto ciò renda abbastanza bene l'idea del dispendio di energie necessarie per scomporre un numero in fattori.

Un metodo molto ingegnoso è dovuto a Fermat. Sia N il numero da scomporre in fattori e sia m il più piccolo numero interro tale che $m^2 > N$. Consideriamo ora i numeri:
$m^2 - N$, $(m+1)^2 - N$, $(m+2)^2 - N$, …
Se ne incontriamo uno, diciamo y^2, che è un quadrato perfetto, abbiamo:
$x^2 - N = y^2$, ossia $N = (x - y)(x + y)$. Naturalmente non è detto che i due fattori trovati siano primi, ma, sicuramente, nel caso in cui si trovi in quadrato perfetto, il problema della fattorizzazione di N viene semplificato. Tra l'altro se N è primo, il procedimento continua fino a raggiungere la soluzione data da $x + y = N$ e $x - y = 1$.

I primi passi nella scomposizione in fattori di un numero non possono che essere compiuti iniziando a conoscere l'algoritmo di Euclide per la determinazione del massimo comune divisore tra due numeri.
Iniziamo con una domanda (che richiede una risposta!)

Quale fra le seguenti definizioni di *massimo comune divisore* (d'ora innanzi mcd) preferisci e perché?
a) Il mcd fra a e b è il prodotto di tutti i fattori primi comuni ad a e a b presi una sola volta con il minimo esponente con cui compaiono nella fattorizzazione di a e b in fattori primi
b) Il mcd fra a e b è il maggiore dei divisori comuni ad a e a b
c) d si dice mcd fra a e b se:

- d divide a e d divide b
- se e è un divisore comune ad a e a b, allora e divide d

d) d si dice mcd fra a e b se:

- d divide a e d divide b
- se e è un divisore comune ad a e a b, allora $e \leq d$

(Andrea) Scegliamo la prima definizione (a) perché l'unica che permette di calcolare in maniera operativa l'mcd. Le altre, oltre ad essere meno chiare, identificano piuttosto delle proprietà dell'mcd che indicare un rapido procedimento di calcolo.
Prof. In realtà la prima fornisce una procedura di calcolo computazionalmente poco efficiente. Pensate, per esempio, a quante operazioni sono necessarie per scomporre un numero in fattori. All'aumentare del numero di cifre la scomposizione, che in teoria è un problema banale, diventa in pratica impossibile, richiedendo tempi dell'ordine dell'età dell'universo (anche per calcolatori che lavorino in parallelo). Le altre due sono invece definizioni che oltre a essere considerate più chiare ed eleganti dai matematici (ma queste proprietà sono ovviamente soggettive) costituiscono la base dell'algoritmo di Euclide per il calcolo del MCD fra due numeri, algoritmo molto efficiente. Per chiarire perché, chiedete alla vostra calcolatrice di scomporre in fattori un numero di una

cinquantina di cifre e vedete un po' che cosa risponde; poi fate fare il mcd divisore di due numeri di una cinquantina di cifre e vedete la differenza… provate a farlo, dai!
Esperimento effettuato: ne sono convinto!

Dimostriamo ora che, dati due numeri a e b tali che $b > a$, se $d = \text{mcd}(a, b)$, allora $d = \text{mcd}(a, b - a)$
Se $d = \text{mcd}(a, b)$, allora $a = d * k$ e $b = d * h$. Quindi
$b - a = d * (h - k)$ con $h - k$ primo con h, quindi $\text{mcd}(a , b - a) = \text{mcd}(kd, d(h - k)) = d$ c.v.d.

(Andrea) Credo che valga la pena esplicitare un passaggio in più: $b - a = (d * h)-(d * k) = d * (h - k)$.

Ciò equivale a dire che $d = \text{mcd}(a, b - a)$.

Aprite ora il math box di TI-InterActive! e scrivete:
Define mcd(a,b) = when(a=b,a,when(a<b,mcd(a,b-a),mcd(b,a-b)))

Oppure scrivete la seguente funzione con la vostra calcolatrice TI-89:

```
:mcd (a,b)
:Func
:If a = b Then
:Return a
:EndIf
:If a < b Then
: mcd(a,b-a)
:Else
:mcd(b, a-b)
:EndIf
:EndFunc
```

Provate, in entrambi i casi, a calcolare a mano le operazioni che le due funzioni effettuano al comando mcd(12, 8). Utilizzate ora una delle due funzioni per calcolare il mcd(65870,398740).
Provate ora con numeri più grandi, e con la calcolatrice, per esempio con mcd(36965870,366398740).
Che cosa accade e perché?

[(Luca Nucifora) il programma mette in pratica il concetto appena dimostrato.
Nel primo e nel secondo caso la calcolatrice riesce a restituire come risultato il massimo comune divisore.
Il problema sorge quando si prova a usare numeri grandi: la calcolatrice restituisce un errore di memoria insufficiente.
Per l'esempio occorrono ben 81 somme algebriche prima di ottenere a=b.
È consigliabile quindi usare questo algoritmo per numeri piccoli, in modo da non dover effettuare un grande numero di calcoli.]
Funziona per gli esempi, ma il concetto non credo sia valido.
Luca, in una mail precisa la spiegazione sopra fornita:
ho fatto manualmente le operazioni che il programma fa
in pratica ho impostato al massimo la cronologia (99 operazioni)
e ho cominciato a fare:
366398740-36935870=329462870

329462870-36935870=292527000
292527000-36935870=255591130
255591130-36935870=218655260
218655260-36935870=181719390
181719390-36935870=144783520
144783520-36935870=107847650
107847650-36935870=70911780
70911780-36935870=33975910
36935870-33975910=2959960
33975910-2959960=31015950
....
e così fino a raggiungere 10-10 all'81esima operazione (in basso a destra c'è il contatore)
ci ho perso un po' di tempo ma ero curioso di sapere se funzionava..ed ero curioso di sapere il perché dava errore di memoria io me la sono spiegata così: la calcolatrice ha una certa quantità di memoria RAM, diciamo, che utilizza per far funzionare i programmi poi, come tutti questi apparecchietti, la memoria è spaccata in scompartimenti: se faccio le operazioni una a una funziona, perché la cronologia ha spazio più grande della memoria temporanea invece il programma occupa troppo spazio temporaneamente (me la sono spiegata così perché nel testo c'è scritto esplicitamente di usare la calcolatrice.. probabilmente su un computer con questi due numeri non ci sono problemi..)

Intervento del prof. Impedovo: sulla parte evidenziata in giallo: il problema secondo me è semplicemente fisico, di quantità di memoria disponibile, quindi mi pare che Luca abbia ragione. La ricorsione, come sapete, ad ogni chiamata del programma alloca una quantità di memoria necessaria alla iterazione successiva, crea un puntatore a questa area di memoria e prosegue nell'esecuzione del codice. E' ovvio che se le chiamate del programma superano una certa soglia, la calcolatrice non ha più memoria a disposizione e si blocca. Ho degli esempi interessanti a questo proposito, per esempio la definizione ricorsiva di coefficiente binomiale.

cb(n,k) :=if k=0 or k=n then 1
else cb(n-1,k-1)+cb(n-1,k)

qui la ricorsione è mostruosa, provate a contare quante chiamate di programma vengono effettuate anche per calcolare cb(10,5). Se la ricorsione è stupenda per il matematico, l'informatico la guarda come fosse il diavolo. Quando passate da mcd a mcd1 il numero di chiamate diminuisce fortemente (anziché fare tutte le sottrazioni ne fa una sola).

Prof. Ho fatto girare la funzione mcd(a,b) e poi quella sotto riportata mcd1(a,b) su derive e ha ottenuto che il primo programma si impianta (derive dà un messaggio di memoria esaurita) con numeri di circa 30 cifre, mentre il secondo funziona anche con numeri di 70 cifre (non ho provato oltre). Sotto allego il foglio di derive (dove non c'è output il programma dà "memoria esaurita")

```
      mcd(a, b) :=
        If a = b
           a
#1:        If a < b
              mcd(a, b - a)
              mcd(b, a - b)

#2:   mcd(12, 24)

#3:                                                          12

#4:   mcd(24, 12)

#5:                                                          12

#6:   mcd(12, 12)

#7:                                                          12

#8:   mcd(36965870, 366398740)

#9:                                                          10

#10:  mcd(369658709086754890011, 366398740238912356719675)

#11:                                                          3

#12:  mcd(3696587090867548900116789456341234134567892132121 1, 366398740238912356719675345634231987562311111)

#13:  mcd(369658709086754890011678945634123413456789, 36639874023891235671967534563423 19)

#14:  mcd(36965870908675489001167894563412341 3, 3663987402389123567196753456 3)

#15:  mcd(3696587090867548900116789456, 366398740238912356719 67)

#16:  mcd(369658709086754890011678 9, 36639874023891235671 9)

#17:                                                          3

      mcd1(a, b) :=
        If MOD(a, b) = 0
           a
#18:       If a < b
              mcd1(a, MOD(b, a))
              mcd1(b, MOD(a, b))

#19:  mcd1(3696587090867548900116789456341234134567892132121 1, 366398740238912356719675345634231987562311111)

#20:                                                         11

#21:  mcd1(369658709086754890011678945634123413456789213212113456738926 1,

        366398740238912356719675345634231987562311111345213179100 1)


#22:                                                          2
```

Cerchiamo un modo per migliorare la velocità di calcolo. Partiamo dalla considerazione che, dati *a* e *b* tali che *b* > *a*, se *d* =mcd(*a*, *b*) , allora *d* = mcd(*a*, mod(*b*, *a*)) dove con mod(*b*, *a*) abbiamo indicato l'operazione che restituisce il resto della divisione *b* : *a*.

Dimostrate innanzitutto quanto sopra affermato.
help: non ho neppure idea del perché questo funzioni!!!. (Andrea)
Non è difficile: è molto simile a prima.
Supponiamo che *a* non divida *b* (altrimenti mcd(*b*, *a*) = *a*).
Quindi *b* = *k***a* + *r* dove *k* è il quoziente della divisione *b* : *a* e *r* è il resto, ossia mod(*b*, *a*).
Ora, se *d* divide *a* e *b*, allora *a* = *t* **d* e *b* = *h* * *d*.
Quindi *r* = *b* – *k***a* = *d*(*h* – *k***t*), ossia … Riesci ora a metterla a posto?
Certo: a questo punto infatti abbiamo r= *d*(*h* – *k***t*) e a=t*d, quindi l'mcd fra i due è loro fattore comune preso con il minimo esponente, cioè proprio d.

