

### Tabella di funzioni matematiche

Radice quadrata	$\sqrt{x}$	<b>SQRT (x)</b>
Quadrato	$x^2$	<b>SQR (x)</b>
Elevamento a potenza	$x^y$	EXP (LN (x) * y)
Radice ennesima	$\sqrt[y]{x}$	EXP (LN (x) / y)
Seno	<b>SIN</b>	<b>SIN (x)</b>
Coseno	<b>COS</b>	<b>COS (x)</b>
Tangente	<b>TAN</b>	SIN (x) / COS (x)
Arcoseno	<b>ARCSIN</b>	ARCTAN (x / SQRT (1 - SQR (x) ) )
Arcocoseno	<b>ARCCOS</b>	ARCTAN (SQRT (1 - SQR (x) ) / x)
Arcotangente	<b>ARCTAN</b>	<b>ARCTAN (x)</b>
Logaritmo naturale	<b>LN</b>	<b>LN (x)</b>
Logaritmo in base 10	<b>LOG</b>	LN (x) / LN (10)
Logaritmo in base 2	<b>LOG2</b>	LN (x) / LN (2)
Antilogaritmo naturale	$e^x$	<b>EXP (x)</b>
Antilogaritmo in base 10	$10^x$	EXP (LN (10) * x)
Antilogaritmo in base 2	$2^x$	EXP (LN (2) * x)
PI greco	$\pi$	<b>PI</b>

## SOLUZIONI ESERCIZI PROPOSTI

### Esercizi sullo schema sequenziale

*Scrivi un programma che compia le seguenti azioni*

1. Legge in INPUT il raggio di un cerchio e restituisce in OUTPUT l'area e la circonferenza.

```
PROGRAM areacirconferenza;
```

```
VAR rag,ar,cir: REAL;
```

```
BEGIN
```

```
ar:=0; cir:=0;
```

```
  READLN(rag);
```

```
ar:=pi*sqr(rag);
```

```
cir:=2*pi*rag;
```

```
  WRITELN ('RAGGIO=', rag:6:0,' CIRCONFERENZA=',cir:6:0, '
AREA=',ar:6:0);
```

```
END.
```

2. Legge in INPUT le basi e l'altezza di un trapezio e restituisce in OUTPUT l'area.

```
PROGRAM areatrapezio;
```

```
VAR Bmag,Bmin,h,ar: REAL;
```

```
BEGIN
```

```
ar:=0;
```

```
  READLN(Bmag,Bmin,h);
```

```
ar:=((Bmag+Bmin)*h)/2;
```

```
WRITELN (' AREA=',ar:6:0);
```

```
END.
```

3. Legge in INPUT i cateti di un triangolo rettangolo e restituisce in OUTPUT l'ipotenusa.

```
PROGRAM IPOTENUSA;
```

```
VAR a,b,c: REAL;
```

```
BEGIN
```

```
c:=0;
```

```
  READLN(a,b);
```

```
c:=sqr(sqr(a)+sqr(b));
```

```
WRITELN (' IPOTENUSA=',C:6:3);
```

```
END.
```

4. Legge in INPUT i coefficienti di un'equazione di primo grado e restituisce in OUTPUT la soluzione (se esiste).

### Esercizi sullo schema di selezione if

```

PROGRAM equazioneprimogrado;
VAR a,b,s:REAL;
BEGIN
(*inserimento coefficienti equazione ax+b=0*);
READLN(a,b);
WRITELN('equazione ',a:3:3,'x+',b:3:3,'=0');
IF a<>0 THEN
    BEGIN
        s:=-b/a;
        WRITELN('la soluzione della equazione è ',s:10:3);
    END;
IF a=0 THEN
    BEGIN
        IF b=0 THEN
            WRITELN('equazione indeterminata')
        ELSE
            WRITELN('equazione impossibile');
        END;
    END;
END.

