

/\*

Programma che:

Dati A e N (A reale &gt;0 e N intero &gt;=0) comunica il valore di A elevato alla N.

Variabili di Input: N di tipo intero, A di tipo reale

Variabili di Output: Pot di tipo reale

Variabili di Lavoro: i (contatore) di tipo intero

\*/

```
#include <iostream>
using namespace std;
```

```
//-----
```

```
int main ( )
```

```
{ // --- Dichiarazione delle variabili
```

```
int N ;
```

```
float A, Pot;
```

```
// --- Acquisizione del valore A intero positivo
```

```
do {cout << "CALCOLO di una Potenza \nInserisci il valore della base (>0) : ";
```

```
cin >> A;
```

```
}while (A<=0);
```

```
// --- Acquisizione del valore N intero non negativo
```

```
do {cout << "Inserisci il valore dell'esponente (>=0): ";
```

```
cin >> N;
```

```
}while (N<0);
```

```
// --- calcola base elevata a esponente moltiplicando A per N volte
```

```
Pot = 1; // Inizializza la variabile accumulatore Pot a 1
```

```
for(int i = 1; i <= N; i++) // Inizializza il contatore i a 1 e ripete N volte
```

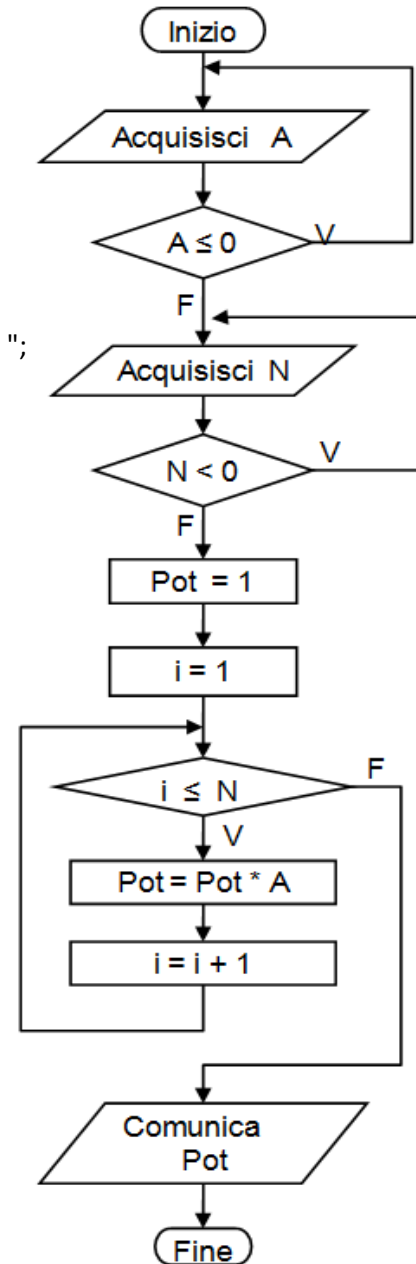
```
{Pot = Pot * A; // moltiplica per A N volte: 1*A*A*A....*A
```

```
}
```

```
cout << "\nLa potenza: " << A << "^" << N << " = " << Pot << "\n";
```

```
system ("pause");
```

```
}
```



/\*

Programma che:

dato N intero positivo comunica la prima potenza di 2 superiore a N.

Variabili di Input: **N** di tipo interoVariabili di Output: **Pot** di tipo intero, **Esponente** (contatore) di tipo intero

Variabili di Lavoro: nessuna

\*/

```
#include <iostream>
using namespace std;
```

```
//-----
```

```
int main ()
```

```
{ // --- Dichiarazione delle variabili
```

```
int N, Pot, Esponente;
```

```
// --- Acquisizione del valore N intero positivo o zero
```

```
do {cout <<"Inserisci numero intero (>0): ";
```

```
cin >> N;
```

```
}while (N <=0);
```

```
// --- Inizializza Pot e Esponente
```

```
Pot = 1;
```

```
Esponente = 0;
```

```
// --- Ciclo che calcola le potenze di 2
```

```
while (Pot <= N)
```

```
{Pot = Pot * 2; // P alla fine conterrà la potenza cercata
```

```
Esponente++;
```