Calcolando l'mcd fra a e b avremmo trovato di nuovo (come logico che sia) d, infatti $a = t * d$ e $b = h * d$. Questa proprietà risulta estremamente utile in quanto riduce il peso computazionale dell'operazione del calcolo del mcd.

Provate ora a scrivere il seguente programma con la TI-89:

```
:mcd1 (a,b)
:Func
:If a = b Then
:Return a
:EndIf
:If a < b Then
: mcd1(a,mod(b,a))
:Else
:mcd1(b, mod(a,b))
:EndIf
:EndFunc
```

Può funzionare? Perché?

(Luca) non può funzionare poiché a=b non è mai verificato.
a-b≠0 b-a≠0

Provate a scrivere invece questo programma:

```
:mcd1 (a,b)
:Func
:If mod(b,a)=0 Then
:Return a
:EndIf
:If a < b Then
: mcd1(a,mod(b,a))
:Else
:mcd1(b, mod(a,b))
:EndIf
:EndFunc
```

Che cosa è cambiato rispetto al precedente? Funziona? Provate ora a calcolare con la calcolatrice mcd(36965870,366398740). Si ottengono miglioramenti?
(Luca) Questo algoritmo utilizza il concetto appena dimostrato e come metodo di verifica il resto della divisione dei due numeri, invece che l'uguaglianza.
Quando il resto è uguale a zero, il divisore è il MCD dei due numeri iniziali.
(miglioramenti…è l'unico algoritmo che funziona per adesso…)

Intervento del prof. Impedovo: Sul mcd, vi suggerisco il più agile algoritmo seguente (è solo una semplificazione del vostro: non è necessario distinguere sull'ordinamento tra a e b)

```
mcd(a,b) :=if b=0 then a
else mcd(b,mod(a,b))
```

Teorema di Bachet (o di Bézout):
se d = mcd (a, b), allora esistono due numeri interi x e y tali che:
$ax - by = d$

Possiamo dire che a e b sono primi fra loro se e solo se esistono due interi x e y tali che:
$ax - by = 1$? Giustificate la risposta

(Andrea) Per la condizione imposta ($ax - by = 1$), l'mcd fra a e b deve essere uguale a 1. Questo significa che non esistono ulteriori fattori primi in comune fra a e b. E poiché 1 è divisore naturale comune a tutti i numeri, allora a e b devono essere primi.
In realtà ti sei limitato a dimostrare che se a e b sono primi fra loro , ossa SE mcd (a, b) = 1, ALLORA esistono x e y tali che $ax - by = 1$, utilizzando (correttamente) il teorema di Bézout. Ma non hai dimostrato che SE $ax - by = 1$, ALLORA mcd(a, b) = 1. Puoi completare la dimostrazione?
Per il teorema di Bézout, se d = mcd (a, b), allora esistono due numeri interi x e y tali che: $ax - by = d$. Quindi se $ax - by = 1$, allora d=1=mcd(a,b).
Attento in generale è errato. Cerchiamo di capire perché: considerate la proposizione
"se un triangolo è equilatero allora è isoscele"
Potete dedurre da essa la proposizione
"se un triangolo è isoscele allora è equilatero"?
Ovviamente no. Lo stesso è per il teorema (così come è stato enunciato) di Bézout. Dire che
SE mcd (a, b) = 1, ALLORA esistono x e y tali che $ax - by = 1$
Non consente, in generale, di affermare che
SE esistono x e y tali che $ax - by = 1$, ALLORA mcd (a, b) = 1
Possiamo effettuare, però, questa dimostrazione:
Se $ax - by = 1$, allora $ax = by + 1$. Quindi mcd(ax, by) = 1 (vedete la dimostrazione di P3 che ha fatto Andrea all'inizio). Ora, se, per assurdo, a e b avessero un fattore d in comune (d diverso da 1), avremmo che mcd(ax, by) = d, in contraddizione con quanto appena dimostrato. Quindi mcd(a, b) = 1

Teorema Se un numero a divide il prodotto bc e a e b sono primi fra loro, allora a divide c.
Dimostrate il precedente teorema (suggerimento: utilizzate il teorema di Bachet e moltiplicate entrambi i membri …)

(Andrea) Non so scrivere una dimostrazione vera e propria…. Però è abbastanza chiaro che se a non è fra i fattori che compongono il numero b (a e b sono fra loro primi, quindi nessun fattore in comune) e se a divide bc allora il fattore a deve essere contenuto in c. Quindi c è divisibile per a. Ora il problema sta nel scrivere ciò col il linguaggio formale di una dimostrazione matematica.
Seguiamo il suggerimento. Se mcd(a, b) = 1, allora esistono x e y tali che:
$ax - by = 1$. Moltiplichiamo entrambi i membri per c. Otteniamo
$acx - bcy = c$.
Poiché, per ipotesi, a divide il prodotto bc, abbiamo che $bc = k * a$.
Quindi $acx - k*ay = c$ Quindi $c = a$ *(Q) il che vuol dire che a divide c.
Se a e b non fossero primi fra loro e a dividesse il prodotto bc, potremmo dire che a divide c? Perché?
Nel caso a e b non siano primi avremo mcd(a,b)=d ed allora esisteranno $ax - by$ = d. Moltiplichiamo ancora una volta per c ed avremo: $acx - bcy = c$d. Ancora una volta abbiamo che a divide il prodotto bc, quindi scriviamo $acx - k*ay = cd$. Avremo che c=a*(Q)/d. Io direi che a continua a dividere c, infatti poniamo (Q)/d=(F) ed avremo c=a*(F).
Ti propongo il seguente controesempio:

Siani $a = 6$ e $b = 20$. Sia ora $c = 3$. $bc = 60$. Ovviamente a divide bc (6 divide 60), ma a non divide c (6 non divide 3). Come vedi, il tuo ragionamento è errato: riesci a trovare l'errore?
Vedete come è facile cadere in errore quando si effettuano dimostrazioni: è necessario un controllo logico (e magari non solo strettamente logico) molto molto forte.
In questo passaggio? "poniamo (Q)/d=(F) ed avremo c=a*(F).". Evidentemente non si può considerare Q/d come F poiché non è detto che F sia un numero naturale....

Congettura: se scrivo un numero di tre cifre e le ripeto, ottengo un numero che è divisibile per 7, 11 e 13. Provate, per esempio, con 123123 oppure con 135135 e così via…
Questa congettura è un teorema? Giustificate la risposta (suggerimento: $abcabc = abc * 1000 + abc$ … quindi….)

(Andrea) Dimostrazione banale: $abcabc = abc*1000+abc =$
$=abc*(1000+1)=abc*1001=abc*(13*11*7)$.

Prof: bene! Mi fermo qui, perché è importante che seguiate passo passo la lezione e, prima di andare avanti, mettiate a posto le parti lasciate in sospeso.

Teorema: mcd(a, b) * mcm(a, b) = $a*b$ dove si è indicato con mcm(a, b) il minimo comune multiplo dei numeri a e b.
Con quanto avete imparato fino a ora, come determinereste mcm(9876543910, 723789100)? Giustificate la risposta.

(Luca) Si può determinare facendo (a*b) / mcd(a,b).
ecco il codice per un possibile programma per la calcolatrice TI-89

```
mcm(a,b)
Prgm
(a*b)/mcd(a,b)→c
Disp c
DelVar c
EndPrgm
```

[Prendendo come mcd l'ultimo algoritmo provato]

Aiutandovi anche con quanto scritto sul sito web, http://alpha01.dm.unito.it/personalpages/cerruti/cp0/crittoprimistart.html scrivete due tre paginette sull'aritmetica modulare, in modo che siano comprensibili da uno studente di terza liceo scientifico. Eventualmente potete anche indicare indirizzi di siti web dove tale argomento sia spiegato in forma semplice e chiara, almeno per comprendere quanto scritto qui di seguito:

(Andrea)
L'aritmetica modulare è proprio l'aritmetica che alla base degli orologi. Per capire come funziona può essere utile immaginare alcuni esempi con un orologio: se la mia lancetta oraria (io d'ora in poi con "lancetta" intenderò sempre e solo quella) segna le 4, per eseguire l'operazione di addizione mi sufficiente spostarla in senso orario di 2 ore. Ho che 4+2=6. Analogamente, ruotandola in senso antiorario, avrei avuto 4-2=2. E se io invece avessi voluto portare indietro la mia lancetta di 7 ore? Sarei andato a posizionarmi sulle 9, fatto che significa che 4-7=9.
Questi risultati sembrano strani se non si nota il seguente fatto: se noi aggiungiamo, o togliamo, 12 torniamo al punto di partenza, poiché questo equivale a fare un giro completo attorno al nostro

orologio! Quindi 2+12=2, o anche 2-12=2: possiamo aggiungere o sottrarre 12 a piacere senza che il risultato della sottrazione venga in qualche modo inficiato.
A questo punto possiamo riscrivere l'operazione precedente: 4-7=-3+12=9.
Quello che stiamo facendo si chiama aritmetica modulo 12, poiché addizione e sottrazione sono insensibili ai multipli di 12.
Per semplicità, per ora considereremo solo, nelle operazioni modulari, i numeri naturali (l'estensione ai numeri interi, una volta capito come le cose funzionano nei naturali non dovrebbe comportare alcun problema): nell'esempio precedente, ciò corrisponde a considerare solo rotazioni in senso orario della lancetta dell'orologio.

In generale, per effettuare delle operazioni in modulo, bisognerà procedere in questo modo:

* Effettuare il calcolo secondo l'aritmetica normale;
* Dividere il risultato per il modulo (n);
* Il resto ottenuto corrisponderà al risultato (mod n).

Esempio:

Calcolare 11 * 99 mod 13.

Effettuiamo il calcolo secondo l'aritmetica normale ed otteniamo:

11 * 99 = 1089;

Dividiamo il risultato per il modulo:

1089 : 13 = 83 con un resto di 10 (visto che 13 * 83 = 1079);

Il risultato finale è quindi:

11 * 99 = 10 mod 13

Questo processo viene anche chiamato riduzione in modulo. In pratica, sottraendo il modulo (e tutti i multipli del modulo) un numero viene "ridotto" in un numero più piccolo del modulo.
Nell'esempio precedente, quando il numero 1089 viene "ridotto" a 10 si potrebbe dire che "1089 viene ridotto modulo 13 a 10".