```

5. Legge in INPUT i coefficienti di una disequazione di primo grado e restituisce in OUTPUT la soluzione (se esiste).

```

PROGRAM disequazioneprimogrado;
VAR a,b,s:REAL;
BEGIN
(*inserimento coefficienti disequazione ax+b>0*);
READLN(a,b);
WRITELN('equazione ',a:3:3,'x+',b:3:3,'=0');
IF a>0 THEN
    BEGIN
        s:=-b/a;
        WRITELN('la soluzione della disequazione è x> ',s:10:3);
    END;
IF a<0 THEN
    BEGIN
        s:=-b/a;
        WRITELN('la soluzione della disequazione è x< ',s:10:3);
    END;
IF a=0 THEN

```

```

        BEGIN
            IF b>0 THEN
                WRITELN('sempre verificata')
            ELSE
                WRITELN('NESSUNA SOLUZIONE')
            END;
        END.

```

6. Legge in INPUT due numeri naturali  $a$ ,  $b$  e restituisce in OUTPUT un terzo numero naturale  $x$ , che sommato al più piccolo riproduce quello più grande.

```

PROGRAM terzonumero;
    VAR a,b,x:INTEGER;
    BEGIN
        READLN(a,b);
        x:=0;
        IF a<b THEN
            begin
                x:=b-a;
                WRITELN('a<b e x=',x);
            end
        ELSE
            begin
                x:=a-b;
                WRITELN('a=>b e x= ',x);
            end
        end
    END.

```

7. Legge in INPUT il valore di un angolo di un triangolo e restituisce in OUTPUT il tipo (ottuso, acuto, retto).

```

PROGRAM angolo;
    VAR a:INTEGER;
    BEGIN
        READLN(a);
        IF a>180 THEN
            begin
                WRITELN('angolo di ',a,' gradi eccedente 180, riprova!');
            end;
        IF a<=180 THEN
            BEGIN

```

```

        IF a=90 THEN
            begin
                WRITELN('angolo di ',a,' gradi è retto');
            end;
IF a<90 THEN
begin
WRITELN('angolo di ',a,' gradi è acuto');
end;
        IF a>90 THEN
            begin
                WRITELN('angolo di ',a,' gradi è ottuso');
            end;
        END;
END.

```

8. Legge in INPUT i lati di un triangolo e restituisce in OUTPUT la risposta alla domanda se è rettangolo oppure no.

```

program triangolorettangolo;
var a,b,c,s:real;
begin
READLN(a,b,c);
s:=0;
IF a>b THEN
    BEGIN
        IF a>c THEN
            BEGIN
                writeln(a:3:2,' ipotenusa ',b:3:2,' cateto ',c:3:2, ' cateto');
                s:=sqr(a)-sqr(b)-sqr(c)
            END
        ELSE
            BEGIN
                writeln(c:3:2,' ipotenusa ',b:3:2,' cateto ',a:3:2, ' cateto');
                s:=sqr(c)-sqr(a)-sqr(b)
            END
        END
    ELSE
        BEGIN
            IF b>c THEN
                BEGIN
                    writeln(b:3:2,' ipotenusa ',c:3:2,' cateto ',a:3:2, ' cateto');
                END
            END
        END
    END

```

```

                s:=sqr(b)-sqr(c)-sqr(a)
            END
        ELSE
            BEGIN
                writeln(c:3:2,' ipotenusa ',b:3:2,' cateto ',a:3:2, ' cateto');
                s:=sqr(c)-sqr(a)-sqr(b)
            END
        END;
    if s=0 then
        writeln('TRIANGOLO RETTANGOLO')
    ELSE
        writeln('Triangolo non rettangolo')
    end.

```

9. Legge in INPUT le misure di tre segmenti e restituisce in OUTPUT la risposta alla domanda se si può costruire con essi un triangolo oppure no.

```

program tresegmenti;
var a,b,c:REAL;
begin
    READLN(a,b,c);
    IF a+b<c then
        begin
            writeln(a:3:2,' ',b:3:2,' ',c:3:2,' non formano un triangolo')
        end
    ELSE
        begin
            if sqr(b-a)<c then
                begin
                    writeln(a:3:2,' ',b:3:2,' ',c:3:2,' formano un triangolo')
                end
            else
                begin
                    writeln(a:3:2,' ',b:3:2,' ',c:3:2,' non formano un
triangolo')
                end;
            end;
        end;
end;

```

end.