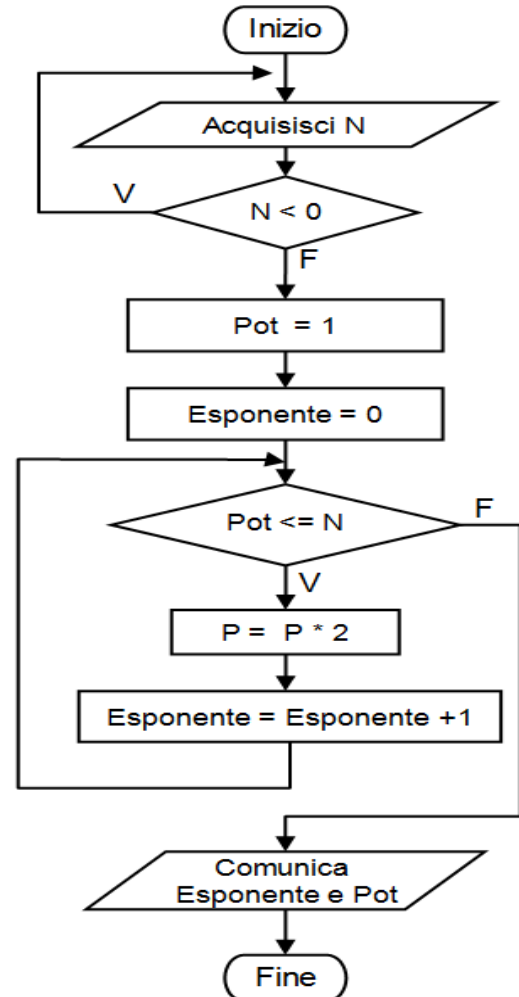
```
} // fine ciclo while
```

```
// --- Ha trovato la prima potenza di 2 superiore a N
```

```
cout << "La prima potenza di 2 superiore a "<< N <<" e' 2^"<< Esponente << "="<<Pot<<endl;
```

```
system ("pause");
```

```
}
```



```

/*
Programma che:
  dati due numeri interi non negativi, calcola il loro prodotto supponendo
  che l'esecutore possa eseguire solo somme e sottrazioni.

```

```

Variabili di Input: A e B di tipo intero
Variabili di Output: P di tipo intero
Variabili di Lavoro: k (contatore) di tipo intero
*/

```

```

#include <iostream>
using namespace std;

