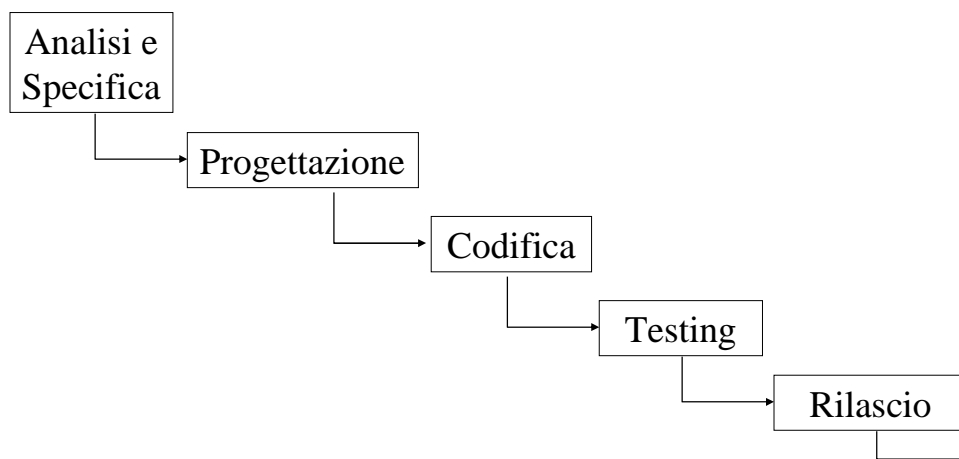


Esempio

Problema: risolvere l'equazione di I grado $ax+b=0$, con i coefficienti a e b numeri reali

Sviluppo di programmi

... mediante una serie di fasi ...



Sviluppo di programmi

Analisi e Specifica

Definizione di *cosa fa* il programma

Individuazione e specificazione

- dei *dati* di *ingresso* e di *uscita*
- delle *precondizioni*
- delle *postcondizioni*

Definizione di un *dizionario dei dati*

Sviluppo di programmi

Progettazione

- Definizione di *come* il programma effettuerà la trasformazione specificata;

Progettazione dell' algoritmo per raffinamenti successivi
(*stepwise refinement*) ...

Decomposizione funzionale ...

Codifica

Redazione del programma con il linguaggio di programmazione scelto

Sviluppo di programmi

Testing

Una volta effettuata la codifica e la compilazione del programma bisogna 'testarlo' per verificare la eventuale presenza di malfunzionamenti nell'esecuzione del programma dovuti a difetti presenti nel codice

Sviluppo di programmi

Testing

Una volta effettuata la codifica e la compilazione del programma bisogna 'testarlo' per verificare la eventuale presenza di malfunzionamenti nell'esecuzione del programma dovuti a difetti presenti nel codice

- Definizione dei casi di prova:

- Definizione/selezione di valori delle variabili di input con cui 'testare' (provare) il programma
- Esecuzione del programma
- Verifica dei risultati ottenuti rispetto ai *risultati attesi* (corretti)

.... Se risultati ottenuti sono diversi da quelli attesi, il test è positivo in quanto ha rilevato un malfunzionamento ...

.... Il testing ha lo scopo di rivelare e rilevare i malfunzionamenti ...

... rimozione difetti che li causano (Debugging) ...

Sviluppo di programmi

Scelta dei casi di prova e verifica

I casi di prova non vanno scelti “a casaccio”

l'obiettivo è quello di una verifica il più possibile esaustiva ...

(NB: in generale però è impossibile effettuare un testing esaustivo ...)

... cercare di evitare i casi di prova ridondanti ...

... i casi di prova e i risultati del testing vanno documentati ...

Sviluppo di programmi

I malfunzionamenti sono dovuti a difetti (errori) presenti nel codice del programma

Tipologie di difetti in programmi

-Difetti di tipo lessicale/sintattico

- individuati dal compilatore
- dovuti ad errata scrittura o a mancanza di conoscenza delle regole del linguaggio (ad es. mancanza del “;” alla fine di una istruzione)

- Difetti di tipo logico

dovuti ad una errata definizione dell'algoritmo (o sua trascrizione durante la codifica, ad es. scrittura dell'operatore + invece del *)

.... *provocano malfunzionamenti durante l'esecuzione del programma possono essere difficili da individuare e correggere*

Sviluppo di programmi

Difetti di tipo Logico

Possono essere dovuti ad errori di diverso tipo, commessi durante lo sviluppo del programma:

- Errori nella definizione del problema (fasi di analisi e specifica)
- Errori di progettazione (es. nella definizione/scelta dell'algoritmo)
- Errori nella stesura (*codifica*) dei programmi

Più alta è la fase di sviluppo in cui è commesso l'errore generante il difetto più è difficile individuarlo e correggerlo:

correzione degli errori => eliminazione dei difetti => *debugging*

Esempio

Analisi e Specifica

Definizione del problema: risolvere l'equazione di I grado $ax+b=0$, con i coefficienti a e b numeri reali

Definizione dei dati del problema:

I: A, B coefficienti dell'equazione, di tipo reale

Pi:

U: X , soluzione dell'equazione, di tipo reale

Pu: per $A=0, B \neq 0$, x è indefinito, equazione impossibile

per $A=0, B=0$, equazione indeterminata

Esempio

Tabella delle variabili di ingresso

Nome variabile	Descrizione	Tipo
A, B	Coefficienti dell'equazione	FLOAT

Tabella delle variabili di uscita

Nome variabile	Descrizione	Tipo
X	Risultato dell'equazione	FLOAT

Esempio

Progettazione

Descrizione del metodo di elaborazione:

- sono letti i valori dei coefficienti A e B;
- se $A = 0$ stampare 'equazione impossibile' se $B \neq 0$ oppure 'equazione indeterminata' se $B = 0$
- se $A \neq 0$ si calcola il risultato dell'equazione $X := -B/A$ e lo si visualizza;

Esempio

Codifica

- sono letti i valori dei coefficienti A e B;
write ('immetti coefficiente A ');
read (A);
write ('immetti coefficiente B ');
read (B);

Esempio

Codifica

- se $A = 0$ stampare 'equazione impossibile' se $B \neq 0$ oppure 'equazione indeterminata' se $B = 0$

```
if A=0 then
    if B<>0 then
        write ('Equazione Impossibile')
    else
        write ('Equazione indeterminata')
    endif
```

.....