10. Legge in INPUT un numero intero e restituisce in OUTPUT il suo valore assoluto.

```

program valoreassoluto;
var n:REAL;
begin
  READLN(n);
  if n >= 0 then
    writeln('il valore assoluto di ',n:3:2,' è ',n:3:2)
  else
    writeln('il valore assoluto di ',n:3:2,' è ',-n:3:2)
end.

```

end.

11. Legge in INPUT i coefficienti  $a$ ,  $b$ ,  $c$ , di un'equazione di secondo grado  $ax^2 + bx + c = 0$  e restituisce in OUTPUT le soluzioni se esistono, oppure un messaggio d'errore).

```

program equazione2grado;
var a,b,c,x1,x2:real;
begin
  (*inserimento coefficienti nell'ordine*);
  readln(a,b,c);
  writeln('a= ',a:2:2);
  writeln('b= ',b:2:2);
  writeln('c= ',c:2:2);
  x1:=0; x2:=0;
  if a=0 then
    begin
      writeln('equazione di primo grado');
      if b<>0 then
        begin
          writeln('soluzione x= ',(c/b):2:2)
        end
      else
        if c=0 then
          begin
            writeln('equazione indeterminata')
          end
        else
          begin

```

```

                                writeln('equazione impossibile')
                            end
                        end
else
    begin
        if (sqr(b)-4*a*c)<0 then
            begin
                writeln('nessuna soluzione')
            end
        else
            if (sqr(b)-4*a*c)=0 then
                begin
                    x1:=-b/(2*a);
                    writeln('due soluzioni coincidenti x1=x2= ',x1:2:2);
                end
            else
                begin
                    x1:=(-b-sqrt((sqr(b)-4*a*c)))/(2*a);
                    x2:=(-b+sqrt((sqr(b)-4*a*c)))/(2*a);
                    writeln('due soluzioni x1= ',x1:2:2,'e x2= ',x2:2:2);
                end
            end;
        end.

```

12. Legge in INPUT i coefficienti  $a$ ,  $b$ ,  $c$ , di una disequazione di secondo grado  $ax^2 + bx + c > 0$  e restituisce in OUTPUT le soluzioni se esistono, oppure un messaggio d'errore.

```

program disequazione2grado;
var a,b,c,x1,x2:real;
begin
(*inserimento coefficienti nell'ordine*);
readln(a,b,c);
writeln('a= ',a:2:2);
writeln('b= ',b:2:2);
writeln('c= ',c:2:2);
x1:=0; x2:=0;
if a=0 then
    begin
        writeln('disequazione di primo grado');