```

```

//-----
int main ( )
{ // --- Dichiarazione delle variabili
  int A, B, P ;

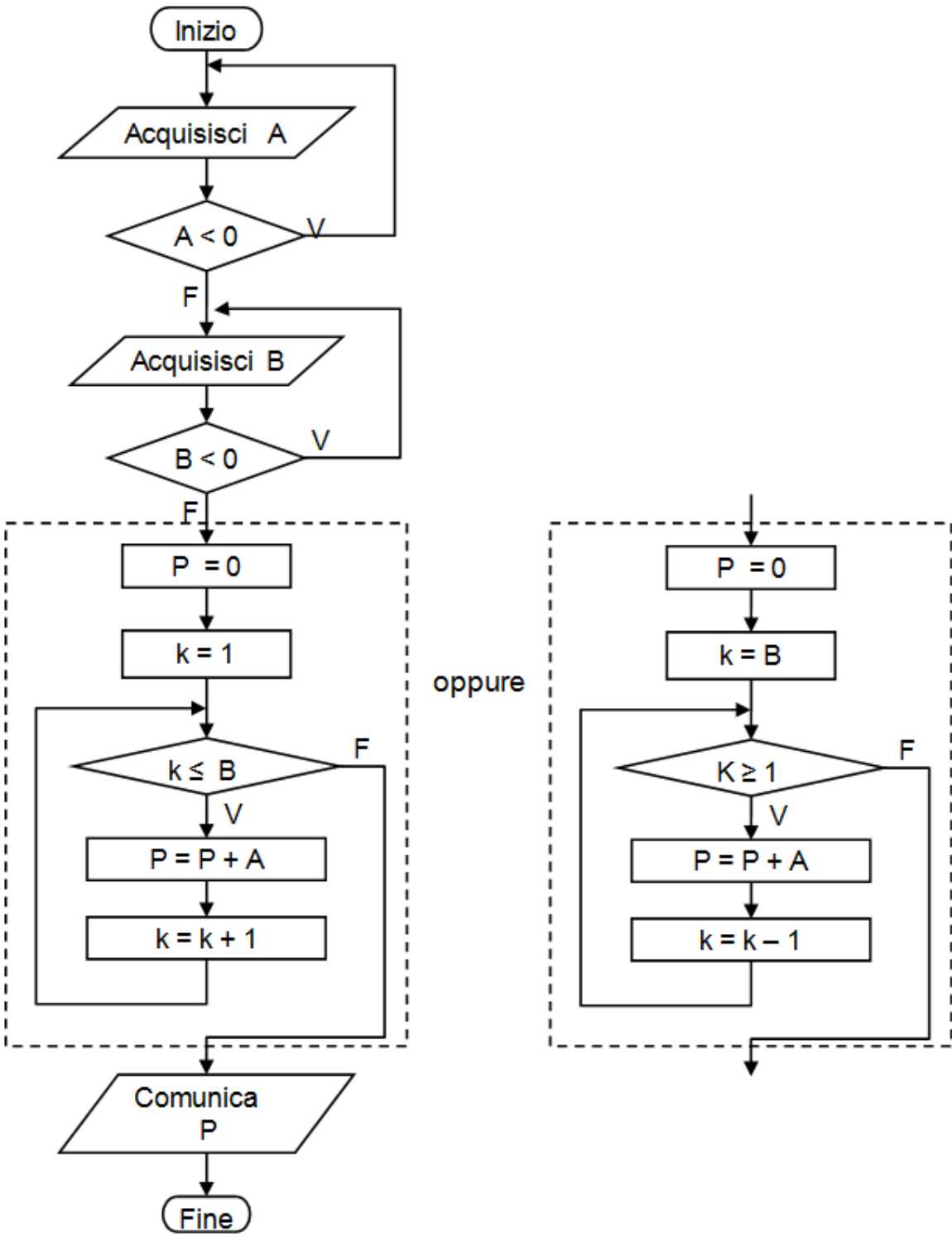
  // --- Acquisizione del valore A intero non negativo
  do {cout << "Inserisci primo valore(>=0) : ";
    cin >> A;
  }while (A<0);
  // --- Acquisizione del valore B intero non negativo
  do {cout << "Inserisci secondo valore (>=0): ";
    cin >> B;
  }while (B<0);

  // --- calcola il prodotto sommando A+A+A+....+A B volte
  P = 0; // Inizializza la variabile accumulatore P a 0
  for (int k = 1; k <= B; k++) // inizializza il contatore k a 1 e ripete per B volte
    // incrementando k ogni volta (quando k>B termina il ciclo)
    {P = P + A; // Aggiunge A B volte: 0+A+A+A+....+A
    }

  // -----In alternativa al precedente -----
  // --- calcola il prodotto sommando A+A+A+....+A B volte
  //P = 0; // Inizializza la variabile accumulatore P a 0
  //for (int k = B; k >=1; k--) // inizializza il contatore k a B e ripete per B volte
  // decrementando k ogni volta (quando k<1 termina il ciclo)
  // {P = P + A; // Aggiunge A B volte: 0+A+A+A+....+A
  // }
  // -----

  cout << "\nIl prodotto: " << A << "*" << B << " = " << P << "\n";
  system ("pause");
}

```



/\*

Programma che:  
 calcola il prodotto di una serie di numeri interi positivi inseriti da tastiera,  
 la somma di quelli dispari  
 e la media di quelli pari.  
 L'inserimento di 0 indica che la serie è terminata.

Variabili di Input: **N** di tipo intero

Variabili di Output: **P** di tipo intero, **SommaDisp** (totalizzatore) di tipo intero, **MediaPari** di tipo reale

Variabili di Lavoro: **SommaPari** (totalizzatore) di tipo intero, **ContaPari** (contatore) di tipo intero

\*/

```
#include <iostream>
using namespace std;
```

```
//-----
```

```
int main ( )
```

```
{ // --- Dichiarazione delle variabili
```

```
int N, P=1, SommaDisp=0, SommaPari=0, ContaPari=0;
float MediaPari;
```

```
// --- Acquisizione del valore N intero positivo o zero
```

```
do {cout <<"Inserisci numero intero (>=0, 0=Fine): ";
    cin >> N;
    }while (N <0);
```

```
// --- Inizio Ciclo while per chiedere valori interi positivi fino a quando non si inserisce 0
```

```
while (N != 0)
{ P = P * N; // P alla fine conterrà il prodotto di tutti i numeri inseriti
  // --- controlla se il numero è dispari
  if (N%2 == 1)
    {SommaDisp = SommaDisp + N; // Somma i numeri dispari
    }
  else
    {SommaPari = SommaPari + N; // Somma i numeri pari per poi fare la media
    ContaPari++; // Incrementa il contatore dei numeri pari
    }
}
```

```
// --- Si deve ripetere l'acquisizione del valore N prima di tornare su al while
```

```
do {cout <<"Inserisci numero intero (>=0, 0=Fine): ";
    cin >> N;
    }while (N <0);
```

```
} // fine ciclo while
```

```
// --- Comunica il prodotto di tutti i numeri e la somma dei numeri dispari
```

