PARI/GP

Introduzione

PARI/GP

- PARI/GP è la combinazione di due ambienti:
  - PARI – libreria di funzioni (scritte in C) orientate alla teoria dei numeri
  - GP – interprete che fornisce una interfaccia a linea di comando alle funzioni di PARI che permette di fruire di un ambiente di sviluppo per applicazioni numeriche.
- Le funzioni di PARI/GP sono raccolte in forma di libreria di funzioni C e possono essere richiamate da un qualunque programma in linguaggio C.
- Scaricabile dalla rete http://pari.math.u-bordeaux.fr/
- Software libero con licenza GPL (General Public License)

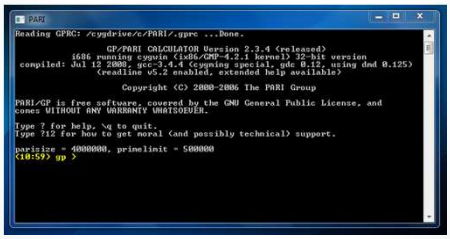
Filosofia di PARI

- L'idea alla base del linguaggio è quella di fornire per le operazioni un risultato il più corretto possibile.
- Per esempio utilizza le frazioni come risultato delle divisioni intere:
  - 1 / 3 fornisce come risultato "un terzo"
  - 1. / 3 fornisce 0.33333333333333333333

Perché utilizziamo PARI

- Fornisce funzioni potenti dedicate alla teoria dei numeri
  - Potremmo scriverle utilizzando le funzioni base (ma sarebbe oneroso in termini di tempo di programmazione)
- Dà la possibilità di operare con aritmetica intera con valori di lunghezza arbitraria
  - Numeri «molto grandi»

Interfaccia «povera»



Help on line

- ? per ottenere il manuale in linea
- tutorial.pdf presente nei file di installazione
- user.pdf manuale più approfondito

Operatori

- +, -, *, / con ovvio significato
- / divisione (la filosofia del linguaggio fa sì che il calcolo sia il più possibile preciso: 4/3 fornisce come risultato quattro terzi)
- \ (divisione Euclidea) (divisione intera)
- % (resto Euclideo) (resto della divisione intera)
- ^ esponenziale
- operatori logici
  - ! not
  - && and
  - || or
  - i valori logici sono intesi come 1=true e 0=false

«macchina calcolatrice»

- E' possibile utilizzare l'ambiente come «macchina calcolatrice» evoluta
- Esempio:
  - 3+5
  - 5^4
  - 4/3
  - 4\3
  - 5!
- Permette di operare con numeri molto grandi
  - 100!
- E' possibile assegnare valori a variabili
  - x = 2^10
  - y=x^x
- I valori booleani sono rappresentati da 0(false) e 1(true)

Strutture di controllo

- In PARI le strutture di controllo sono in realtà funzioni
- Esempio if
  - if(<condizione>,<istr.vero>,<istr.falso>)
  - if (voto>=6,print("promosso"),print("bocciato"))
  - massimo(a,b)={ if (a>b,a,b);}
- Esempio while
  - while(<condizione>,<corpo ciclo>)
  - while(x<10,print(x);x++);
  - numeriDaA(vmin,vmax)={
    local n;
    n=vmin;
    while(n<=vmax,
    print(n);
    n++);}

for

- Sintassi ciclo for (ripetizione di una sequenza di istruzioni)
- for(<var>=<inizio>,<fine>,<istruzioni>)
- for(i=10,20,print(i*2));
- for(i=10,20,print(i*2);print(i));

File di funzioni

- Le funzioni possono essere memorizzate in file di testo (con estensione .gp)
- I file possono essere letti con il comando read(nomefile)
- Suggerimento memorizzare i file nella sottocartella examples

Esempi di funzioni

```
dieci()={
    local n;
    for(n=1,10,print(n));
}

valuta(voto)={
   if (voto>=6,print("promosso"),print("bocciato"));
}

numeriDaA(vmin,vmax)={
    local n;
    n=vmin;
    while(n<=vmax,
       print(n);
       n++      );
}
```

Esempi

```
dispari(x)={
  if((x%2)==0,0,1);
}

divisori(n)={
  local d;
  fordiv(n,d,print(d));va
}

fatt(n)={
  if(n==0,1,n*fatt(n-1));
}
```

Funzioni utili …

- … per le nostre applicazioni di crittografia
- isprime(n)
  - 1 se n è primo, 0 se non primo
- factor(n)
  - restituisce la fattorizzazione di n

Alcuni esempi

```
/* m modulo n */
modulo(m,n)=lift(Mod(m,n))

/* prodotto in Zn */
prodottoZn(x,y,n)=modulo(x*y,n)

/* tavola della moltiplicazione Zn */
moltiplicazioneZn(n)=
{
  local(mat,i,j);
  mat = matrix(n,n);
  for(i=1,n,
    for(j=1,n,mat[i,j]=prodottoZn(i-1,j-1,n));
          );
  print("tavola moltiplicazione ",n);
  return(mat);
}
```

… esempi

```
/* rappresentazione (max 8 cifre) binaria "a rovescio" */
rbinr(n)=
{
  local(c);
  for(c=0,7,print(bittest(n,c)));
}

/* restituisce rappresentazione (max 8 cifre) binaria  */
rbin(n)=
{
  local(c);
  local(ris);
  ris="";
  for(c=0,7,ris=concat(bittest(n,c),ris));
  return(ris);
}
```

Conversione lettere numeri

```
/* restituisce un numero compreso fra 1 e 26 che rappresenta la posizione della
lettera nell'alfabeto */
lToN(lett)=
{
local(numeri);      /* array utilizzato per la conversione */
local(ris); /* risultato numerico della conversione */
num=Vecsmall(lett); /* array con rappresentazione numerica della stringa */
ris = num[1];       /* primo elemento dell'array (indici partono da 1) */
ris = ris - 64;     /* rientra nell'intervallo 1::26 */
}

/* dato un numero compreso fra 1 e 26 restituisce la corrispondente lettera
dell'alfabeto */
nTol(num)=Strchr(num+64);
```

Videolezioni

- http://www.youtube.com/watch?feature=player_embedded&v=0G-9JzlrzBM