```



```

if b<>0 then
    begin
        if b>0 then
            writeln('soluzione x> ',(-c/b):2:2)
        else
            writeln('soluzione x< ',(-c/b):2:2)
        end
    else
        if c=0 then
            begin
                writeln('disequazione impossibile ')
            end
        else
            begin
                if c>0 then
                    writeln('disequazione sempre vera')
                else
                    writeln('disequazione impossibile ')
                end
            end
        end
    (*da qui inizia il caso a>0*)
else
    begin
        if (sqr(b)-4*a*c)<0 then
            begin
                if a>0 then
                    writeln('disequazione sempre vera')
                else
                    writeln('disequazione mai verificata')
                end
            end
        else
            if (sqr(b)-4*a*c)=0 then
                begin
                    x1:=-b/(2*a);
                    if a>0 then
                        writeln('x< ',x1:2:2,' v x> ',x1:2:2 )
                    else

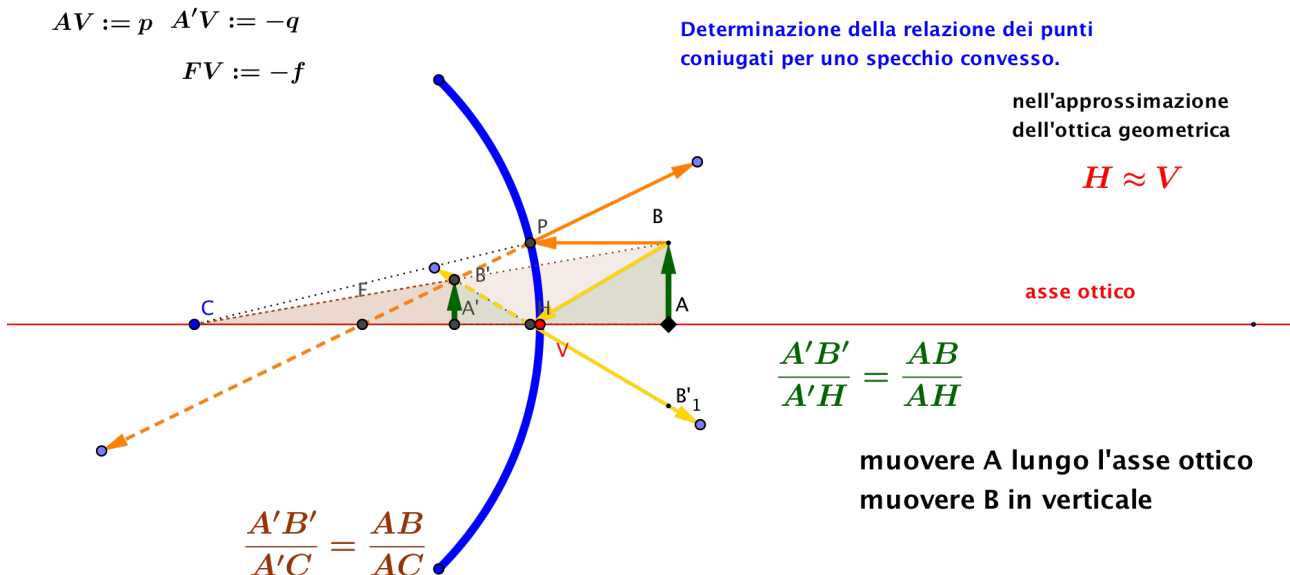
```

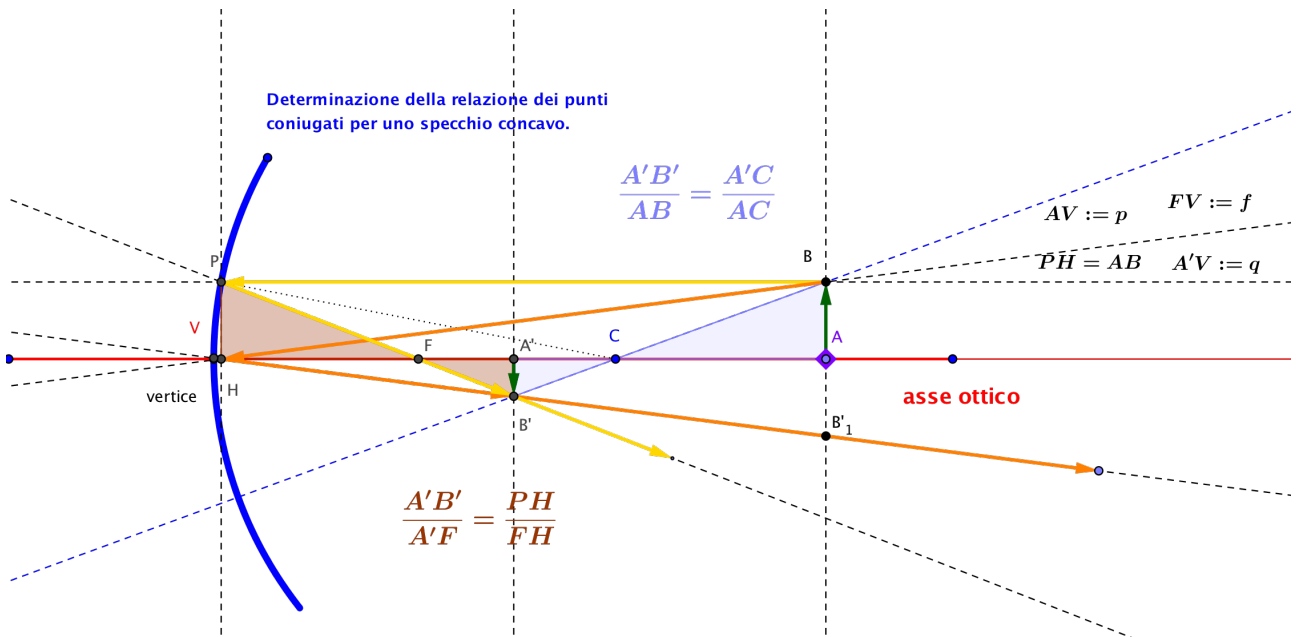
```

        writeln('disequazione mai verificata')
    end
else
    begin
        x1:=(-b-sqrt((sqr(b)-4*a*c))/(2*a);
        x2:=(-b+sqrt((sqr(b)-4*a*c))/(2*a);
        if a>0 then
            writeln('x< ',x1:2:2,' v x> ',x2:2:2)
        else
            writeln(x2:2:2,'< x < ',x1:2:2)
        end
    end
end;
end.