Esempio

Codifica

- se $A \neq 0$ si calcola il risultato dell'equazione $X := -B/A$ e lo si visualizza;

```
    if A=0 then
        .....
    else
        begin
            X = -B/A;
            write ('X = ', X);
        end
    endif
```

Esempio - il programma in LDP

```
Program equazioneIgrado;
FLOAT A, B, X;
begin
write ('immetti coefficiente A ');
read (A);
write ('immetti coefficiente B ');
read (B);
```

Esempio - il programma in LDP

```
if A=0 then
  if B<>0 then
    write ('Equazione Impossibile')
  else
    write ('Equazione indeterminata')
  endif
else
  begin
    X = -B/A;
    write ('X = ', X);
  end
endif
end.
```

Elementi di Informatica
Prof. G. A. Di Lucca - Univ. del Sannio

17

Risoluzione Equazione I grado - LDP

```
Program equazioneIgrado;
FLOAT A, B, X;
begin
write ('immetti coefficiente A ');
read (A);
write ('immetti coefficiente B ');
read (B);
```

```
if A=0
then
  if B<>0
  then
    write ('Equazione Impossibile')
  else
    write ('Equazione indeterminata')
  endif
else
  begin
    X = -B/A;
    write ('X = ', X);
  end
endif
end.
```

Elementi di Informatica
Prof. G. A. Di Lucca - Univ. del Sannio

18

Esempio

Testing

criteri per la scelta dei casi di prova:

almeno un caso di test (prova) con valori di A e B entrambi $\neq 0$

un caso di test (prova) con $A = 0$ e $B \neq 0$

un caso di test (prova) con $A = 0$ e $B = 0$

Risoluzione Equazione I grado - LDP

Esecuzione 'carta e penna'

```
Program equazioneIgrado;  
FLOAT A, B, X;  
begin  
write ('immetti coefficiente A ');  
read (A);  
write ('immetti coefficiente B ');  
read (B);
```

Supponendo che l'equazione sia

$$3x + 9 = 0$$

Primo caso di prova:

$$A=3$$

$$B=9$$

Risoluzione Equazione I grado - LDP

```
if A=0
  then
    if B<>0
      then
        write ('Equazione Impossibile')
      else
        write ('Equazione indeterminata')
    endif
  else
    begin
      X = -B/A;
      write ('X = ', X);
    end
  endif
end.
```

Esecuzione 'carta e penna'

```
A= 0 ?
      Falso

else
X = - B/A = - 9/3 = -3

X= -3
```

Risoluzione Equazione I grado - LDP

```
Program equazioneIgrado;
FLOAT A, B, X;
begin
write ('immetti coefficiente A ');
read (A);
write ('immetti coefficiente B ');
read (B);
```

Esecuzione 'carta e penna'

Secondo caso di prova:

```
A=0
B=9
```

Risoluzione Equazione I grado - LDP

```
if A=0
  then
    if B<>0
      then
        write ('Equazione Impossibile')
      else
        write ('Equazione indeterminata')
    endif
  else
    begin
      X = -B/A;
      write ('X = ', X);
    end
  endif
end.
```

Esecuzione 'carta e penna'

```
A= 0 ?
      VERO
  then
B<> 0 ?
      VERO
  then
Equazione Impossibile
```

Risoluzione Equazione I grado - LDP

```
Program equazioneIgrado;
Float A, B, X;
begin
write ('immetti coefficiente A ');
read (A);
write ('immetti coefficiente B ');
read (B);
```

Esecuzione 'carta e penna'

```
Terzo caso di prova:
A = 0
B = 0
```

Risoluzione Equazione I grado - LDP

```
if A=0
  then
    if B<>0
      then
        write ('Equazione Impossibile')
      else
        write ('Equazione indeterminata')
    endif
  else
    begin
      X = -B/A;
      write ('X = ', X);
    end
  endif
end.
```

Esecuzione 'carta e penna'

```
A= 0 ?
      VERO
then
B<> 0 ?
      FALSO

else
Equazione indeterminata
```

Risoluzione Equazione I grado - il programma in C

LDP

```
Program equazioneIgrado;
FLOAT A, B, X;
begin
write ('immetti coefficiente A ');
read (A);
write ('immetti coefficiente B ');
read (B);
```

C

```
#include <stdio.h>
main ()
{
// questo programma risolve un'equazione di I grado
float A, B, X;
printf ("immetti coefficiente A \n");
scanf ("%d", &A);
printf ("immetti coefficiente B \n");
scanf ("%d", &B);
```

Risoluzione Equazione I grado - il programma in C

LDP

```
if A=0
then
  if B<>0
  then
    write ('Equazione Impossibile')
  else
    write ('Equazione indeterminata')
  endif
else
  begin
    X = -B/A;
    write ('X = ', X);
  end
endif
end.
```

C

```
if (A==0)
{ if (B!=0)
  printf ("Equazione Impossibile \n");
  else
  printf ("Equazione indeterminata \n");
}
else
{
  X= -B/A;
  printf("X = %f \n", X);
}
```