Questo vuol dire che 1089 è congruo a 10 modulo 13 e si scrive

$1089 \equiv 10 \pmod{13}$

Ovviamente ci sono infiniti numeri congrui a 10 modulo 13: tutti quelli che, divisi per 13, danno come resto 10, ossia tutti i numeri del tipo
k* 13 + 10 con k numero naturale.
Consideriamo ora un qualunque numero naturale a; al variare di a quanti resti si possono ottenere dividendo a per 13? Per la definizione di resto (numero naturale minore del divisore), la risposta è immediata: sono i tredici numeri: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12.
Ora è chiaro che ci saranno infiniti numeri naturali che divisi per 13 danno come resto 0, infiniti che divisi per 13 danno come resto 1 e così via. I matematici amano effettuare riduzioni significative: tutti i numeri naturali che divisi per 13 danno come resto 0 li mettono in uno stesso insieme (o classe, come si dice anche); analogamente mettono in una stessa classe tutti i numeri che divisi per

13 danno come resto 1 e così via, ottenendo un insieme di 13 classi, detto, insieme delle classi di resto modulo 13 o, più semplicemente, ma con meno rigore, insieme dei resti (modulo 13).
Una classe di resto modulo 13 si indica con una scrittura del tipo $[x]$.
Per esempio alla classe di resto [0] (modulo 13) appartengono tutti i multipli i multipli di 13:
[0] è la classe di resto (o di equivalenza) di tutti i numeri del tipo $k * 13$
[1] è la classe di resto (o di equivalenza) di tutti i numeri naturali del tipo $k * 13 + 1$
e così via.
Si può allora considerare l'insieme delle classi di resto modulo 13, che si indica con la scrittura $\mathbf{Z}_{13}$:
$\mathbf{Z}_{13}$ = {[0], [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12]}.
(Talvolta si usa per $\mathbf{Z}_{13}$ anche l'espressione insieme quoziente $\mathbf{Z}/13\mathbf{Z}$)

In generale un'aritmetica modulo n opera sui possibili resti che si ottengono dividendo un numero naturale a per n e cioè sugli n elementi 0, 1, 2, …., $n - 1$. Un qualunque elemento x di tale insieme è una classe di equivalenza, ossia rappresenta tutti i numeri naturali che, divisi per n danno come resto x.
$\mathbf{Z}_n$ {[0], [1], [2], …, $[n - 1]$} è l'insieme dei resti modulo n.
Due interi a, b (supponiamo per semplicità $b > a$) stanno nella stessa classe dei resti modulo n se e solo se
$b = k*n + a$
In tal caso si dice che a è congruo a b modulo n e si scrive, come già detto,
$a \equiv b \pmod{n}$
Si noti che la condizione $b = k*n + a$
equivale a dire che $b - a = k*n$, ossia equivale a dire che n divide $a - b$.

Per esempio, 60 e 34 appartengono alla stessa classe di resto modulo 13, la classe del resto 8.
Infatti
$60 = 4*13 + 8$
$34 = 2*13 + 8$
Quindi $60 \equiv 34 \equiv 8 \pmod{13}$.

Come abbiamo visto, La divisione con resto, per come è stata definita, assicura che per ogni numero intero x esiste uno e un ben determinato numero naturale r compreso tra 0 e m-1 che è congruo a x modulo m.
Ciò equivale a dire che ogni classe di equivalenza della relazione *"essere congrui modulo* m" contiene uno e un solo elemento dell'insieme {0,1,2,3,....,m-1}.
Abbiamo detto che tale insieme viene detto anche *insieme dei resti*. Ciascun elemento r dell'insieme è il resto comune a tutti quei numeri che divisi per m danno come resto r.
Si ha allora che l'insieme quoziente $\mathbf{Z}/m\mathbf{Z}$ che indichiamo con $\mathbf{Z}_m$ contiene m classi:
$\mathbf{Z}_m$ = {[0],[1],[2],....,[m-1]}

L'idea è quella di dotare di struttura algebrica, così come si è fatto per gli ordinari insiemi numerici, anche le classi di resti, ossia gli insiemi $\mathbf{Z}_m$.
È possibile dimostrare che
$(x \equiv x' \ \ e \ \ y \equiv y') \ \Rightarrow (x + y \equiv x' + y' \ \ \ e \ \ xy \equiv x'y')$
Allora è possibile definire anche in $\mathbf{Z}_m$ le operazioni di addizione e moltiplicazione ponendo:
$[x] + [y] := [x+y]$ e $[x]\cdot[y] := [xy]$
Le definizioni date sono ben poste, nel senso che le classi di equivalenza che compaiono ai secondi membri non dipendono dagli elementi scelti per rappresentare le classi di cui si definiscono somma e prodotto.

Andrea: non mi è chiaro quanto scritto nelle precedenti righe.

Si afferma che se si considera dei numeri congrui modulo n (per esempio 60 e 8 che sono congrui modulo 13 e 34 e 8 che sono congrui modulo 13), anche la loro somma e il loro prodotto saranno congrui modulo 13 (in altri termini, anche 60 + 34 è congruo a 8 + 8 modulo 13 e così 60*34 è congruo a 8*8 modulo 13).
Questa proprietà suggerisce che se si deve fare la somma tra due classi di resto, la si può fare considerando qualunque elemento delle classi. Ossia, se si deve fare x + y, puoi prendere qualunque numero x' che sia congruo a x e qualunque numero y' che sia congruo a y e si ottiene lo stesso risultato: la somma e il prodotto non dipendono dai particolari elementi delle classi di equivalenza che si sceglie. Sei abituato a farlo con le frazioni.
Se si deve addizionare 1/5 + 1/3 si può anche fare 3/15 + 5/15 e scrivere 8/15. Chi assicura che si possa fare così? Il fatto che la somma tra numeri razionali rappresentati sotto forma di frazioni non dipende dalle particolari frazioni che si considerano nell'insieme di frazioni equivalenti. Poiché 1/5 è equivalente a 3/15 e 1/3 è equivalente a 5/15 si può effettuare la somma fra 3/15 e 5/15. Lo stesso con le classi di resto.

Andrea: Dalle ipotesi che x = x' (mod n) e y = y' (mod n) segue che:
x diviso n e x' diviso n hanno stesso resto, diciamo r; y diviso n e y' diviso n danno stesso resto, diciamo r'
Quindi
x = k*n + r y = p * n + r'
x' = h*n + r y' = q * n + r'
Da cui x + y = (k+p)*n + r + r' e x' + y' = (h+q)*n+r+r' Quindi x'+y' congruo x+y mod (n) poiché hanno entrambi lo stesso resto.
Analogamente per x*y

Che cosa fa il seguente programma scritto per la calcolatrice TI-89? Provatelo e poi rispondete:

```
:moz4()
:Prgm
:For j,0,3
:For i,0,3
:Disp mod(i*j,4)
:Pause
:EndFor
:EndFor
:EndPrgm
```

(Luca) prima di rispondere alla domanda vorrei sapere se ha senso avere un resto di qualsiasi divisione che abbia come divisore o dividendo 0.
L'istruzione mod ha delle convenzioni speciali: per esempio mod(n,0) restituisce n, mod(0,n) restituisce 0.

Il problema è questo: matematicamente non ha senso dividere per 0. Infatti se esistesse x tale che 5:0 = x, vorrebbe dire che x*0 = 5. Invece 0:5 = 0. Quindi dire che mod(0,n) = 0 vuol dire che *n* divide esattamente 0 e ciò è coerente con la definizione di divisione, infatti 0 = 0*n + 0. Invece mod(n,0) =n non è compatibile con la definizione matematica. Bisognerebbe dire che non è possibile dividere per 0 e quindi non ha senso chiedersi quale è il resto della divisione n:0. In

informatica può però essere utile dare comunque una risposta . Affermare che il resto della divisione di n per 0 è n è come dire che non si divide n per 0. Comunque questo non è un problema per interpretare il programma contenuto nel testo.
Manca la risposta di Luca e quindi la fornisco io: il programma, con due cicli annidati, dà la tabella della moltiplicazione modulo 4. Provare per credere.

Intervento del prof. Impedovo: il programma precedente dà tutti i valori degli elementi di Z_4, $\cdot$, ossia della tabellina di moltiplicazione modulo 4, ma non organizzati in tabella. Se si usa il seguente programma invece si hanno i risultati organizzati in una vera e propria tabella:

```
: m4( )
: Prgm
: Disp seq(seq(mod(i*j,4),i,0,3),j,0,3)
: EndPrgm
```

La funzione seq genera una lista e quindi seq(seq(genera una lista di liste, ossia una matrice.

Nelle aritmetiche modulari vale il seguente teorema:
Se i numeri interi *a* e *n* sono primi fra loro, allora *a* possiede un inverso modulo *n*.

Teorema. In Zn l'elemento [a] ha un inverso se e solo se a è primo con n.
Il teorema esprime una condizione necessaria e sufficiente, ossia del tipo A se e solo se B

Iniziamo a dimostrare la prima implicazione (se A allora B), ossia che se [a] ha un inverso, allora a e n sono primi tra loro.
Ipotesi: [a] ha un inverso, ossia esiste [b] tale che [a] * [b] = 1 (ovviamente "modulo n)
Tesi: mcd(a, n) = 1
Partiamo quindi dall'ipotesi e vediamo di aggiungere, mediante conseguenze logiche delle ipotesi alla tesi.
Poiché esiste [b] tale che [a] * [b] = 1 (ipotesi), allora esiste un numero k tale che a*b = 1 + n *k (ossia la divisione, nei numeri naturali, di a*b con n dà come resto 1). Ciò equivale a dire che a*b - n*k = 1 e abbiamo già dimostrato che due numeri p e q sono primi fra loro se e solo se esistono due interi x e y tali che px - qy = 1.
Quindi possiamo concludere che, poiché a*b - n*k = 1 , a e n sono primi fra loro, ossia mcd(a, n) = 1, che è la tesi.