```

13. Come l'esercizio 11 per il caso  $ax^2 + bx + c \geq 0$ .
14. Come l'esercizio 11 per il caso  $ax^2 + bx + c < 0$ .
15. Come l'esercizio 11 per il caso  $ax^2 + bx + c \leq 0$ .
16. Ed ora il gran finale: oltre ai coefficienti, il programma legge in INPUT anche il segno della disequazione  $>$ ,  $\geq$ ,  $<$ ,  $\leq$  e restituisce in output la soluzione della disequazione.





17. Legge in INPUT il tipo di specchio (*concavo* o *convesso*) la posizione della sorgente  $p$ , la distanza (in valore assoluto) a cui si forma l'immagine  $q$  e restituisce in OUTPUT il raggio di curvatura dello specchio  $r$ , la distanza focale  $f$  e l'ingrandimento  $I$ . Il programma deve mandare un messaggio che ci informa sul tipo di immagine che si forma (*reale* o *virtuale*). (Ricordiamo che per gli specchi sferici di piccola apertura vale la legge  $\frac{1}{p} + \frac{1}{q} = \frac{1}{f}$ , che  $r = 2f$  e che  $I = \frac{q}{p}$ ).

### Esercizi sullo schema for

Scrivi un programma che compia le seguenti azioni

18. Legge in INPUT i voti riportati in  $n$  prove di verifica e restituisce in OUTPUT la media aritmetica.

```

program mediaaritmetica;
var s,v,m:real;
n,c:integer;
begin
  readln(n);
  m:=0;c:=0;
  for c:=1 to n do
    begin
      read(v);
      s:=s+v;
    end;
  m:=s/n;
  writeln('LA MEDIA SU ',n,' VOTI: ',m:2:2);
end.

```

19. Legge in INPUT i voti riportati in  $n$  prove di verifica e restituisce in OUTPUT un messaggio “bene” se la media aritmetica è maggiore o uguale a 6, “male” se è minore di 6.

```

program mediaaritmeticabis;
var s,v,m:real;
n,c:integer;
begin
  readln(n);
  m:=0;c:=0;
  for c:=1 to n do
    begin
      read(v);
      s:=s+v;
    end;
  m:=s/n;
  writeln('LA MEDIA SU ',n,' VOTI: ',m:2:2);
  if m>=6 then
    writeln('sufficiente')
  else
    writeln('insufficiente');
  end.

```

20. Restituisce in OUTPUT (stampa) e calcola la somma dei primi  $n$  numeri pari.

```

program sommapari;
var
n,c,s:integer;
begin
  readln(n);
  c:=0; s:=0;
  for c:=1 to n do
    begin
      s:=s+2*c;
    end;
  writeln('La somma dei primi ', n,' numeri pari è ', s);
  end.