Esempio

Analisi e Specifica

Definizione del problema: Assegnati tre numeri interi trovare quello con valore massimo

Definizione dei dati del problema:

I: i tre numeri di tipo intero, A, B, C

Pi: nessuna

U: MAX, il numero con il valore massimo, di tipo intero

Pu: nessuna

Esempio

Tabella delle variabili di ingresso

Nome variabile	Descrizione	Tipo
A, B,C	I tre numeri interi tra cui trovare il massimo	INT

Tabella delle variabili di uscita

Nome variabile	Descrizione	Tipo
MAX	Il numero con valore massimo	INT

Esempio

Progettazione

Descrizione del metodo di elaborazione:

- sono letti i valori dei tre numeri interi;
- inizialmente si considera il primo come quello massimo;
- lo si confronta con il secondo e si decide quale è quello massimo tra questi due;
- il numero massimo risultante dal precedente confronto viene confrontato con il terzo e si decide quale è il massimo tra essi, che sarà anche il massimo tra tutti e tre;
- si stampa il valore massimo così trovato

Esempio

Codifica

- sono letti i valori dei tre numeri interi;

```
write ('immetti il primo numero A ');  
read (A);  
write ('immetti il secondo numero B ');  
read (B);  
write ('immetti il terzo numero C ');  
read (C);
```

Esempio

Codifica

- inizialmente si considera il primo come quello massimo;

```
MAX=A;
```

- lo si confronta con il secondo e si decide quale è quello massimo tra questi due;

```
if B>MAX then  
    MAX=B  
endif
```

Esempio

Codifica

- il numero massimo risultante dal precedente confronto viene confrontato con il terzo e si decide quale è il massimo tra essi, che sarà anche il massimo tra tutti e tre;

```
if C>MAX then
    MAX=C
endif
```

- si stampa il valore massimo così trovato

```
write ('Valore massimo = ', MAX);
```

Esempio - il programma in LDP

```
Program massimo_tra_tre;
INT A, B, C, MAX;
begin
write ('immetti il primo numero A ');
read (A);
write ('immetti il secondo numero B ');
read (B);
write ('immetti il terzo numero C ');
read (C);
MAX=A;
if B>MAX then
    MAX=B
endif
if C>MAX then
    MAX=C
endif
write ('Valore massimo = ', MAX);
end.
```

Esempio

Testing

criteri per la scelta dei casi di prova:

almeno un caso di test (prova) con valori di A e B e C tutti diversi tra loro

un caso di test (prova) con due numeri uguali ed il terzo diverso

un caso di test (prova) con tutti e tre i numeri uguali

LDP

Max fra tre - il programma in C

```

Program massimo_tra_tre;
INT A, B, C, MAX;
begin
write ('immetti il primo numero A ');
read (A);
write ('immetti il secondo numero B ');
read (B);
write ('immetti il terzo numero C ');
read (C);
MAX=A;
if B>MAX then
    MAX=B
endif
if C>MAX then
    MAX=C
endif
write ('Valore massimo = ', MAX);
end.
    
```

```

#include <stdio.h>
main ()
{
    // massimo_tra_tre numeri interi
    int A, B, C, MAX;
    printf ("immetti il primo numero A \n");
    scanf ("%d", &A);
    printf ("immetti il secondo numero B \n");
    scanf ("%d", &B);
    printf ("immetti il terzo numero C \n");
    scanf ("%d", &C);
    MAX=A;
    if (B>MAX) MAX=B;
    if (C>MAX) MAX=C;
    printf ("Valore massimo = %d \n",
    MAX);
}
    
```

Esempio

Analisi e Specifica

Definizione del problema: Calcolare le soluzioni del sistema di due equazioni lineari

$$ax+by=e$$

$$cx+dy=f$$

Definizione dei dati del problema:

I: a,b,c,d,e,f, coefficienti del sistema di tipo intero

Pi: determinante dei coefficienti diverso da zero

U: x, y soluzioni del sistema, di tipo reale

Pu: se il determinante dei coefficienti è uguale a zero, x e y sono indefiniti ed il sistema impossibile

Elementi di Informatica
Prof. G. A. Di Lucca - Univ. del Sannio

37

Esempio

Tabella delle variabili di ingresso

Nome variabile	Descrizione	Tipo
A, B,C,D,E,F	I coefficienti del sistema	INT

Tabella delle variabili di uscita

Nome variabile	Descrizione	Tipo
X, Y	Soluzioni del sistema	REAL

Tabella delle variabili di algoritmo

Nome variabile	Descrizione	Tipo
G	Determinate dei coefficienti x e y	INT

Elementi di Informatica
Prof. G. A. Di Lucca - Univ. del Sannio

38

Esempio

Progettazione

Descrizione del metodo di elaborazione:

- sono letti i valori dei coefficienti;
- si calcola il determinante dei coefficienti ($g=(a*d)-(b*c)$)
- se g è diverso da zero si calcolano le soluzioni

$$x=((e*d)-(f*b))/g$$

$$y=((a*f)-(c*e))/g$$

- se $g=0$ stampare 'Sistema impossibile'

Esempio

Codifica

- sono letti i valori dei coefficienti;

```
write ('immetti i coefficienti ');
```

```
read (A, B, C, D, E, F);
```

- si calcola il determinante dei coefficienti

```
G=(A*D)-(B*C);
```

Esempio

Codifica

- se $g \neq 0$ si calcolano le soluzioni

```
if G<>0 then
  begin
    X=(E*D)-(F*B)/G ;
    Y=((A*F)-(C*E))/G;
    write ('le soluzioni sono X=', X, ' Y = ', Y);
  end
```