Dimostriamo ora la seconda implicazione (se B allora A), ossia che se a e n sono primi fra loro, allora [a] ha un inverso.
Ipotesi: mcd(a, n) = 1
Tesi: esiste [b] tale che [a] * [b] = 1
Dall'ipotesi, ossia dal fatto che mcd(a, n) = 1 discende, per il teorema di Bachet (o di Bézout) che esistono b e k tali che
a*b + k*n = 1, il che vuol dire che, in Zn si ha che [a*b] = 1, ossa [a] * [b] = 1, il che vuol dire che [b] è l'inverso di [a]. Ciò conclude la dimostrazione.

Utilizzando il teorema appena enunciato dimostrate che in Z_p, con *p* primo, tutti i numeri 1,2,3, …, *p* – 1 ammettono inverso.

Se p è primo, tutti i numeri 1,2,3,…p-1 sono coprimi con p e questa è la condizione sufficiente per cui sia possibile calcolare l'inverso.

È vero che in Z_n, qualunque sia n dati due numeri a e b tali che $a * b = 0$ allora o $a = 0$, oppure $b = 0$? Giustificate la risposta aiutandovi sia con le esplorazioni che vi permette il precedente programma, sia con i teoremi presi in considerazione.

Non necessariamente: è sufficiente che n sia un sottomultiplo del prodotto a*b con n non primo. Se n è primo allora dati due numeri a e b tali che $a * b = 0$ allora o $a = 0$, oppure $b = 0$.

Esempio:
6*4 = 24 = 0 mod(12) ma sia 6, sia 4 sono diversi da 0.
(Andrea)

Nel 1640 il matematico francese Pierre de Fermat enunciò una congettura che riveste una notevole importanza per i nostri scopi, essendo all'origine di molti test di primalità. Tale congettura fu poi dimostrata nel 1736 dal matematico svizzero Leonhard Euler, anche se è ancora nota come "piccolo teorema di Fermat" (da non confondersi con l'ultimo teorema di Fermat dimostrato da Andrews Wiles nel 1995).

Piccolo teorema di Fermat:
Per ogni intero a che non sia multiplo di un numero primo p si ha che $a^{p-1} \equiv 1 \bmod(p)$
Dimostrazione:
Sia a un numero che non sia multiplo del numero primo p. La funzione f che fa corrispondere al numero n, tale che $0 \leq n \leq p-1$, il prodotto $na \pmod p$ manda l'insieme $\{0, 1, \ldots, p-1\}$ in sé stesso. Due numeri n e m che sono diversi (modulo p), hanno anche immagini diverse mediante f. Infatti moltiplicando per l'inverso di a entrambi i membri della disuguaglianza $an \neq am \pmod p$ si ottiene ancora una disuguaglianza: $n \neq m \pmod p$ (l'inverso di a esiste perché p è primo e a non è un suo multiplo).
Ne deriva che moltiplicando per a gli elementi dell'insieme $\{1, \ldots, p-1\}$, si ottengono tutti elementi fra loro distinti che formano l'insieme $\{a, \ldots, a(p-1)\}$. Naturalmente, per quanto detto prima sulla funzione f, tali elementi appartengono all'insieme $\{1, \ldots, p-1\}$, pertanto i due insiemi coincidono, ossia sono formati dagli stessi elementi (ciò che può mutare è l'ordine degli elementi, cosa non importante negli insiemi).
Quindi
$1 \cdot 2 \cdot 3 \cdot \ldots \cdot (p-1) \equiv a \cdot 2a \cdot 3a \cdot \ldots \cdot (p-1)\, a \pmod p$
Semplificando successivamente per 2, 3, $(p-1)$, il che è possibile perché ogni numero ammette inverso modulo p essendo p primo, si ottiene
$1 \equiv a^{p-1} \pmod p$ c.v.d.

Dimostrate che il teorema di Fermat può essere enunciato nella seguente forma:
Sia p un numero primo. Per ogni intero a , si ha che: $a^p \equiv a \pmod p$

(Andrea) Dimostrazione banale: è sufficiente moltilicare entrambi i membri di $1 \equiv a^{p-1} \pmod p$ per a si ottiene: $a^p \equiv a*\bmod(p)$.

Dal piccolo teorema di Fermat si deduce che se p è primo, allora $2^p \equiv 2 \pmod p$. In alcuni testi di storia della matematica si dice che alcuni matematici dell'antica Cina si è scritto che
p è primo se e solo se $2^p \equiv 2 \pmod p$.
Perché questa affermazione, se fosse vera, sarebbe molto importante per le ricerche sui numeri primi?

(Andrea) Sarebbe molto importante perché permetterebbe di verificare se un numero è primo con una semplice operazione con un bassissimo costo computazionale. Gli altri algoritmi per i test di primalità richiedono invece un grandissimo numero di operazioni.

Per dimostrare che la proposizione "p è primo se e solo se $2^p \equiv 2 \pmod{p}$." è falsa, basta trovare un controesempio. Il più piccolo intero che costituisce un controesempio è $p = 341$ che non è primo, perché si fattorizza in $11 \cdot 31$, ma $2^{341} \equiv 2 \pmod{341}$.
In effetti esistono infiniti controesempi e per ogni a, anche se sono piuttosto rare. Quindi dal piccolo teorema di Fermat non si può dedurre alcuna caratterizzazione elementare dei numeri primi.
Il piccolo teorema di Fermat viene utilizzato per i test probabilistici che consentono di determinare se un numero è o non è primo con un determinato livello di fiducia che si esprime in termini probabilistici.

Il teorema cinese del resto.
Siano dati due numeri interi positivi m ed n primi fra loro (ossia con MCD(m , n) = 1).
Comunque siano assegnati due resti r, s ($0 \leq r < m$; $0 \leq s < n$), è possibile trovare un numero intero x che nella divisione per m dia resto r e nella divisione per n dia resto s.

Dimostrazione.
Si tratta di risolvere il sistema:

(1) $$\begin{cases} x = r \pmod{m} \\ x = s \pmod{n} \end{cases}$$

ossia di vedere se esistono interi x; y; z tali che:

$$\begin{cases} x = r + my \\ x = s + nz \end{cases}$$

Sottraendo membro a membro si ottiene:

(2) $$my - nz = s - r$$

Indichiamo con la coppia di interi $(y_0 ; z_0)$ una soluzione di questa equazione.
Questo vuol dire che

(3) $$my_0 - nz_0 = s - r \ .$$

Ovviamente se $(y_0 ; z_0)$ è una soluzione, tale è anche ogni coppia del tipo $(y_0 + nk; z_0 + mk)$, come si dimostra facilmente sostituendo in (2) al posto di y il termine $y_0 + nk$ e al posto di z il termine $z_0 + mk$:

$m(y_0 + nk) - n(z_0 + mk) = my_0 - nz_0 = s - r$.
Quindi $(y_0 ; z_0)$ è soluzione se e solo se lo è ogni coppia del tipo $(y_0 + nk; z_0 + mk)$.

Allora per ottenere una soluzione x nel sistema (1) basta sostituire y e z, rispettivamente, con $y_0 + nk$ e $z_0 + mk$.
Otteniamo:

$x = r + my_0 + mnk$; ossia $x = r + my_0 \pmod{mn}$.
$x = s + nz_0 + nmk$, ossia $x = s + nz_0 \pmod{nm}$

Quindi la soluzione generale è una classe di congruenza modulo mn.

Esempio:
abbiamo degli oggetti i cui non conosciamo il numero, ma sappiamo che:

a) se li contiamo a gruppi di 3 ne restano 2
b) se li contiamo a gruppi di 5 ne restano 3
c) se li contiamo a gruppi di 7 ne restano 2

Quanti oggetti ci sono?
Dobbiamo trovare un numero x tale che:
$x = 2 \pmod 3$ $x = 3 \pmod 5$ $x = 2 \bmod(7)$
Ripercorriamo la dimostrazione, anche per capirla meglio:

$\begin{cases} x = 2 + 3y \\ x = 3 + 5z \\ x = 2 + 7t \end{cases}$ da cui si ottiene $\begin{cases} 3y - 7t = 0 \\ 3y - 5z = 1 \end{cases}$ (*)

Determiniamo una terna di soluzioni intere $(y_0; z_0; t_0)$. Abbiamo: (7; 4; 3)
Per ottenere x si può sostituire nel sistema $\begin{cases} x = 2 + 3y \\ x = 3 + 5z \\ x = 2 + 7t \end{cases}$ ottenendo $x = 23$. Il teorema cinese del resto afferma che una soluzione generale è una classe di congruenza modulo 3*5*7 = 105.
Quindi $x = 23 + k * 105$

La funzione indicatrice di Eulero
Una delle funzioni più utilizzate per lo studio dei numeri interi è la funzione φ indicatrice di Eulero, ossia la funzione che, per ogni numero n, restituisce il numero di interi che precedono n e che sono primi con n.
A questo proposito leggete anche quanto scritto all'indirizzo web http://alpha01.dm.unito.it/personalpages/cerruti/cp0/crittoprimistart.html dove potete trovare anche relazioni tra la funzione di Eulero e il piccolo teorema di Fermat.

Calcolate $\varphi(5)$, $\varphi(10)$ *e* $\varphi(20)$

Andrea
$\varphi(5) = 4$
$\varphi(10) = \{1,3,7,9\} = 4$
$\varphi(20) = \{1,3,7,9,11,13,17,19\} = 8$

Teoremi sulla funzione di Eulero

1) Se n e m sono due numeri interi maggiori di 0 e primi fra loro, allora $\varphi(n \cdot m) = \varphi(n) \cdot \varphi(m)$
2) Se n è una potenza k – esima di un numero primo p (ossia $n = p^k$), allora $\varphi(n) = p^k \left(1 - \frac{1}{p}\right)$
3) Se n si scompone nei fattori primi q_1, q_2, …, q_m, allora $\varphi(n) = n\left(1 - \frac{1}{q_1}\right)\left(1 - \frac{1}{q_2}\right) \cdot \ldots \cdot \left(1 - \frac{1}{q_m}\right)$

Provate a dimostrare i teoremi 2) e 3) utilizzando il teorema 1).

Andrea: 2)
Non mi tornano i conti: infatti se $\varphi(p^k)$ io potrei scrivere $\varphi(p)_1 * \varphi(p)_2 * \ldots * \varphi(p)_k = (p-1)_1 * (p-1)_2 * (p-1)_3 * \ldots * (p-1)_k = (p-1)^k$.