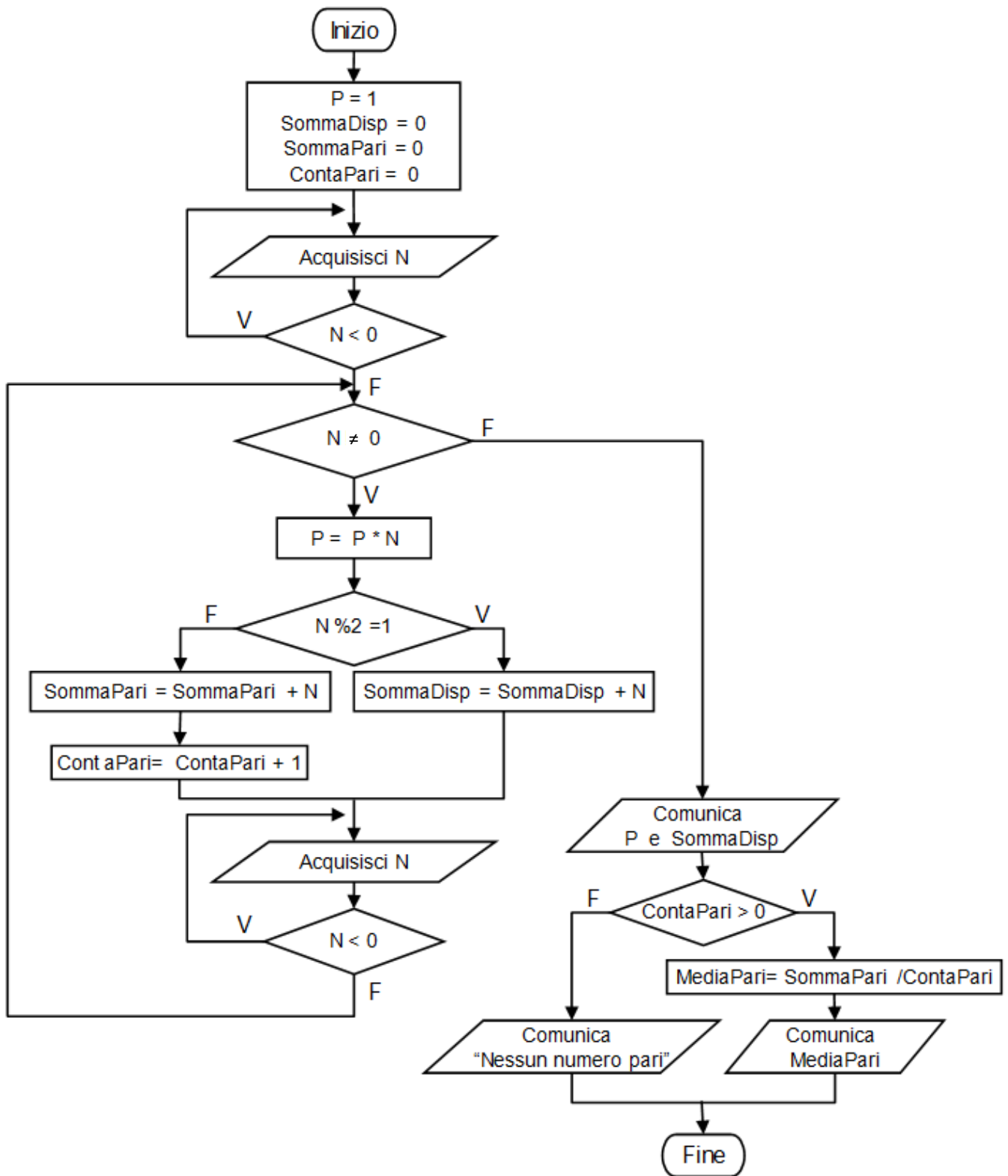
```
cout << "\nIl prodotto dei valori inseriti e' : "<< P << endl;
cout << "\nLa somma dei valori dispari e' : "<< SommaDisp << endl;
```

```
// --- Calcola la media dei pari dividendo la SommaPari per il numero ContaPari,
```

```
// ma prima di fare la divisione dobbiamo controllare che siano stati inseriti numeri pari  

// (dobbiamo evitare di dividere per 0)
```

```
if (ContaPari > 0)
{ MediaPari = (float) SommaPari / ContaPari;
  cout << "\nLa media dei valori pari e' : "<< MediaPari << endl;
}
else
{ cout << "\nNon e' stato inserito nessun valore pari" << endl;
}
system ("pause");
}
```