Esempio

Codifica

- se $g=0$ stampare 'Sistema impossibile'

```
if G<>0 then
  .....
else
  write ('Sistema Impossibile');
endif
```

```
Program sistema_2_equazioni;  
INT A, B, C, D, E, F,G;  
REAL X, Y;  
begin  
write ('immetti il coefficiente A ');  
read (A);  
....  
write ('immetti il coefficiente F ');  
read (F);  
G=(A*D)-(B*C);  
if G<>0 then  
  begin  
    X=((E*D)-(F*B))/G ;  
    Y=((A*F)-(C*E))/G;  
    write ('le soluzioni sono X=', X, ' Y = ', Y);  
  end  
else  
  write ('Sistema Impossibile');  
endif  
end.
```

Esempio - il programma in LDP

Esempio

Testing

criteri per la scelta dei casi di prova:

.... Lo studente definisca, quale esercizio, i casi di prova ...

Risoluzione sistema_2_equazioni - il programma in C

```

Program sistema_2_equazioni;
INT A, B, C, D, E, F,G;
REAL X, Y;
begin
write ('immetti il coefficiente A ');
read (A);
....
write ('immetti il coefficiente F ');
read (F);
G=(A*D)-(B*C);
if G<>0 then
begin
X=(E*D)-(F*B)/G ;
Y=((A*F)-(C*E))/G;
write ('le soluzioni sono X=', X, ' Y = ', Y);
end
else
write ('Sistema Impossibile');
endif
end.
    
```

```

#include <stdio.h>
main ()
{ // sistema_2_equazioni
int A, B, C, D, E, F,G;
float X, Y;
printf ("immetti il coefficiente A \n");
scanf ("%d", &A);
printf ("immetti il coefficiente B \n");
scanf ("%d", &B);
printf ("immetti il termine noto E \n");
scanf ("%d", &E);
printf ("immetti il coefficiente C \n");
scanf ("%d", &C);
printf ("immetti il coefficiente D \n");
scanf ("%d", &D);
printf ("immetti il termine noto F \n");
scanf ("%d", &F);
    
```

```

G=(A*D)-(B*C);
if G<>0 then
begin
X=(E*D)-(F*B)/G ; Y=((A*F)-(C*E))/G;
write ('le soluzioni sono X=', X, ' Y = ', Y);
end
else
write ('Sistema Impossibile');
endif
end.
    
```

Risoluzione sistema_2_equazioni - il programma in C

LDP

C

```

G=(A*D)-(B*C);
if (G!=0)
{ X=((float)(E*D)-(float) (F*B))/(float)G ;
Y=((float)(A*F)-(float) (C*E))/(float)G;
printf ("le soluzioni sono X= %f Y= %f \n", X, Y);
}
else
printf("Sistema Impossibile \n");
}
    
```

Esempio

Analisi e Specifica

Definizione del problema: dato in input un numero compreso tra 1 e 12 indicare, il nome del mese corrispondente, il numero di giorni componenti il mese corrispondente a quel numero (per febbraio si assume 28 giorni) e stampare un asterisco '*' se il mese è un mese invernale, il segno '+' se è estivo, il segno '=' se primaverile o autunnale (si considera come stagione quella corrente all'inizio del mese)

Definizione dei dati del problema:

I: un numero intero

Pi: il numero deve essere compreso tra 1 e 12

U: numero di giorni componenti il mese, segno della stagione

Pu: nessuna

Esempio

Tabella delle variabili di ingresso

Nome variabile	Descrizione	Tipo
NumMese	numero indicante un mese	INT

Tabella delle variabili di uscita

Nome variabile	Descrizione	Tipo
Giorni	numero dei giorni del mese	INT
Stagione	simbolo indicante la stagione	CHAR

Esempio

Progettazione

Descrizione del metodo di elaborazione:

è letto il valore del numero in ingresso NumMese

Se:

NumMese è 1: Giorni=31, Stagione='*', stampa "Gennaio",

.....

NumMese è 4: Giorni=30, Stagione='=', stampa "Aprile",

.....

NumMese è 7: Giorni=31, Stagione='+', stampa "Luglio",

se NumMese non è compreso tra 1 e 12 stampa 'Mese inesistente'

Se (0<NumMese <12) stampa i valori di Giorni e Stagione.

Esempio - il programma in C

```
#include<stdio.h>
main()
{
  int NumMese, Giorni;
  char Stagione;
  printf("Dammi un mese (numero compreso tra 1 e 12): ");
  scanf("%d", &NumMese);
```

```

switch (NumMese)
{
case 1: Giorni=31; Stagione='*'; printf("Gennaio "); break;
case 2: Giorni=28; Stagione='*'; printf("Febbraio "); break;
case 3: Giorni=31; Stagione='*'; printf("Marzo "); break;
case 4: Giorni=30; Stagione='='; printf("Aprile "); break;
case 5: Giorni=31; Stagione='='; printf("Maggio "); break;
case 6: Giorni=30; Stagione='='; printf("Giugno "); break;
case 7: Giorni=31; Stagione='+'; printf("Luglio "); break;
case 8: Giorni=31; Stagione='+'; printf("Agosto "); break;
case 9: Giorni=30; Stagione='+'; printf("Settembre "); break;
case 10: Giorni=31; Stagione='='; printf("Ottobre "); break;
case 11: Giorni=30; Stagione='='; printf("Novembre "); break;
case 12: Giorni=31; Stagione='='; printf("Dicembre "); break;
default: printf("Mese inesistente"); break;
}
if ((NumMese>0) && (NumMese<13))
printf("Il mese ha: %d giorni e stagione = %c \n", Giorni, Stagione);
}
                
```