Non ho idea di come si possa quindi far venire fuori $\varphi(n) = p^k (1-\frac{1}{p})$

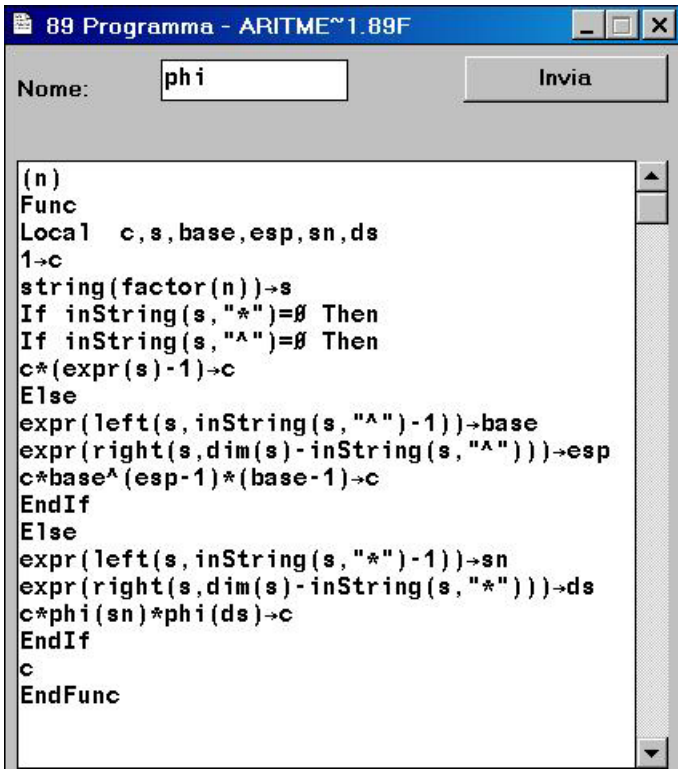
3) Capita la prima scrittura dovrebbe essere possibile completare questa dimostrazione.
(Andrea)

Prof. Intanto potresti iniziare a effettuare un po' di esplorazioni, per acquisire esperienza e verificare che "i conti tornano". Sia $n = 2^3$. Secondo il teorema 2 dobbiamo avere $\varphi(8) = 2^3 (1-\frac{1}{2}) = 4$, come puoi facilmente verificare. Prova a farlo con qualche altro esempio, in modo da acquisire più fiducia nella validità del teorema (che non sia una fiducia del tipo "lo ha detto il prof." O "sta scritto sul libro"). Una volta che si acquisisce fiducia si è psicologicamente più disponibili a cercare e a condividere una dimostrazione che spieghi *perché* il teorema "funziona". Ecco la dimostrazione: poiché, per definizione, $\varphi(p^k)$ restituisce il numero di interi che precedono p^k e che sono primi con p^k, non dobbiamo considerare tutti i numeri compresi tra 0 e $p^k - 1$ che hanno un fattore comune con p^k. Poiché p è primo, tali numeri da non considerare sono tutti e soli i multipli di p distinti da p^k, ossia $0, p, 2p, 3p, \ldots, p^k - p$. Nota che essi sono tutti i numeri del tipo: pt con t appartenente all'insieme $\{0, 1, 2, 3, \ldots, p^{k-1} - 1\}$, che ha, ovviamente, p^{k-1} elementi. Ora, il numero rotale dei dei numeri interi di p^k costituisce l'insieme $\{, 1, 2, 3, \ldots, p^k - 1\}$ che ha, ovviamente, p^k elementi.
Quindi $\varphi(p^k) = p^k - p^{k-1} = p^k (1-\frac{1}{p})$ c.v.d.

Per dimostrare che se n si scompone nei fattori primi $q_1, q_2, \ldots, q_m$, allora $\varphi(n) = n\left(1-\frac{1}{q_1}\right)\left(1-\frac{1}{q_2}\right)\cdot\ldots\cdot\left(1-\frac{1}{q_m}\right)$, basta applicare il teorema 1), ossia $\varphi(q_1 \cdot q_2 \cdot \ldots \cdot q_m) = \varphi(q_1) \cdot \varphi(q_2) \cdot \ldots \cdot \varphi(q_m)$ e, a ciascun termine $\varphi(q_1)$, $\varphi(q_2)$, …, $\varphi(q_m)$ applicare il teorema 2.

Intervento del prof. Impedovo:
Per quel che riguarda i teoremi su $\varphi(n)$, mi piacciono di più nella seguente forma, senza tirare in ballo scritture con frazioni (perché sono tutti numeri naturali):
1) se mcd(m,n)=1 allora $\varphi(n*m) = \varphi(n) * \varphi(m)$
2) se p è primo allora $\varphi(p^k) = p^{k-1} * (p-1)$

Sollecitato da questi problemi, ho scritto un programma, anzi una function, per la TI-89 che calcola in modo ricorsivo, $\varphi(n)$. E' molto efficiente, nel senso che sfrutta proprio i teoremi che conoscete su φ e calcola φ di numeri anche molto grandi in poco tempo. Se poi n è primo, la risposta è immediata.



La relazione di congruenza fra interi consente di enunciare (lo facciamo senza dimostrarla) una condizione necessaria e sufficiente affinché un numero p sia primo:

Teorema di Wilson

condizione necessaria e sufficiente affinché p sia primo è che $(p-1)! \equiv p-1 \pmod{p}$

Quello che segue è il lavoro da completare assolutamente entro il 26 Marzo (potremo dedicarvi anche due mercoledì di lavoro in classe. Sono d'accordo con il prof. Romeni che la prima ora del mercoledì potete lavorare con me). Iniziate comunque a lavorarci al più presto. Questa volta non suddividetevi il lavoro, ma rispondete entrambi a tutte le domande ed effettuate entrambi tutte le attività richieste.

Ricerche sui numeri primi

Vogliamo ora dare un'idea, compatibilmente con quelle che sono le conoscenze che attualmente possedete, di due filoni della ricerca matematica sui numeri primi:

a) la ricerca di formule che generano solo numeri primi
b) lo studio della distribuzione dei numeri primi.

Diciamo subito che, a un primo esame, la suddivisione in due filoni può apparire alquanto singolare: in fondo, se si riuscissero a trovare formule che generano solo numeri primi, si risolverebbe anche il secondo problema. Il fatto che esista un filone di ricerca anche sul punto b) suggerisce quindi che le ricerche relative al filone a) non abbiano portato a risultati definitivi, almeno in senso positivo.

Partiamo dalla ricerca di formule polinomiali che generino solo numeri primi, ma prima di tutto introduciamo due funzioni, disponibili su un qualunque manipolatore simbolico e, in particolare su TI-InterActive! e sulla vostra calcolatrice, che consentono di dire se un numero è o non è primo.

La prima funzione è "factor(n)" sia in Ti-InterActive!, sia con la TI-89. Questa funzione, applicata a un numero n restituisce la sua scomposizione in fattori primi. Il problema è che la scomposizione in fattori è un'operazione i cui tempi di calcolo aumentano in modo esponenziale all'aumentare del numero di cifre. Provate per esempio a scomporre il numero 67004171689247114711 in fattori con la calcolatrice scrivendo factor(67004171689247114711) e battendo enter. Vi accorgerete che il

tempo di calcolo non è brevissimo. Molto meglio con TI-InterActive! che, girando su un PC può utilizzare risorse di memoria molto più potenti. Il numero immesso è di 20 cifre e, per i numeri che considereremo in questa attività, le risorse di calcolo messe a disposizione dalla calcolatrice saranno più che sufficienti, quindi potremo chiedere tranquillamente la scomposizione in fattori primi di un numero. Con numeri il cui numero di cifre aumenti di uno o due ordini di grandezza (100 – 1000 cifre), le cose non sarebbero più così semplici. La scomposizione rischia di essere impraticabile, in tempi accettabili, anche su un potente PC. È allora bene sapere che esiste un'altra funzione che non dà la scomposizione in fattori, ma costituisce un test di primalità efficiente, nel senso che risponde alla domanda "il numero *n* è primo?" con un "vero" o "falso", a seconda che *n* sia primo o composto' in tempi molto più ragionevoli. Per convincervene provate a scrivere sulla vostra calcolatrice IsPrime(67004171689247114711) e vedrete che immediatamente la calcolatrice restituisce "False": ciò vuol dire che 67004171689247114711 non è primo, ma composto, anche se l'informazione non dice alcunché sulla sua scomposizione in fattori.
Provate ora con un numero molto più grande, ossia
501437616158417197799899992729091516461044745837369419636403861262 31
Vedrete che con la TI-89 la fattorizzazione è in pratica improponibile, mentre la funzione IsPrime restituisce il risultato False, anche se il tempo di calcolo è considerevole a causa della limitata memoria della TI-89. Con Ti- interActive la funzione IsPrime dà una risposta immediata, menrtre la fattorizzazione richiede un tempo senza dubbio apprezzabile.
Naturalmente questo guadagno di tempo non è a costo zero: in effetti il test ora utilizzato mediante la funzione IsPrime(*n*) (sia su TI-89 che su TI-InterActive!) è un test che dà una risposta certa solo nel caso in cui affermi che il numero è composto (ossia quando la risposta è "False"), mentre quando restituisce la risposta "True" (ossia "Vero") possiamo solo stimare la probabilità che questa risposta sia corretta. Dovrebbe consolare il fatto che questa probabilità può essere resa grande a piacere (ossia vicina a 1 quanto si vuole), naturalmente con tempi che aumentano all'aumentare del grado di fiducia, ma sempre gestibili. Non so quanto sia la probabilità di sbagliare con il test di primalità della TI-89, ma non penso di prendere grosse cantonate affermando che è inferiore all' 1%, il che vuol dire che su 100 numeri riconosciuti primi dalla calcolatrice , uno è in realtà composto. Vedremo che questa limitazione non costituirà un problema nello studio della distribuzione dei numeri primi, quindi per simulare quel fruttuoso filone di ricerca matematica, utilizzeremo, ove ve ne fose bisogno, la funzione IsPrime(*n*).