/\*

Programma che:

Chiede ripetutamente valori interi positivi fino a quando il loro prodotto non è maggiore di 1000.  
Comunicare quindi il prodotto finale, il numero di valori inseriti e la media dei numeri divisibili per 5.

Variabili di Input: **N** di tipo interoVariabili di Output: **P** di tipo intero, **Cont** (contatore) di tipo intero, **MediaDiv5** di tipo realeVariabili di Lavoro: **SommaDiv5** (totalizzatore) di tipo intero, **ContaDiv5** (contatore) di tipo intero

\*/

```
#include <iostream>
using namespace std;
```

//-----

**int main ( )****{ // --- Dichiarazione delle variabili**

```
int N, P=1, Cont=0, SommaDiv5=0, ContaDiv5=0;
float MediaDiv5;
```

**// --- Inizio Ciclo while per chiedere valori interi positivi****// fino a quando il prodotto non supera 1000**

while (P &lt;= 1000)

**{ // --- Acquisizione del valore N intero positivo**

do { cout &lt;&lt; "Inserisci numero intero: ";

cin &gt;&gt; N;

}while (N &lt;= 0);

Cont++; // Incrementa il contatore dei valori inseriti

P = P \* N; // Moltiplica P per il nuovo valore N acquisito

**// --- Controlla se il numero è divisibile per 5**

if (N%5 == 0)

{ SommaDiv5 = SommaDiv5 + N;

ContaDiv5++;

}

**} // --- Fine ciclo while****// --- Comunica il prodotto e quanti numeri sono stati inseriti**

cout &lt;&lt; "\nLa prodotto dei valori inseriti e' : "&lt;&lt; P &lt;&lt; endl;

cout &lt;&lt; "\nIl numero di valori inseriti e' : "&lt;&lt; Cont &lt;&lt; endl;

**// --- Calcola la media dei numeri divisibili per 5 dividendo la somma per il numero di valori****// ma prima di fare la divisione dobbiamo controllare che siano stati inseriti numeri divisibili per 5****// (dobbiamo evitare di dividere per 0)**

if (ContaDiv5 &gt; 0)

{ MediaDiv5 = (float) SommaDiv5 / ContaDiv5;

cout &lt;&lt; "\nLa media dei valori divisibili per 5 e' : "&lt;&lt; MediaDiv5 &lt;&lt; endl;