Esempio - il programma in C

Elementi di Informatica
Prof. G. A. Di Lucca - Univ. del Sannio

51

Esempio - il programma in C

... altra possibile soluzione ...

```

#include<stdio.h>
main()
{
int NumMese, Giorni;
char Stagione;
printf("Dammi un mese (numero compreso tra 1 e 12): ");
scanf("%d", &NumMese);
                
```

Elementi di Informatica
Prof. G. A. Di Lucca - Univ. del Sannio

52

```
if ((NumMese>0) && (NumMese<13))
{ switch (NumMese)
  { case 1: Giorni=31; Stagione='*'; printf("Gennaio "); break;
    case 2: Giorni=28; Stagione='*'; printf("Febbraio "); break;
    case 3: Giorni=31; Stagione='*'; printf("Marzo "); break;
    case 4: Giorni=30; Stagione='='; printf("Aprile "); break;
    case 5: Giorni=31; Stagione='='; printf("Maggio "); break;
    case 6: Giorni=30; Stagione='='; printf("Giugno "); break;
    case 7: Giorni=31; Stagione='+'; printf("Luglio "); break;
    case 8: Giorni=31; Stagione='+'; printf("Agosto "); break;
    case 9: Giorni=30; Stagione='+'; printf("Settembre "); break;
    case 10: Giorni=31; Stagione='='; printf("Ottobre "); break;
    case 11: Giorni=30; Stagione='='; printf("Novembre "); break;
    case 12: Giorni=31; Stagione='='; printf("Dicembre "); break;
  }
  printf("Il mese ha: %d giorni e stagione = %c \n", Giorni, Stagione);
}
else
  printf("Mese inesistente");
}
```

Esempio - il programma in C

Esempio

Analisi e Specifica

Definizione del problema: Dato in input il valore di N , numero intero maggiore di zero, calcolare la sommatoria dei numeri da 1 a N

Definizione dei dati del problema:

I: N

Pi: $N > 0$

U: la sommatoria dei numeri da 1 a N

Pu: nessuna

Esempio

Tabella delle variabili di ingresso

Nome variabile	Descrizione	Tipo
N	estremo superiore della sommatoria	INT

Tabella delle variabili di uscita

Nome variabile	Descrizione	Tipo
Sommatoria	Sommatoria	INT

Esempio

Progettazione

Descrizione del metodo di elaborazione:

- è letto il valore di N
- si verifica il valore di N e la lettura è ripetuta se $N \leq 0$
- si inizializza il valore di Sommatoria a zero
- si inizializza il valore di i a uno
- si ripetono, mentre i è non maggiore di N, le seguenti due istruzioni:
 - si incrementa di i il valore di Sommatoria
 - si incrementa di (uno) 1 il valore di i
- alla fine del ciclo si visualizza il valore di Sommatoria

Esempio - il programma in C

```
#include<stdio.h>
main()
{ // Calcola la sommatoria dei numeri interi da 1 a N
  int i, Sommatoria, N;
  // legge il valore di N finché è maggiore di zero
  printf ("Immetti il valore di N \n");
  scanf ("%d", &N);
  while (N<=0)
  { printf ("Immetto un valore non positivo, Immetti il valore di N\n");
    scanf ("%d", &N);
  }
  // inizializza il valore di Sommatoria a zero e di i a 1
  Sommatoria=0; i=1;

  .....
```

Esempio - il programma in C

```
#include<stdio.h>
main()
{ // Calcola la sommatoria dei numeri interi da 1 a N
  int i, Sommatoria, N;
  // legge il valore di N finché è maggiore di zero
  printf ("Immetti il valore di N \n");
  scanf ("%d", &N);
  while (N<=0)
  { printf ("Immetto un valore non positivo, Immetti il valore di N\n");
    scanf ("%d", &N);
  }
  // inizializza il valore di Sommatoria a zero e di i a 1
  Sommatoria=0; i=1;

  // ciclo while per calcolare la sommatoria
  while (i<=N)
  Sommatoria=Sommatoria + i;
  printf ("Sommatoria = %d \n", Sommatoria);
}
```

Errore!! Manca l'incremento
della variabile i !!!
Sommatoria incrementata solo
di 1 (uno) !
Ciclo NON convergente !

Esempio - il programma in C

```
#include<stdio.h>
main()
{ // Calcola la sommatoria dei numeri interi da 1 a N
  int i, Sommatoria, N;
  // legge il valore di N finché è maggiore di zero
  printf ("Immetti il valore di N \n");
  scanf ("%d", &N);
  while (N<=0)
  { printf (" Immesso un valore non positivo, Immetti il valore di N\n");
    scanf ("%d", &N);
  }
  // inizializza il valore di Sommatoria a zero e di i a 1
  Sommatoria=0; i=1;

  // ciclo while per calcolare la sommatoria
  while (i<=N)
  {Sommatoria=Sommatoria + i;
   i++; }
  printf("Sommatoria = %d \n", Sommatoria);
}
```

Elementi di Informatica
Prof. G. A. Di Lucca - Univ. del Sannio

59

Esempio

... un modo differente per risolvere lo stesso problema con l'uso di cicli a condizione finale ...