Detto questo, prendete in considerazione le seguenti affermazioni e, aiutandovi con la vostra calcolatrice o con TI-InterActive! e, magari con un foglio elettronico (eventualmente anche quello di TI-InterActive! o Excel, se preferite) dire se sono vere o false, giustificando le risposte:

1. Il polinomio n^2-n+41 genera solo numeri primi.
No perché con 41 genera 41^2
2. Il polinomio $n^2-79n+1601$ genera solo numeri primi.
No perché con 80 genera 41^2

In effetti si può dimostrare il seguente teorema:
Non esistono polinomi $P=P(x)$ non costanti a coefficienti interi tali che $P(x)$ sia primo per ogni numero naturale x
Considerate la seguente argomentazione:
supponiamo per assurdo che la formula $P(x) = a_n x^n + a_{n-1}x^{n-1}+\ldots+a_0$ dia solo numeri primi. Calcoliamo $P(a_0)$; abbiamo $P(a_0)=a_n a_0^{\,n} + a_{n-1}a_0^{\,n-1}+\ldots+a_0=a_0 Q$. Ciò permette di concludere che $P(a_0)$ non è primo, in contraddizione con l'ipotesi fatta.
Si tratta di una dimostrazione? Perché? (suggerimento: un numero naturale *n* non è primo se e solo se $n = ab$ con *a* e *b* naturali diversi da 1)

Sì perché è un'eccezione della dimostrazione per assurdo
Questo dimostra che se prendiamo come variabile il coefficiente, il numero risultante sarà divisibile per il coefficiente
Prof. No: non si tratta di una dimostrazione. Proprio non funziona! Per dire che $P(a_0)$ non è primo, non basta dire che si può scomporre in a_0Q, ma bisognerebbe dimostrare che sia a_0, sia Q sono numeri diversi da 1, altrimenti a_0Q non è una scomposizione di $P(a_0)$! Presentai questa dimostrazione non solo a studenti di una classe, ma anche ad alcuni insegnanti di matematica e nessuno trovò qualcosa da ridire (non c'era malizia in me, solo presi anch'io una cantonata). La dimostrazione che segue, proposta su un teso sacro di teoria dei numeri (Hardy & Wright, pubblicato verso la fine degli anni trenta del secolo scorso) è invece corretta.

La dimostrazione corretta è ben più complicata e qui viene proposta solo per completezza, ma non è necessario studiarla.
Dimostrazione corretta
Notiamo innanzitutto che abbiamo escluso il caso di polinomi costanti, perché, altrimenti, avremmo polinomi come $P(x) = 13$ che, banalmente, generano uno e un solo numero primo per ogni valore di x: evidentemente non si tratta di polinomi che possono dar luogo a formule interessanti per noi!
La seconda osservazione che facciamo è che il coefficiente del monomio di grado massimo deve essere positivo. In caso contrario esisterà un valore di x, diciamo n_0 a partire dal quale il polinomio $P(x)$ assumerà valori negativi. Detto in altri termini, dato $P(x) = a_n x^n + a_{n-1}x^{n-1}+\ldots+a_0$, abbiamo che se $a_n < 0$, allora esiste un numero naturale n_0 tale che $P(x) < 0$ per ogni $x > n_0$ (infatti per x abbastanza grandi il polinomio $P(x)$ assumerà il segno del suo primo termine). In tal caso, quindi, per $x > n_0$ non potremmo più ottenere numeri primi. Supponiamo allora $a_n > 0$. In tal caso esiste un valore di x, diciamo N tale che, per ogni $x > N$, $P(x) > 1$. Sia ora t un numero naturale tale che $t > N$.
Allora abbiamo che $P(t) = a_n t^n + a_{n-1}t^{n-1}+\ldots+a_0 = y > 1$.
Consideriamo ora il termine $P(ry+t) = a_n(ry+t)^n + a_{n-1}(ry+t)^{n-1}+\ldots+a_0$, che è divisibile, per y, qualunque sia l'intero r e, quindi, non è primo. Ciò completa la dimostrazione.

Avendo dimostrato che non esistono formule polinomiali che generano solo numeri primi, la ricerca di tali formule non può che estendersi a funzioni non polinomiali (è possibile dimostrare che anche considerando formule polinomiali a più variabili, non si può trovare una formula che generi solo numeri primi diversa da quella banale $f = k$ con k numero primo).

Il matematico francese Pierre Fermat trovò la seguente formula ritenendo che generasse solo numeri primi: $2^{2^n} + 1$ aveva ragione? Provate a verificare la primalità dei numeri che ottenete per $n = 1, 2, 3, 4, 5$ … che cosa potete concludere?
No, perché semplicemente mettendo n = 5 si ottiene un numero non primo
Il matematico francese Mersenne notò che diversi numeri del tipo $2^p - 1$ con p primo sono primi. Vale per tutti i numeri primi p?
Vale con p = 2, 3, 5, 7 ma con p = 11 restuisce un numero non primo

Il matematico francese Mersenne (1588-1648) notò che diversi numeri del tipo $2^p - 1$ con p primo sono primi.. Per $p = 11$ si ottiene, però, un numero non primo $2047 = 23 \cdot 89$.
Non si sa ancora se la formula di Mersenne consenta di ottenere infiniti numeri primi oppure no.
I matematici Hardy e Wright, nel 1938, scrivono che "il più grande numero primo noto attualmente è

$2^{127}-1 = 170141183460469231731687303715884105727$ “
che è un numero di Mersenne.
Con l’aiuto degli elaboratori automatici sono stati scoperti numeri primi con un numero enorme di cifre: nel 1985 fu scoperto da un computer CRAY-MP/24 un numero primo di Mersenne avente 65050 cifre, per la precisione il numero $2^{216091}-1$. (Per aggiornamenti vedere i siti http://www.liceofoscarini.it/studenti/crittografia/critto/veprimo.htm e, soprattutto, http://primes.utm.edu/).
Attualmente non si sa se esiste un numero finito o infinito di numeri primi di Mersenne.

La ricerca di formule che consentono di generare solo numeri primi, è stata spesso costellata di insuccessi (se considerati rispetto all’obiettivo dichiarato), ma ha portato anche alla determinazione di formule (ovviamente non polinomiali) che generano solo numeri primi, ma la cui complessità di calcolo, in genere, è talmente elevata da renderne impraticabile l’uso per la generazione di numeri primi abbastanza grandi da essere interessanti.

I matematici hanno quindi avviato un altro campo di ricerca, legato alla individuazione di come si distribuiscono i numeri primi. Sappiamo infatti che sono infiniti, ma si vede anche, da un’osservazione sperimentale, che sembrano anche rarefarsi sempre più. Diciamo che sembra che la loro densità diminuisca. In altri termini, fissato un numero n, se calcolate il numero di primi che non seguono n, chiamiamolo $\pi(n)$ vi accorgerete che questo numero ovviamente cresce con n, ma cresce sempre meno. Per la precisione, il rapporto $\frac{\pi(n)}{n}$ sembra diminuire sempre più e addirittura tendere a 0 per n che tende a infinito. Questo per esempio non accade con i numeri pari. Detto $P(n)$ il numero di numeri pari che non seguono n abbiamo che $\lim_{n\to\infty}\frac{P(n)}{n}=\frac{1}{2}$, ossia la densità limite dei numeri pari è ½.
Se $T(n)$ è il numero dei numeri divisibili per 3 che precedono n, qual è la densità limite dei numeri divisibili per 3? E dei numeri divisibili per n? E qual è la densità limite dei numeri che non sono divisibili per 3? E di quelli che non sono divisibili per n? Giustificate la risposta.
Se $T(n)$ è il numero dei numeri divisibili per 3 che precedono n, qual è la densità limite dei numeri divisibili per 3?
1/3
E dei numeri divisibili per n?
1/n
E qual è la densità limite dei numeri che non sono divisibili per 3?
2/3
E di quelli che non sono divisibili per n?
1-1/n

Giustificate la risposta.
Semplicemente perché ogni n numeri ve ne è uno e solo uno divisibile per n.

Si può dimostrare che la sensazione che la densità dei numeri primi tenda a diminuire all’aumentare del numero n considerato è corretta. Siano 2, 3, 5, ..., p tutti i numeri primi non maggiori di p. Allora, per il teorema fondamentale dell’aritmetica (ogni numero naturale è fattorizzabile in modo unico nel prodotto di numeri primi), ogni numero naturale non maggiore di p è divisibile per almeno uno dei numeri primi non maggiori di p. Consideriamo ora il numero $q = 2 \cdot 3 \cdot 5 \cdot .... \cdot p$
Abbiamo che q +2 è divisibile per 2 (è la somma di due numeri divisibili per 2); perciò è composto.
Analogamente, q + 3 è divisibile per 3, q+4 è divisibile per 4, q+p è divisibile per p.
Quindi i numeri q+2, q+3,....q+p non sono numeri primi.

Da quanto ora affermato segue che tra q e $q+p$ esistono almeno p-1 numeri che non sono primi. Possiamo quindi concludere che all'aumentare di p (ossia al crescere del valore dei numeri primi), aumenta anche la distanza fra un numero primo e il suo successivo.
Nella parte che segue cercheremo, insieme, di utilizzare le risorse messe a disposizione dell'elaboratore per esplorare il comportamento asintotico della funzione $\pi(n)$, ossia la funzione che dà, per ogni n, il numero di primi non maggiori di n, in modo da avere informazioni anche sulla densità limite dei numeri primi. Faremo osservazioni aiutati dal software Derive (la calcolatrici hanno risorse di memoria limitate per le esplorazioni che seguono e TI-InterActive! non lo conosco bene da riuscire a fare le cose che riesco a fare con Derive).
Ora vi chiederemo di fare un po' di pratica con Derive. Alcune schede esplicative precederanno esercitazioni guidate, che abbiamo aggiunto per chiarire maggiormente, anche se le schede suggeriscono già le esercitazioni.

Alla scoperta di Derive: il comando Vector

1. Selezionare il comando **Author expression**
Nella riga di inserimento definire la funzione scelta p(n):= $n^2 - n + 41$
e battere <INVIO>
In tal modo si definisce la funzione di nome p(n) che, al variare di n vale $n^2 - n + 41$. Il segno ":=" è un operatore di assegnazione e si scrive digitando, in successione, i segni ":" e "=".