```

21. Restituisce in OUTPUT (stampa) e calcola la somma dei primi  $n$  numeri dispari.

```

program sommadispari;
var
n,c,s:integer;
begin
  readln(n);
  c:=0; s:=0;
  for c:=1 to 2*n do
    begin
      if c mod 2 = 1 then
        begin
          write(c, ' ');
          s:=s+c;
        end;
    end;
  writeln(' ');
  writeln('La somma dei primi ', n, ' numeri dispari è ', s);
end.

```

22. Scrive la successione di numeri 1,-2,3,-4,5,-6,...,-100.

```

program sommalternata;
var
c:integer;
begin
  for c:=1 to 100 do
    begin
      if c mod 2 = 1 then
        write(c, ' ')
      else
        write(-c, ' ');
    end;
  end.

```

23. Scrive la successione di numeri  $1/2, 2/3, 3/4, \dots, 99/100$ .

```

program sommalternata;
var
c:integer;

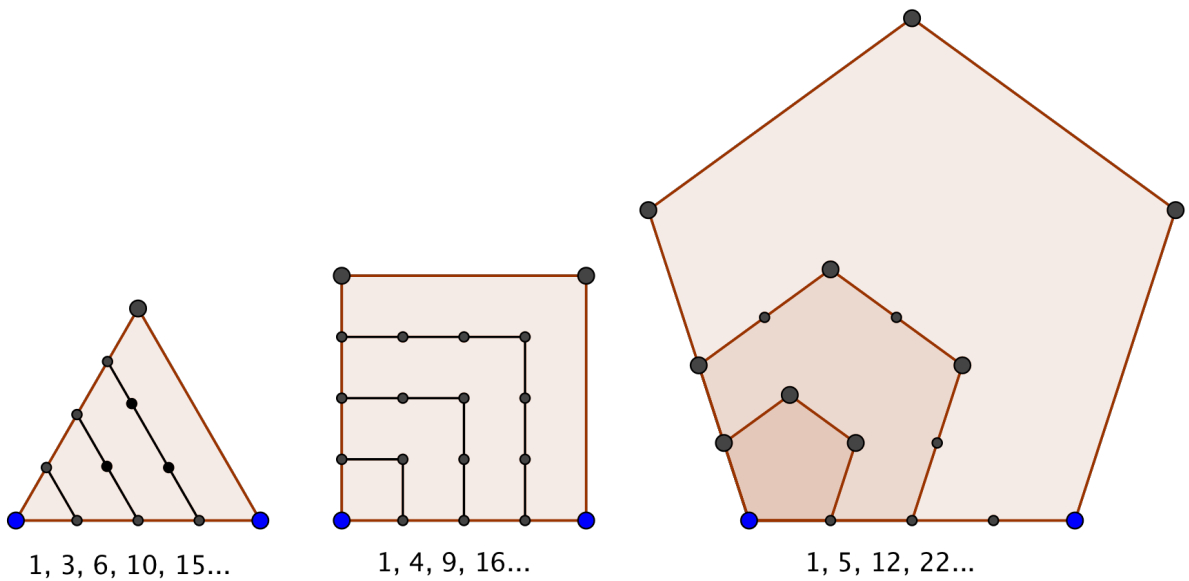
```

```

begin
  for c:=1 to 100 do
    begin
      write(c,'/',(c+1),' ');
    end;
end.

```

24. Scrive i primi 10 numeri poligonali nei tre casi seguenti (triangolari, quadrati, pentagonali).



```

program NUMERIPOLIGONALI;
(* questo programma è generale: chiede in input il numero di lati del
poligono e quanti numeri vogliamo stampare, e la somma di questi*);
VAR d,n,s,c,m,somma:INTEGER;
BEGIN
  (*inserisci il numero di lati del poligono*);
  read(n);
  (*inserisci la quantità di numeri da stampare*)
  read(m);
writeln(' ');
s:=0; d:=0;somma:=0;
  FOR c:=1 to m do
    begin

```

```

        d:=3-n+(n-2)*c;
        s:=s+d;
        somma:=somma+s;
        write(s,' ');
    end;
writeln(' ');
writeln('la somma dei primi ',m,' numeri poligonali è: ',somma);
END.