Analisi e Specifica

Definizione del problema: Dato in input il valore di **N**, numero intero maggiore di zero, calcolare la sommatoria dei numeri da 1 a **N**

Definizione dei dati del problema:

- I: **N**
- Pi: **N>0**
- U: la sommatoria dei numeri da 1 a **N**
- Pu: nessuna

Elementi di Informatica
Prof. G. A. Di Lucca - Univ. del Sannio

60

Esempio

Tabella delle variabili di ingresso

Nome variabile	Descrizione	Tipo
N	estremo superiore della sommatoria	INT

Tabella delle variabili di uscita

Nome variabile	Descrizione	Tipo
Sommatoria	Sommatoria	INT

Esempio

Progettazione

Descrizione del metodo di elaborazione:

- è letto il valore di N
- la lettura è ripetuta se $N \leq 0$
- si inizializza il valore di Sommatoria a zero
- si inizializza il valore di i a uno
 - si incrementa di i il valore di Sommatoria
 - si incrementa di 1 (uno) il valore di i
- si ripetono, con un ciclo, i due passi precedenti finché i è non maggiore di N
- alla fine del ciclo si visualizza il valore di Sommatoria

Esempio - il programma in C

```
#include<stdio.h>
main()
{ // Calcola la sommatoria dei numeri interi da 1 a N
  int i, Sommatoria, N;
  // legge il valore di N finché è maggiore di zero
  do
  { printf (" Immetti il valore di N\n");
    scanf("%d", &N);
  }
  while (N<=0);
  // inizializza il valore di Sommatoria a zero e di i a 1
  Sommatoria=0; i=1;
  // ciclo per calcolare la sommatoria
  do
  {Sommatoria=Sommatoria + i;
   i++; }
  while (i<=N) ;
  printf("Sommatoria = %d \n", Sommatoria);
}
```

Elementi di Informatica
Prof. G. A. Di Lucca - Univ. del Sannio

63

Esempio

... un ulteriore modo differente per risolvere lo stesso problema con l'uso di ciclo ripetitivo ...

Analisi e Specifica

Definizione del problema: Dato in input il valore di **N**, numero intero maggiore di zero, calcolare la sommatoria dei numeri da 1 a N

Definizione dei dati del problema:

- I: N
- Pi: N>0
- U: la sommatoria dei numeri da 1 a N
- Pu: nessuna

Elementi di Informatica
Prof. G. A. Di Lucca - Univ. del Sannio

64

Esempio

Tabella delle variabili di ingresso

Nome variabile	Descrizione	Tipo
N	estremo superiore della sommatoria	INT

Tabella delle variabili di uscita

Nome variabile	Descrizione	Tipo
Sommatoria	Sommatoria	INT

Esempio

Progettazione

Descrizione del metodo di elaborazione:

- è letto il valore di N, la lettura è ripetuta se $N \leq 0$
- si inizializza il valore di Sommatoria a zero
- si ripete N volte, con un ciclo for:
- si aggiunge al valore di Sommatoria il valore della variabile di controllo del ciclo for
- alla fine del ciclo for si visualizza il valore di Sommatoria

Esempio - il programma in C

```
#include<stdio.h>
main()
{ // Calcola la sommatoria dei numeri interi da 1 a N
  int i, Sommatoria, N;
  // legge il valore di N finché è maggiore di zero
  do
  { printf ("Immetti il valore di N, maggiore di zero \n");
    scanf ("%d", &N); }
  while (N<=0);
  // inizializza il valore di somma a zero
  Sommatoria=0;
  // ciclo ripetitivo per calcolare la sommatoria
  for (i=1; i<=N; i++)
    Sommatoria=Sommatoria + i;
  printf ("Sommatoria = %d \n", Sommatoria);
}
```

Esempio

Analisi e Specifica

Definizione del problema: Dato in input il valore di **N**, numero intero maggiore di zero, calcolare la sommatoria di **N** numeri interi immessi dall'utente

Definizione dei dati del problema:

I: **N** indicante su quanti numeri è effettuata la sommatoria;
un numero intero elemento della sommatoria

Pi: $N > 0$

U: la sommatoria di **N** numeri

Pu: nessuna

Esempio

Tabella delle variabili di ingresso

Nome variabile	Descrizione	Tipo
N	estremo superiore della sommatoria	INT
Numero	elemento della sommatoria	INT

Tabella delle variabili di uscita

Nome variabile	Descrizione	Tipo
Sommatoria	Sommatoria	INT

Esempio

Progettazione

Descrizione del metodo di elaborazione:

- è letto il valore di N, la lettura è ripetuta se $N \leq 0$
- si inizializza il valore di Sommatoria a zero
- si ripete N volte:
 - la lettura di un numero intero
 - si aggiunge al valore di Sommatoria il valore del numero letto
- alla fine si visualizza il valore di Sommatoria

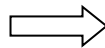
Esempio - il programma in C

```
#include<stdio.h>
main()
{ // Calcola la sommatoria di N numeri interi
  int i, Sommatoria, Numero, N;
  // legge N finché è maggiore di zero
  do
  { printf ("Immetti il valore di N \n");
    scanf ("%d", &N); }
  while (N<=0);
  // inizializza Sommatoria a zero
  Sommatoria=0;
```

```
// ciclo ripetitivo per calcolare la sommatoria
for (i=1; i<=N; i++)
{ // Legge un numero della sommatoria
  printf ("Dammi il numero %d
         della sommatoria: \n ", i");
  scanf ("%d", &Numero);
  // aggiunge Numero a Sommatoria
  Sommatoria=Sommatoria + Numero;
}
printf("Sommatoria = %d \n",
       Sommatoria);
}
```

... supponiamo di eseguire il programma per calcolare la sommatoria di 4 numeri: (6, 9, 12, 18)

```
do
  printf ("Immetti il valore di N \n");
  scanf ("%d", &N);
while (N<=0);
```



Immetti il valore di N
4

4 <= 0 ? Falso !