}

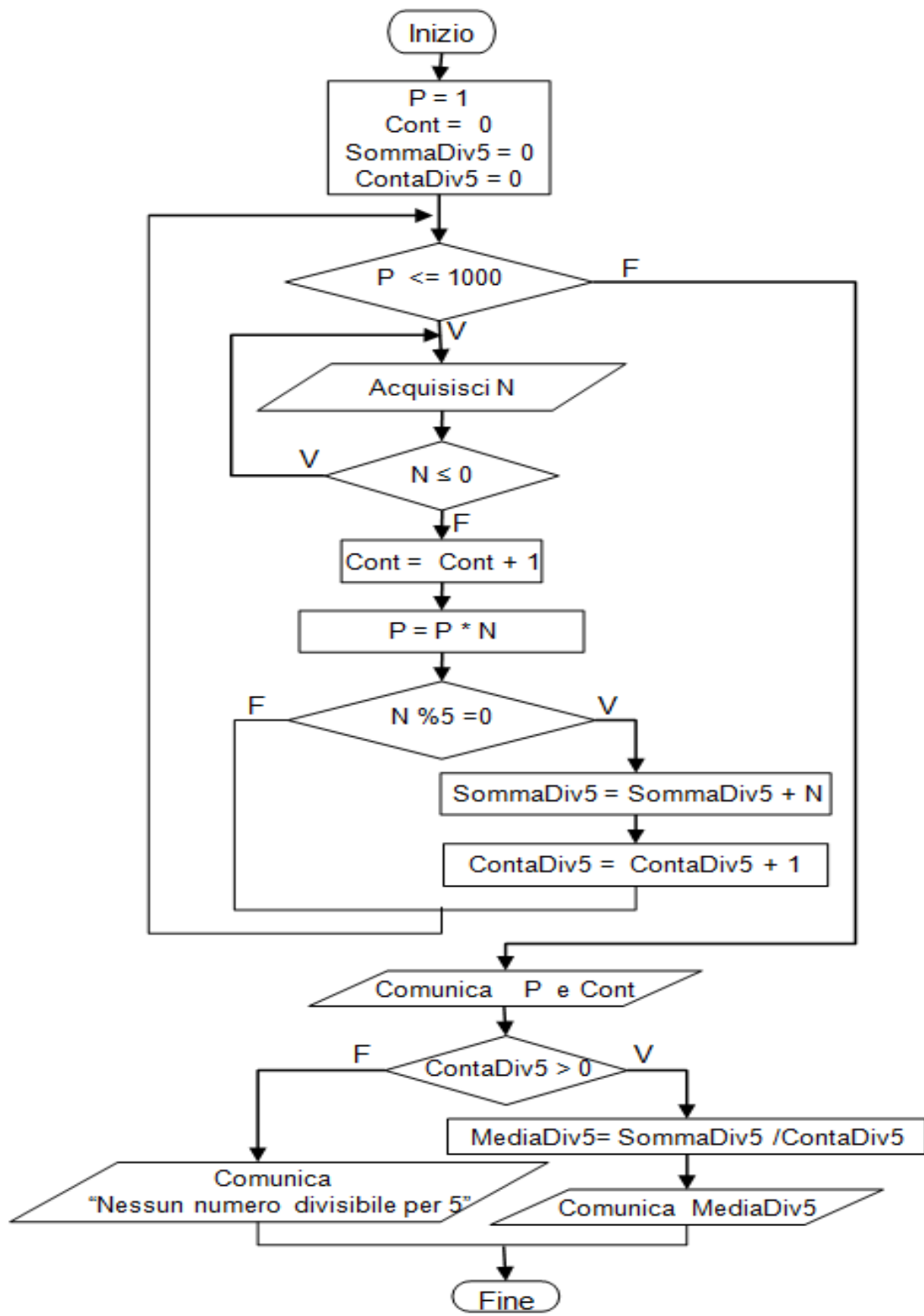
else

{ cout &lt;&lt; "\nNon e' stato inserito nessun valore divisibile per 5" &lt;&lt; endl;

}

system ("pause");

}





/\*

Programma che:

dati due numeri interi A e B ( $A \geq 0$  e  $B > 0$ ), calcolare il quoziente e il resto della divisione intera tra il primo e il secondo, supponendo che l'esecutore possa eseguire solo somme e sottrazioni.

Variabili di Input: **A** e **B** di tipo intero

Variabili di Output: **Quoz** di tipo intero, **Resto** di tipo intero

Variabili di Lavoro: nessuna

\*/

```
#include <iostream>
using namespace std;
```

```
//-----
```

```
int main ( )
```

```
{ // --- Dichiarazione delle variabili
```

```
int A, B, Quoz, Resto ;
```

```
// --- Acquisizione del valore A intero non negativo
```

```
do {cout << "Inserisci primo valore(>=0) : ";
```

```
cin >> A;
```

```
}while (A<0);
```

```
// --- Acquisizione del valore B intero positivo
```

```
do {cout << "Inserisci secondo valore (>0): ";
```

```
cin >> B;
```

```
}while (B<=0);
```

```
// --- Si pone il quoziente = 0 e il resto = A
```

```
Quoz = 0;
```

```
Resto = A;
```

```
// --- poi, inizio ciclo while finché il Resto >= B
```

```
while (Resto >= B)
```

```
{ Resto = Resto - B ; // si sottrae il valore B
```

```
Quoz++; // e si incrementa di 1 il Quoziente
```