2. Selezionare il comando **Author expression**
Digitare
v := VECTOR(p(n), n, n_{min}, n_{max})
in modo tale da definire un vettore avente per componenti i valori della funzione p(n), di variabile n, per n che va da un valore n_{min} (per esempio 1), a un valore massimo n_{max} che dipende dal numero di valori che si vogliono calcolare per la funzione

3. Selezionare **Author Expression**, digitare
primi_v := VECTOR(IF(PRIME(ELEMENT(v, i)) = true, 1, 0), i, 1, n_{max}-n_{min}+1)
In tal modo si esaminano tutte le componenti del vettore v, ossia i valori calcolati della funzione p:=p(n), chiedendosi se si tratta di numeri primi. Quando la componente in esame del vettore v è un numero primo, DERIVE scrive 1 nel vettore primi_v; se, invece, la componente in esame di v non è un numero primo, in primi_v viene scritto 0.
Ciò è stato fatto utilizzando la funzione predefinita di DERIVE **Prime(n)**, la quale restituisce il valore **true** (vero) se il numero n è primo, mentre restituisce **false** (falso) se n non è primo.
La funzione IF è una caratteristica funzione di scelta; mentre la funzione ELEMENT(v, i) consente di estrarre dal vettore v l'i-esimo elemento.

4.Selezionare il comando **Simplify**
per visualizzare le componenti di primi_v.

Esercitazioni guidate

1. Definizione di una funzione polinomiale per generare numeri primi

Seleziona il comando **Author Expression** e digita nella riga di inserimento la stringa:

p(n):=n^2-n+11
Batti <INVIO>

2. Immissione di alcuni valori della funzione generata in un vettore
Selezionare il comando **Author expression**
Digitare
v := VECTOR(p(n), n, 0, 20)
Batti <INVIO>
Ora utilizza il comando **Simplify** per visualizzare le ventuno componenti del vettore v che corrispondono ai valori p(n) per n che va da 0 a 20.
DERIVE scriverà:
[11, 11, 13, 17, 23, 31, 41, 53, 67, 83, 101, 121, 143, 167, 193, 221, 251, 283, 317, 353, 391]

3. Controllo della primalità di un numero
Selezionare il comando **Author expression**
Digita
PRIME(23)
e batti <INVIO>
Quindi utilizza il comando **Simplify** per calcolare PRIME(23). DERIVE restituisce **true**, ossia valore di verità "VERO" alla proposizione "23 è primo".
Prova con
PRIME(45)
Vedrai che otterrai **false** poiché 45 non è un numero primo.

4. Controllo della primalità dei valori p(n) calcolati
Selezionare **Author Expression**, digitare
primi_v := VECTOR(IF(PRIME(ELEMENT(v, i)) = true, 1, 0), i, 1, 21)
quindi battere <INVIO>
e poi semplificare l'espressione mediante il comando **Simplify**
DERIVE scriverà
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0]
Associando il valore 1 alle componenti di v che sono numeri primi e 0 a quelle che non lo sono.

Un'altra possibilità era quella di digitare la riga
VECTOR(PRIME(ELEMENT(VECTOR(P(n), n, 0, 20), i)), i, 1, 21)
Avresti ottenuto, dopo averne chiesto la semplificazione, il vettore:
[true, true, true, true, true, true, true, true, true, true, true, false, false, true, true, false, true, true, true, true, false]

Alla scoperta di DERIVE: il comando NEXT_PRIME

Digita, in **Author Expression** la stringa
NEXT_PRIME(11)
Che cosa accade?

Digita, in **Author Expression** la stringa
NEXT_PRIME(13)
Che cosa accade?

Digita, in **Author Expression** la stringa
NEXT_PRIME(NEXT_PRIME(NEXT_PRIME(11)))
Che cosa accade?

Che cosa fa la funzione predefinita NEXT_PRIME (n)?

Ricordiamo ora che $\pi = \pi(x)$ è la funzione che dà il numero dei numeri primi non maggiori di x.
Per esempio, $\pi(2) = 1$ (infatti esiste solo un numero primo non maggiore di 2, e cioè 2 stesso)
$\pi(3) = 2$, $\pi(4) = 2$, $\pi(5) = 3$ così via, come suggerisce la seguente tabella

numero naturale x	numeri primi non maggiori di x	$\pi(x)$
1	non ve ne sono	0
2	2	1
3	2,3	2
4	2,3	2
5	2,3,5	3
6	2,3,5	3
7	2,3,5,7	4

Il teorema di Euclide sull'infinità dei numeri primi ci dice che quando x tende a infinito, anche $\pi(x)$ tende a infinito. I matematici hanno dimostrato un risultato molto più significativo, detto **Teorema dei numeri primi** che afferma che al tendere di x all'infinito, la funzione $\pi = \pi(x)$ tende ad assomigliare sempre più alla funzione $f(x) = \frac{x}{\log x}$. Abbiamo quindi un comportamento asintotico di $\pi(x)$, che, tra l'altro, fa capire perché la densità limite dei numeri primi è 0.
Giustificate questa affermazione.
Non so come giustificarla
f(x) cresce con pendenza inferiore a 1 ?

Prof. Si può dire così: abbiamo visto che la densità dei numeri primi è data da $\frac{\pi(x)}{x}$. Se $\pi(x)$ tende ad assomigliare sempre più a $f(x) = \frac{x}{\log x}$, allora la densità dei numeri primi tenderà ad assomigliare sempre di più a $\frac{1}{\log x}$ che, x che tende a infinito, tende a 0

Andrea ha scritto:

Basta calcolare il limite per x che tende a infinito della funzione densità:

$$\Omega(x) = \frac{\pi(x)}{x} = \frac{\left(\frac{x}{\log x}\right)}{x} = \frac{1}{\log x}$$

$$\lim_{x\to\infty} \Omega(x) = \frac{1}{\log x} = 0$$

Aiutandovi con Derive, considerate alcuni termini della successione delle differenze tra ciascun numero primo e quello che lo precede nella successione dei numeri primi. E' corretto affermare che tutti i termini della successione delle differenze sono pari? E' corretto affermare che il minimo di tale successione è 2? Quanti valori uguali a 2 avete incontrato nella tabulazione da voi effettuata di alcuni termini della successione delle differenze? Esistono infiniti valori uguali a 2 nella successione delle differenze? (Attenzione: a questa domanda difficilmente potrete dare una risposta: infatti, anche se congetturano che la risposta sia affermativa, fino ad ora i matematici non sono riusciti a dimostrare che la successione delle differenze assume infinite volte il valore 2).

-I numeri primi (tranne 2) sono dispari
Quindi la differenza tra due numeri dispari è pari [(2n+1)-(2m+1)=2*(n-m)]
-Il minimo è 2, perché (n-m) deve essere al minimo uguale a 1
-Non ho fatto le prove ma elenco alcune coppie: 3,5 ; 5,7 ; 11,13 ; 17,19; ecc
-Questa domanda credo vada in conflitto con il teorema precedente:
se la densità tende a diminuire vorrà dire che la differenza tra un numero primo e il suo precedente tende ad aumentare

Prof. L'ultima affermazione non è corretta: dire che la densità tende a vuol dire che i numeri primi che incontro sono "mediamente" sempre più rari … non vuol certo dire che io non possa incontrare ancora coppie di numeri primi gemelli (ossia che differiscono di 2). Infatti le ricerche sui numeri primi continuano a trovare numeri primi gemelli …

Mediante i due seguenti comandi:
NTH_PRIME(n):=ITERATE(NEXT_PRIME(k),k,1,n)
P(n):=VECTOR([k, NTH_PRIME(k)],k,1,n)
La funzione NTH_PRIME(n) restituisce, per ogni *n*, l'ennesimo numero primo. Il comando ITERARATE itera l'operazione NEXT_PRIME da 1 a n e restituisce solo l'ultimo valore calcolato. Per esempio,
NTH_PRIME(5)
Per k = 1 dà 2 (il primo numero primo che viene dopo 1)
Per k = 2 dà 3 (il primo numero primo che viene dopo 2)
Per k = 3 dà 5 (il primo numero primo che viene dopo 3)
Per k = 4 dà 7 (il primo numero primo che viene dopo 5)
Per k = 5 dà 11 (il primo numero primo che viene dopo 7)
Quindi NTH_PRIME(5) restituisce il quinto numero primo.

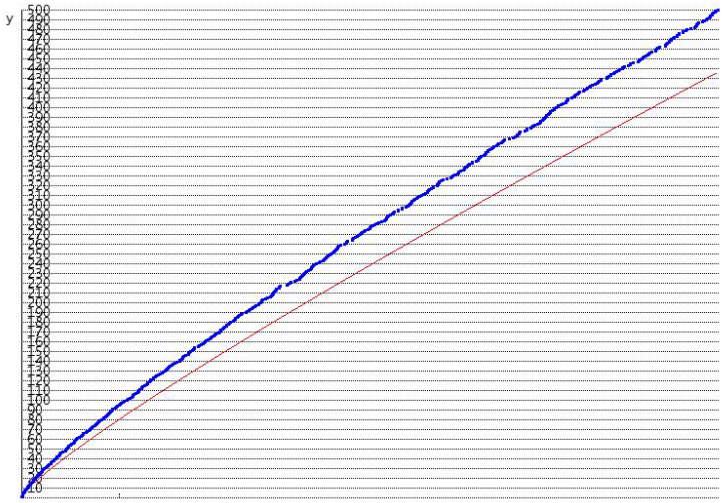
Invece la funzione P(*n*) genera le coppie del tipo (*n*,NTH_PRIME(*n*)), ossia coppie ordinate il cui primo elemento è l'ennesimo numero naturale e il secondo elemento l'ennesimo numero primo.
Quindi tabulando P(n) otteniamo una cosa del tipo:

n	P(n)
1	2
2	3
3	5
4	7
5	11

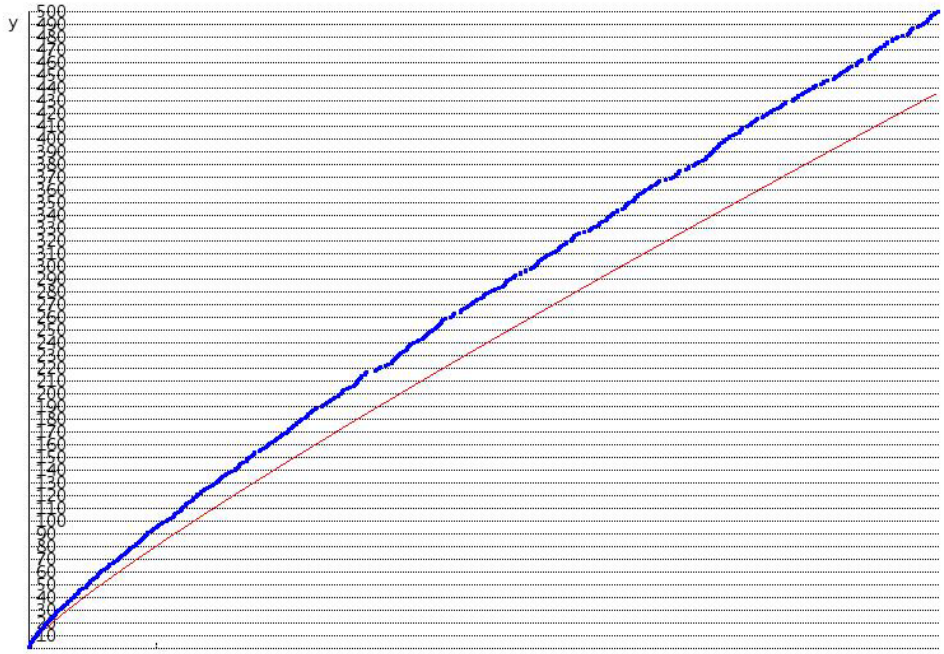
Ora che cosa accade se consideriamo la funzione
PIGRCO(n) :=VECTOR([NTH_PRIME(k),k], k,1,n)
che inverte le componenti delle coppie della funzione P(n)? Otteniamo proprio la funzione che a ogni *n* associa il numero di numeri primi non maggiori di *n*.

N	PIGRECO(n)
2	1
3	2
5	3
7	4
11	5

Tabulate la funzione PIGRECO, ossia π, per i primi 500 e poi 1000 valori (basta digitare PIGRECO(500) e poi chiedere SIMPLIFY base e analogamente con PIGRECO(1000)); quindi e tracciate il grafico dei primi 500 e poi dei primi 1000 valori della funzione PIGRECO (basta selezionare la matrice di punti e cliccare sull'icona che invia alla finestra grafica e poi, in quata finestra, nuovamente sulla stessa icona, ovviamente scegliendo un'adeguata finestra grafica). Confrontate quindi ciascuno dei due grafici con quello della funzione $f(x)=\frac{x}{\log x}$. Che cosa osservate?
Andrea propone il grafico:

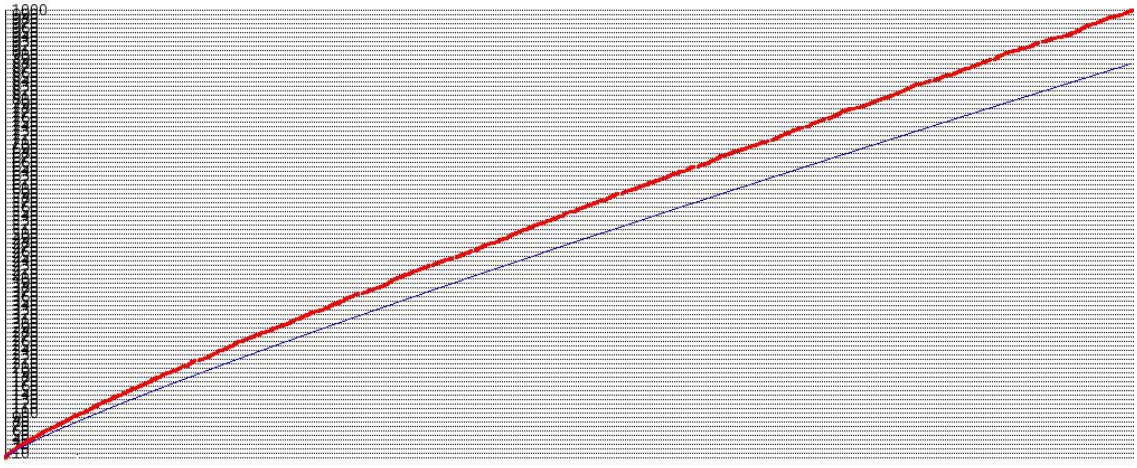


Confrontiamo il plot dei primi 500 primi con quello della funzione $f(x)=\frac{x}{\log x}$:



Come si può notare la funzione non approssima molto bene l'andamento della successione.

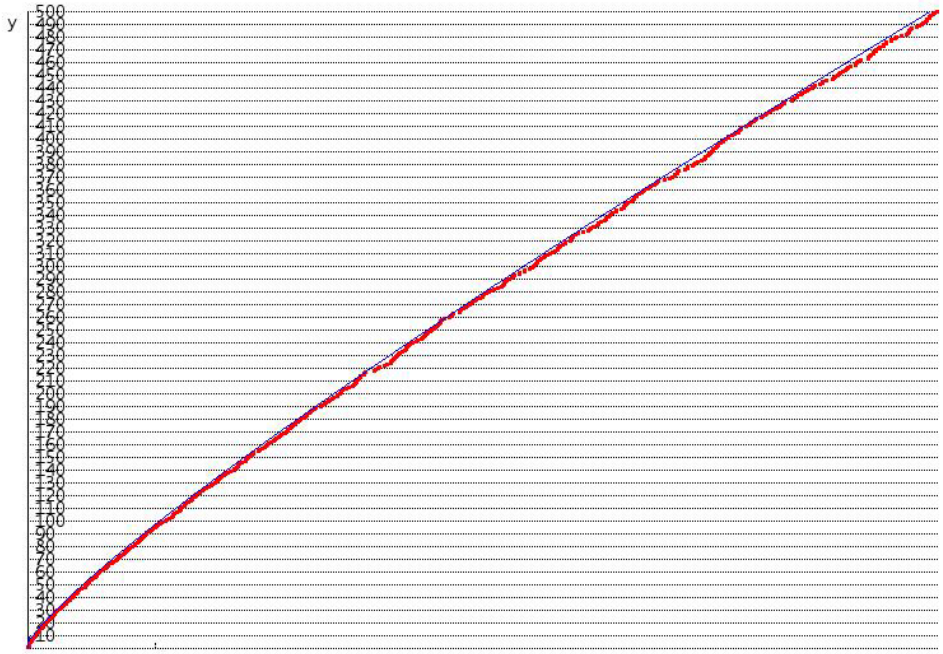
Confrontiamo la stessa funzione ma con i primi 1000 numeri primi questa volta:



Anche questa volta l'andamento della successione non è eccellentemente approssimato dalla funzione.

Provate ora a fare tracciare il grafico di $f(x)=\frac{x}{\log x - 1{,}08366}$. Che cosa accade?

Confrontiamo questa volta la successione dei primi 500 numeri primi con la funzione $f(x)=\frac{x}{\log x - 1{,}08366}$:



Ripetiamo il confronto con la stessa funzione ed i primi 1000 numeri primi:

Come si può facilmente vedere, ora la funzione approssima decisamente meglio l'andamento della successione dei numeri primi.

Per approfondire le problematiche connesse a queste ricerche, suggeriamo la lettura delle pagine riportate al seguente indirizzo web (si tratta di letture senza dubbio utili, anche se non tutto quanto viene scritto può essere compreso da studenti liceali) :
http://matmedia.ing.unina.it/Concorsi%20a%20cattedra/quesiti%2011%20gennaio%202000/Soluzioni%20gruppo2/Distribuzione%20dei%20primi/la_distribuzione_dei_primi.htm
e con
http://www.geocities.com/Heartland/Plains/4142/prime_numbers.html

Uso di Derive per esplorare insiemi modulo n

Propongo alcune attività con l'uso di Derive per esplorare gli insiemi modulo n e per iniziare a effettuare con essi qualche attività preparatoria alle lezioni che seguirete con il prof. Impedovo.
Riporto, insieme agli esercizi proposti, anche una loro possibile soluzione. Suggerisco, però, di guardarla solo dopo aver provato voi stessi a rispondere.

1. Scrivere un programma in Derive che consenta di ottenere le tavole di addizione e di moltiplicazione di $\mathbf{Z}_4$ e di $\mathbf{Z}_5$.

2. Basandosi sul piccolo teorema di Fermat (se p è primo, allora $n^p \equiv n \pmod{p}$ per ogni intero n), scrivere un programma in Derive che consenta di determinare il vettore avente per componenti gli inversi di tutti gli elementi invertibili di $\mathbf{Z}_5$ e di $\mathbf{Z}_7$.

3. Generalizzare il programma precedente scrivendo un vettore che abbia per componenti l'inverso di ogni elemento di $\mathbf{Z}_m$ con m primo.

4. Predisporre una serie di esercizi che possano guidare uno studente di biennio di S.S.S. a congetturare proprietà valide nelle strutture $\{ \mathbf{Z}_m, +, \cdot \}$.

Risposte ai primi quattro esercizi proposti nella scheda 6

Esercizio 1.
TAV42(i,j):=VECTOR(VECTOR(MOD(i*j,,4),i,0,3),j,0,3)
TAV42(i,j):=VECTOR(VECTOR(MOD(i*j,,4),i,0,3),j,0,3)
TAV51(i,j):=VECTOR(VECTOR(MOD(i+j,,5),i,0,4),j,0,4)
TAV52(i,j):=VECTOR(VECTOR(MOD(i*j,,5),i,0,4),j,0,4)

Esercizio 2.
INVERSO:=VECTOR(MOD(i^{5-2}, 5), i,1, 5-1)
INVERSO:=VECTOR(MOD(i^{7-2}, 7), i,1, 7-1)

Esercizio 3.
INVERSO(n):=VECTOR(MOD(i^{n-2}, n), i,1, n-1)

Esercizio 4.
Hanno tutte struttura di anello; un elemento n ammette inverso rispetto alla moltiplicazione se e solo se è primo con m. Se m è primo, allora tutti gli elementi non nulli ammettono inverso rispetto alla moltiplicazione. In tal caso $\mathbf{Z}_m$ è un corpo. Un'equazione del tipo $ax = b$ ammette una e una sola soluzione se e solo se a ammette inverso; altrimenti non è garantita l'esistenza né l'unicità.