Sommatoria=0

```

for (i=1; i<=N; i++)
{
// Legge un numero della sommatoria
printf ("Dammi il numero %d: ", i, "della sommatoria \n");
scanf ("%d", &Numero);
// aggiunge il valore letto a quello della sommatoria
Sommatoria=Sommatoria + Numero;
}
printf("Sommatoria = %d \n", Sommatoria);
    
```

↓

i = 1	Dammi il numero 1 della sommatoria 6	Sommatoria = 6
i = 2	Dammi il numero 2 della sommatoria 9	Sommatoria = 15
i = 3	Dammi il numero 3 della sommatoria 12	Sommatoria = 27
i = 4	Dammi il numero 4 della sommatoria 18	Sommatoria = 45
i = 5	Sommatoria = 45	

Valori
variabile
controllo del
ciclo →

Elementi di Informatica
Prof. G. A. Di Lucca - Univ. del Sannio

73

Esempio

Testing

criteri per la scelta dei casi di prova:

.... Lo studente definisca, quale esercizio, i casi di prova ...

Elementi di Informatica
Prof. G. A. Di Lucca - Univ. del Sannio

74

Esempio

Analisi e Specifica

Definizione del problema: dati due operandi numeri interi NON negativi ed un segno di una operazione aritmetica (+, -, *, /) calcolare il risultato dell'operazione indicata dal segno applicato ai due operandi, considerati secondo l'ordine di immissione (ovvero l'operatore è applicato tra il primo ed il secondo operando immesso). Nel calcolo dell'operazione aritmetica usare solo le operazioni di somma e differenza

Esempio

Analisi e Specifica

Definizione del problema: dati due operandi numeri interi NON negativi ed un segno di una operazione aritmetica (+, -, *, /) calcolare il risultato dell'operazione indicata dal segno applicato ai due operandi, considerati secondo l'ordine di immissione (ovvero l'operatore è applicato tra il primo ed il secondo operando immesso). Nel calcolo dell'operazione aritmetica usare solo le operazioni di somma e differenza

Definizione dei dati del problema:

I: due numeri interi; il segno dell'operazione aritmetica

Pi: i due operandi sono NON negativi

U: il risultato dell'operazione di tipo intero

Pu: nessuna

Esempio

Tabella delle variabili di ingresso

Nome variabile	Descrizione	Tipo
OP1, OP2	i due operandi	INT
Segno	segno dell'operazione	CHAR

Tabella delle variabili di uscita

Nome variabile	Descrizione	Tipo
Risultato	risultato dell'operazione	INT

Esempio

Progettazione

Descrizione del metodo di elaborazione:

sono letti i valori dei due operandi e del segno dell'operazione

Si verifica che i due operandi sono NON negativi, se ciò è vero si prosegue e

se il segno è '+' fare la somma dei due operandi;

se il segno è '-' fare la differenza dei due operandi;

se il segno è '*' calcolare il prodotto dei due operandi con somme successive;

se il segno è '/' calcolare la divisione tra due operandi mediante differenze successive

Si riduce il primo operando (dividendo) del secondo operando immesso (divisore), il risultato della sottrazione è ancora diminuito del dividendo e si continua così fin quando si avrà un risultato della sottrazione minore del divisore; il quoziente è pari al numero di sottrazioni effettuate

Si stampa il valore del Risultato

Esempio - il programma C

```
#include <stdio.h>
main ()
{
// programma scegli operazione
int OP1, OP2, Risultato, I;
char Segno;
printf ("immetti segno operazione \n");
scanf ("%c", &Segno);
do
{
printf("immetti i due operandi \n");
printf(" operando 1 \n");
scanf ("%d", &OP1);
printf (" operando 2 \n");
scanf ("%d", &OP2);
}
while ((operando1<0)||((operando2<0));
```

Esempio - il programma C

```
switch (Segno)
{
case '+': Risultato = OP1 + OP2;
break;
case '-': Risultato = OP1 - OP2; break;
case '*': // con struttura while ( )
{ Risultato=0;
I=1;
while (I<=OP2)
{ Risultato=Risultato+OP1;
I++;
}
break;
}
}
```

```
case '/':
{ Risultato=0;
if (OP2==0)
printf("divisione impossibile \n");
else
while (OP2<=OP1)
{ OP1=OP1-OP2;
Risultato=Risultato + 1;
}
break;
}
default :
{ printf("Segno non valido \n");
Risultato=0;}
}
printf("risultato = %d \n", Risultato);
}
```

Esempio - il programma C

..... soluzione con uso della struttura for () per la moltiplicazione

switch (Segno)

```
{
case '+' : Risultato = OP1 + OP2; break;
case '-' : Risultato = OP1 - OP2; break;
case '*': // con struttura for ( )
    {Risultato=0;
      for (I=1; I <= OP2; I++)
        Risultato=Risultato+OP1;
      break;
    }
}
```

switch (Segno)

```
{ .....
  case '*': {Risultato=0;
    I=1;
    while (I<=OP2)
      {Risultato=Risultato+OP1;
        I++;
      }
    break;
  }
}
```

Esempio - il programma C

... analogia tra strutture while () e for ()

switch (Segno)

```
{
.....
case '*': {Risultato=0;
  for (I=1; I <= OP2; I++)
    Risultato=Risultato+OP1;
  break;
}
```

Esempio

Analisi e Specifica

Definizione del problema: Sviluppare un programma per giocare a '30 cerini'.

Regole del gioco:

- Ci sono 30 cerini e 2 giocatori.
- Ciascun giocatore, a turno, prende minimo 3 cerini e massimo 7 cerini.
- Perde il giocatore che prende l'ultimo cerino.