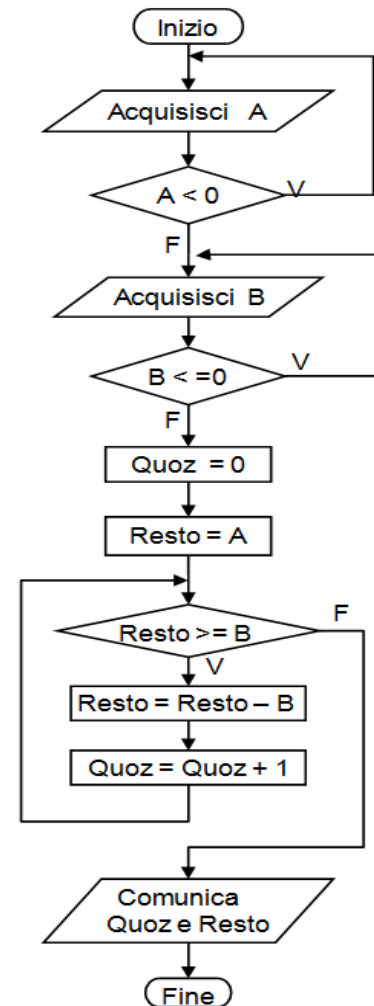
```
} // fine ciclo while
```

```
// --- Comunica il quoziente e il resto
```

```
cout << "\nIl Quoziente di: " << A << "/" << B << " = " << Quoz << " con Resto = " << Resto << endl;
```

```
system ("pause");
```

```
}
```



/\*

Programma che risolve il seguente problema:

"Un'associazione raccoglie delle offerte da destinare per l' 85% ad aiuti umanitari e il rimanente per le spese di gestione dell'associazione stessa. Vengono inseriti gli importi delle singole offerte e per convenzione l'inserimento di 0 indica che le offerte sono terminate.

A fine inserimento devono essere comunicati: l'importo totale delle offerte, l'importo da destinare agli aiuti umanitari e il numero di offerte ricevute. "

Variabili di Input: ImportoOff di tipo reale

Variabili di Output: ImpTot di tipo reale, ImpAiuti di tipo reale, NumOff (contatore) di tipo intero

Variabili di Lavoro: nessuna

\*/

```
#include <iostream>
using namespace std;
```

```
//-----
```

```
int main ( )
```

```
{ // --- Dichiarazione delle variabili
```

```
int NumOff = 0;
float ImportoOff, ImpTot=0, ImpAiuti ;
```

```
// --- Acquisizione del valore della prima offerta o zero
```

```
do {cout <<"Inserisci Importo offerta (0=Fine): ";
cin >> ImportoOff;
}while (ImportoOff <0);
```

```
// --- Inizio ciclo while per chiedere i valori degli importi
// delle offerte fino a quando non si inserisce 0
```

```
while (ImportoOff != 0)
{ // Somma all'importo totale l'importo dell'offerta
ImpTot = ImpTot + ImportoOff;
// Incrementa il contatore del numero delle offerte ricevute
NumOff++;
```

```
// --- Si deve ripetere l'acquisizione dell'importo
// della nuova offerta prima di tornare su al while
```

```
do {cout <<"Inserisci Importo offerta (0=Fine): ";
cin >> ImportoOff;
}while (ImportoOff <0);
```

```
} // fine ciclo while
```

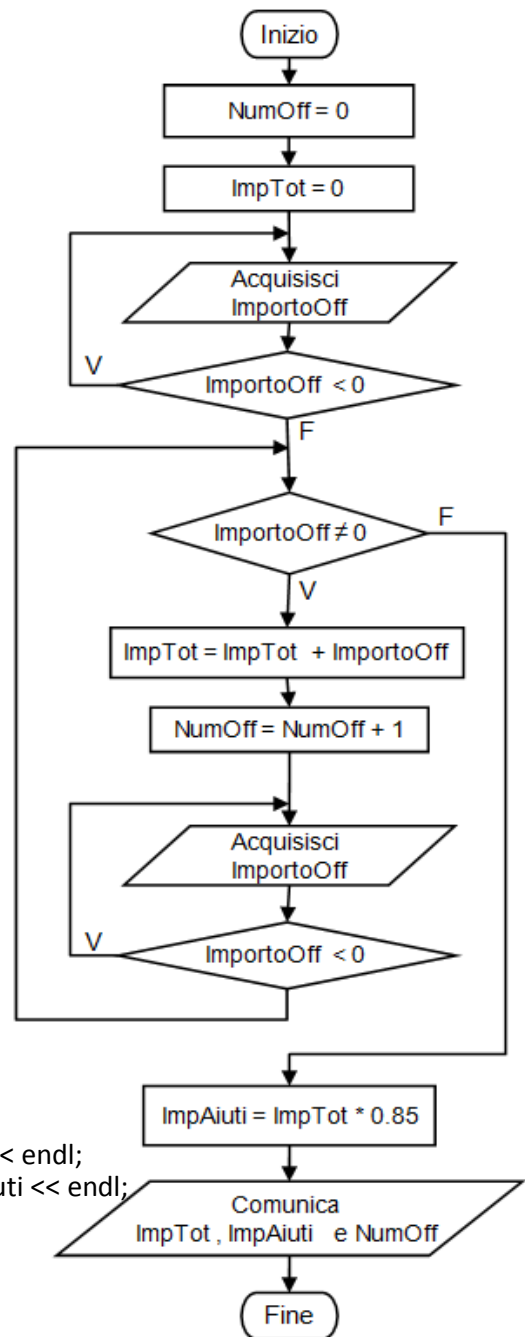
```
// --- Calcola l'importo destinato agli aiuti umanitari
```

```
ImpAiuti= ImpTot * 0.85 ;
```

```
// --- Comunica i dati finali
```

```
cout << "\nL'importo totale delle offerte ricevute e' : "<< ImpTot << endl;
cout << "La somma da destinare agli aiuti umanitari e' : "<< ImpAiuti << endl;
cout << "Il numero delle offerte ricevute e' : "<< NumOff << endl;
system ("pause");
```

```
}
```