```

26. Legge in INPUT un intero  $n$  e restituisce in OUTPUT i primi  $n$  numeri di Fibonacci.

```

program fibonacci;
VAR n,a,b,c,s:INTEGER;
begin
    READLN(n,a,b);

    write(a,' ',b,' ');
    s:=0;
    for c:=3 to n do
        begin
            s:=a+b;
            a:=b;
            b:=s;
            write(s,' ');
        end;

    end.

```

27. Legge in INPUT un intero  $n$  e restituisce in OUTPUT la somma dei primi  $n$  numeri di Fibonacci.

```

program fibonacci2;
VAR n,a,b,c,s,somma:INTEGER;
begin
    READLN(n,a,b);

    write(a,' ',b,' ');
    s:=0;
    somma:=a+b;
    for c:=3 to n do
        begin

```

```

        s:=a+b;
        a:=b;
        b:=s;
        write(s,' ');
        somma:=somma+s;
    end;
    writeln(' ');
writeln('la somma dei primi ',n,' termini della succ.di Fibonacci è:
',somma);
end.

```

28. Legge in INPUT un intero  $n$  e restituisce in OUTPUT la frazione con al numeratore un termine della successione di Fibonacci e al denominatore il termine precedente.

```

program fibonacci3;
VAR n,a,b,c,s:INTEGER;
begin
    READLN(n,a,b);

    write(a,' ',b,' ');
    s:=0;

    for c:=3 to n do
        begin
            s:=a+b;
            a:=b;
            b:=s;
            write(b,'/',a,' ');
            somma:=somma+s;
        end;
    end.

```

Adesso vogliamo indagare il rapporto tra un termine della sequenza di Fibonacci e il precedente.

```

program fibonacci4;
VAR n,a,b,c,s:INTEGER;
VAR r:REAL;
begin
    READLN(n,a,b);

```



```

s:=0;
r:=0;
for c:=3 to n do
  begin
    s:=a+b;
    a:=b;
    b:=s;
    r:=b/a;
    writeln(r:1:3);
  end;
end.

```

29. Legge in INPUT un intero  $n$  e restituisce in OUTPUT il suo fattoriale.

```

program fattoriale;
var n,i,f:integer;

BEGIN
  f:=1;
  readln(n);
  for i:=1 to n do
    begin
      f:=f*i;
    end;
  writeln('Il fattoriale di ',n,' è: ',f);
END.

```

### **Esercizio**

30. Simula il lancio di un dado  $n$  volte.

```

program DADO;
VAR n,uno,due,tre,quattro,cinque,sei,c,lancio:INTEGER;
begin
uno:=0;tre:=0;Quattro:=0;
due:=0;cinque:=0;sei:=0;
READLN(n);
randomize;
  for c:=1 to n do
    begin
      lancio:=0;
      lancio:=random(6);
    end;
  end;
end.

```

```
        if lancio = 0 then
            uno:=uno+1;
        if lancio = 1 then
            due:=due+1;
        if lancio = 2 then
            tre:=tre+1;
        if lancio = 3 then
            Quattro:=Quattro+1;
        If lancio = 4 then
            Cinque:=cinque+1;
        If lancio =5 then
            Sei:=sei+1;

write(lancio+1,' ');
        end;
writeln(' ');
writeln('Lanci: ',n);
write(' 1: ',uno );
c:=0; for c:=1 to uno do
begin
write('*');
end;
writeln(' ');
write(' 2: ',due);
c:=0; for c:=1 to due do
begin
write('*');
end;
writeln(' ');
write(' 3: ',tre);
c:=0; for c:=1 to tre do
begin
write('*');
end;
writeln(' ');
write(' 4: ',quattro);
c:=0; for c:=1 to quattro do
begin
write('*');
end;
```

```
writeln(' ');
write(' 5: ',cinque);
c:=0; for c:=1 to cinque do
begin
write('*');
end;
writeln(' ');
write(' 6: ',sei);
c:=0; for c:=1 to sei do
begin
write('*');
end;
writeln(' ');
end.
```