Esempio

Analisi e Specifica

Definizione dei dati del problema:

I: gli identificativi di ciascun giocatore;

la quantità di cerini presi da un giocatore

P_i : la quantità di cerini presi da un giocatore deve essere compresa tra 3 e 7, estremi inclusi

U: identificativo del giocatore che ha vinto

P_u : nessuna

Esempio

Tabella delle variabili di ingresso

Nome variabile	Descrizione	Tipo
presi	N.ro cerini presi da un giocatore	INT
Nome1	Identificativo del primo giocatore	CHAR
Nome2	Identificativo del secondo giocatore	CHAR

Esempio

Tabella delle variabili di uscita

Nome variabile	Descrizione	Tipo
Vince	Identificativo del giocatore vincente	CHAR

Tabella delle variabili di algoritmo

Nome variabile	Descrizione	Tipo
TotCerini	Numero Totale dei cerini in gioco	INT
Gioca	Identificativo del giocatore attuale	INT

Progettazione

Esempio

Descrizione del metodo di elaborazione:

1. È inizializzato a 30 il valore di TotCerini
2. Si immettono gli identificativi del primo e secondo giocatore
3. Alla variabile Gioca è assegnato il valore di Nome1 (giocatore attuale)
4. Il giocatore attuale immette quanti cerini prende (ovvero si legge il valore della variabile 'presi'); l'immissione di tale valore va ripetuta se è immesso un valore non compreso tra 3 e 7 o se si è immesso un valore maggiore dei cerini ancora in gioco
5. il valore di TotCerini è decrementato di 'presi'
6. Se restano al più tre cerini (cioè $0 < \text{TotCerini} \leq 3$) il giocatore attuale ha vinto (Vince = Gioca); se TotCerini==0 ha vinto l'altro giocatore (ovvero Vince = Nome2 se Gioca==Nome1, o Vince = Nome1 se Gioca==Nome2);
7. Se TotCerini è maggiore di 3 tocca all'altro giocatore (cioè alla variabile Gioca è assegnato il valore di Nome2, se Gioca==Nome1, o il valore di Nome1 se Gioca==Nome2)
8. Si ripetono le operazioni dal punto 4 fin quando restano più di tre cerini;
9. Si stampa il valore della variabile Vince.

Esempio - il programma C

```
#include <stdio.h>
main ()
{ // 30 cerini
  char Nome1, Nome2, Vince, Gioca;
  int presi, TotCerini;
  printf ("Immetti identificativo primo giocatore\n");
  scanf ("%c", &Nome1);
  printf ("Immetti identificativo secondo giocatore\n");
  scanf ("\n%c", &Nome2);
  TotCerini=30;
  Gioca=Nome1;
  printf(" Ci sono %d cerini\n", TotCerini);
```

```
do
{
  do
  { printf ("Quanti cerini prende %c ?\n", Gioca);
    scanf ("%d", &presi);
  }
  while ((presi<3) || (presi>7) || (presi>TotCerini));
  printf("\n");
  TotCerini = TotCerini - presi;
  printf("Restano %d cerini\n", TotCerini);

  if ((TotCerini > 0) && (TotCerini <= 3)) Vince = Gioca;
  else
    if (Gioca == Nome1 ) Gioca = Nome2 ;
    else Gioca = Nome1;

  if (TotCerini == 0) Vince = Gioca;
}
while (TotCerini > 3);
printf ("Ha vinto %c \n", Vince);
}
```

Esempio

Analisi e Specifica

Definizione del problema: Scrivere, producendo la necessaria documentazione, un programma che data una sequenza di 30 numeri interi, determini la sommatoria dei numeri con valore pari e la produttoria dei numeri con valore dispari. Inoltre si contino quanti numeri pari e dispari sono stati immessi.

Definizione dei dati del problema:

I: un numero della sequenza

Pi: nessuna

U: sommatoria numeri con valore pari; produttoria numeri con valore dispari; conteggio numeri pari; conteggio numeri dispari

Pu: nessuna

Esempio

Tabella delle variabili di ingresso

Nome variabile	Descrizione	Tipo
Num	numero della sequenza	INT

Tabella delle variabili di uscita

Nome variabile	Descrizione	Tipo
sompari	sommatoria numeri con valore pari	INT
proddisp	produttoria numeri con valore dispari	INT
contapari	conteggio numeri con valore pari	INT
contadisp	conteggio numeri con valore dispari	INT

Esempio

Progettazione

Descrizione del metodo di elaborazione:

- Si inizializzano a zero le variabili sompari, contapari e contadisp e la variabile proddisp a 1 .
- Con un ciclo ripetitivo (per 30 volte) :
 - si legge un numero della sequenza
 - si analizza il numero immesso applicando l'operatore modulo '%' (con modulo 2),
 - se il risultato dell'operazione modulo è zero il numero è pari (Vero)
 - si incrementa la variabile sompari del valore immesso e si incrementa di 1 il valore di contapari,
 - altrimenti
 - si aggiorna il valore di proddisp moltiplicandolo per il valore immesso.
- Alla fine del ciclo si calcola il valore di contadisp quale differenza tra 30 e contapari.
- Si stampano i valori delle variabili di uscita.

Esempio - il programma C

```
#include<stdio.h>
main()
{
    int i, num, sompari, proddisp, contapari, contadisp;
    sompari=0;
    proddisp=1; contapari = 0; contadisp = 0;
    for (i=1; i<=30; i++)
    { printf(" dammi il %d numero della sequenza\n", i);
      scanf("%d", &num);
      if ((num%2)==0)
          {sompari = sompari + num;
           contapari++;}
      else proddisp=proddisp*num;
    }
    contadisp= 30 - contapari;
    printf("Immessi %d numeri pari \n Somma numeri pari =%d\n",
    contapari,sompari);
    If (contadisp == 0) printf("Nessun numero dispari immesso)
    else
    printf("Immessi %d numeri dispari \n Produttoria numeri dispari =
    %d\n", contadisp,proddisp);
    }
```