Dato il diagramma a blocchi a lato:

- Scrivere il corrispondente programma completo in linguaggio C++.
- Creare la tabella di traccia per  $N = 9$ , indicando, poi, il risultato che viene visualizzato alla fine.
- Scrivere il "testo" del problema risolto dal diagramma a blocchi

a)

```

/*
Variabili di Input: N di tipo intero
Variabili di Output: P (totalizzatore) di tipo intero
Variabili di Lavoro: K di tipo intero
*/

#include <iostream>
using namespace std;

//-----
int main ( )
{ // --- Dichiarazione delle variabili
  int N, K, P=0;

  // --- Acquisizione del valore N intero positivo o zero
  do {cout <<"Inserisci numero intero (>0): ";
      cin >> N;
    }while (N <=0);

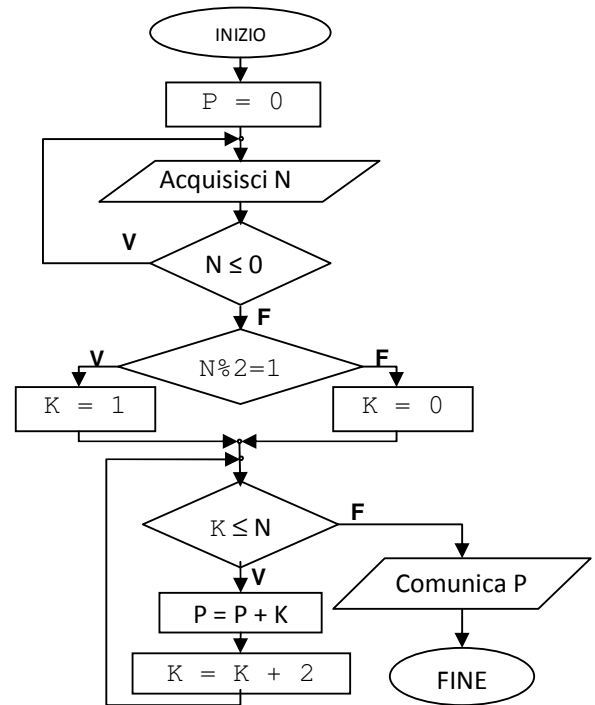
  // Controlla se N è dispari o pari
  if (N%2 == 1)
    { K = 1; // inizializza K a 1
    }
  else
    { K = 0; // inizializza K a 0
    }

  // --- Ciclo while finché K è <=N
  while (K <= N)
    { P = P + K; // P contiene la somma dei numeri
      K = K + 2; // prossimo numero dispari o pari
                // a seconda del valore iniziale di K
    } // fine ciclo while

  // --- In alternativa al ciclo while
  // for ( ; K <= N ; K = K+2 )
  // { P = P + K;
  // } // fine ciclo for

  // --- Comunica la somma dei numeri
  cout << "\nP = " << P << endl;
  system ("pause");
}

```



b)

Tabella di traccia

Passo	N	P	K
1		0	
2	9		
3			1
4		1	3
5		4	5
6		9	7
7		16	9
8		25	11

Viene visualizzato:

**P = 25**

c) Testo del problema

Programma che:

dato N intero positivo

- se N è dispari comunica la somma di tutti i numeri dispari da 1 a N
- se N è pari comunica la somma di tutti i numeri pari da 2 a N