



Esercitazione IF / SWITCH / WHILE / FOR

Informatica B, AA 2017/2018

Luca Cassano

4 Ottobre 2017

luca.cassano@polimi.it



Riepilogo di Teoria





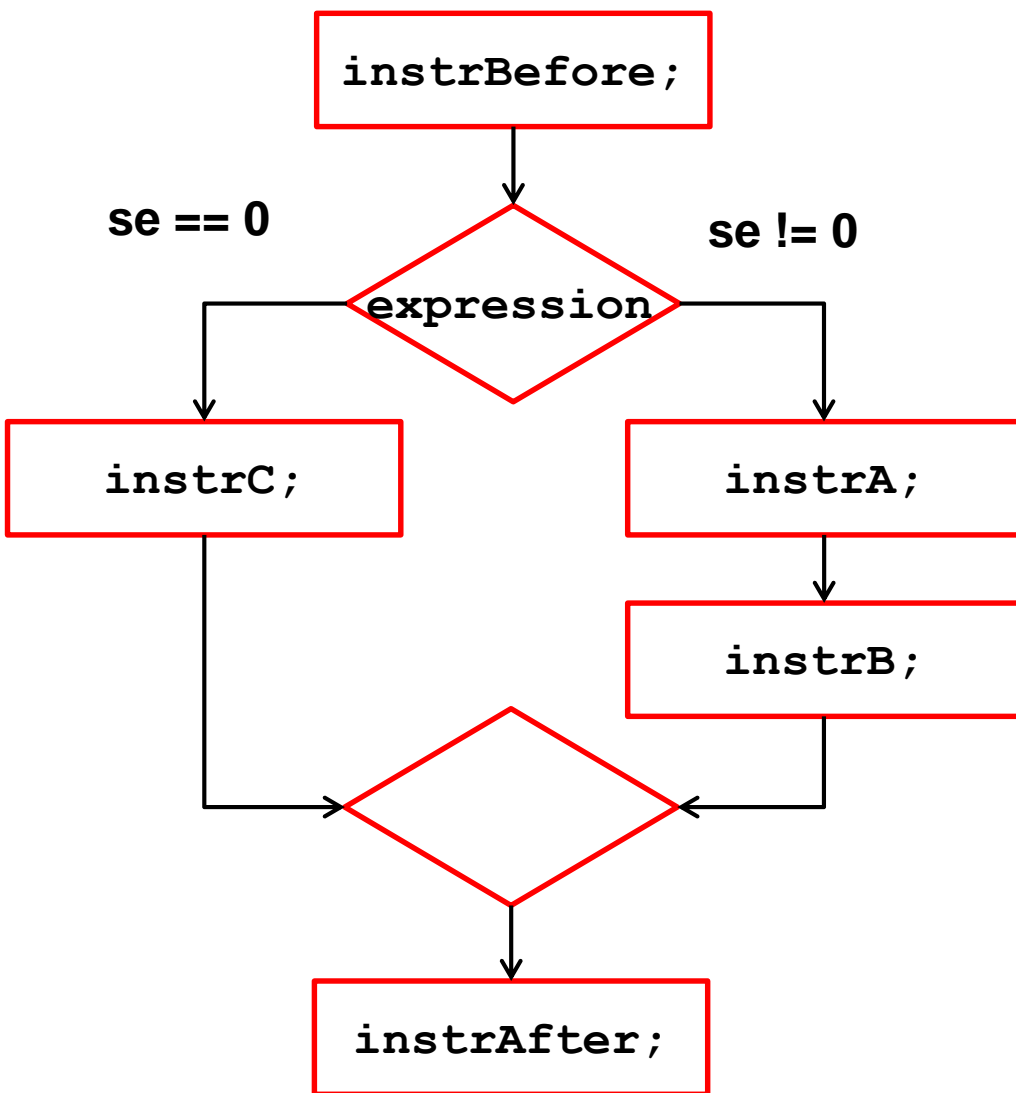
Riepilogo di Teoria

Finora abbiamo visto i seguenti costrutti del linguaggio C:

- **IF - ELSE**
- **IF - ELSE IF - ELSE**
- **SWITCH**
- **WHILE**
- **DO - WHILE**
- **FOR**



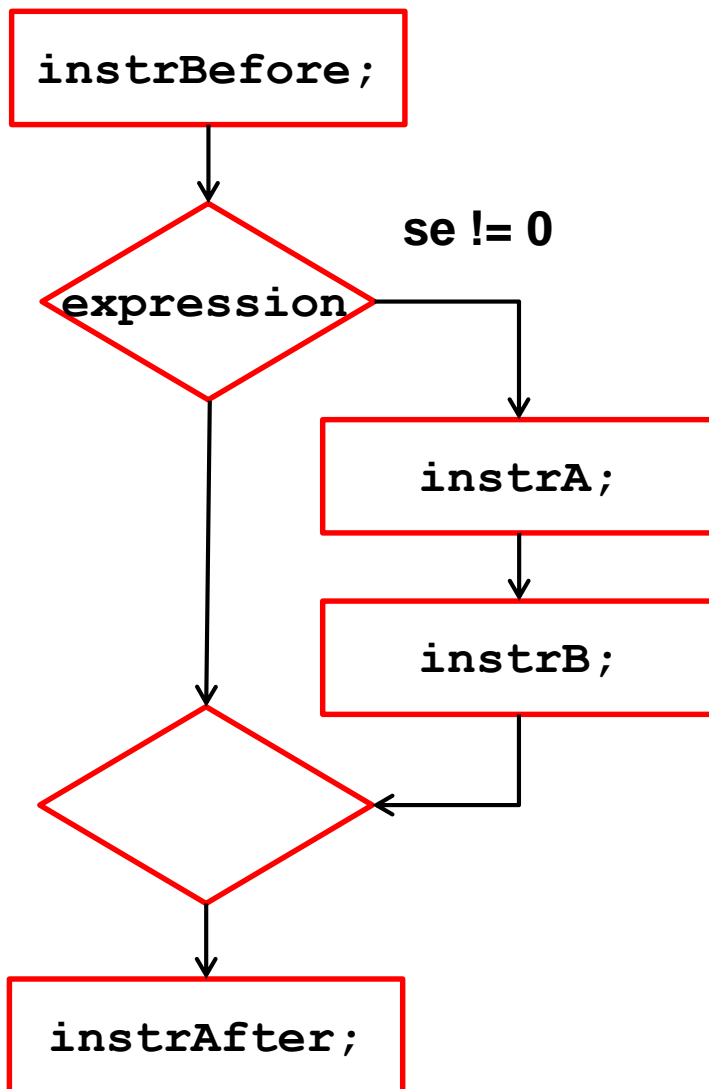
if



```
instrBefore;  
if (expression)  
{  
    instrA;  
    instrB;  
}  
else {  
    instrC;  
}
```



if

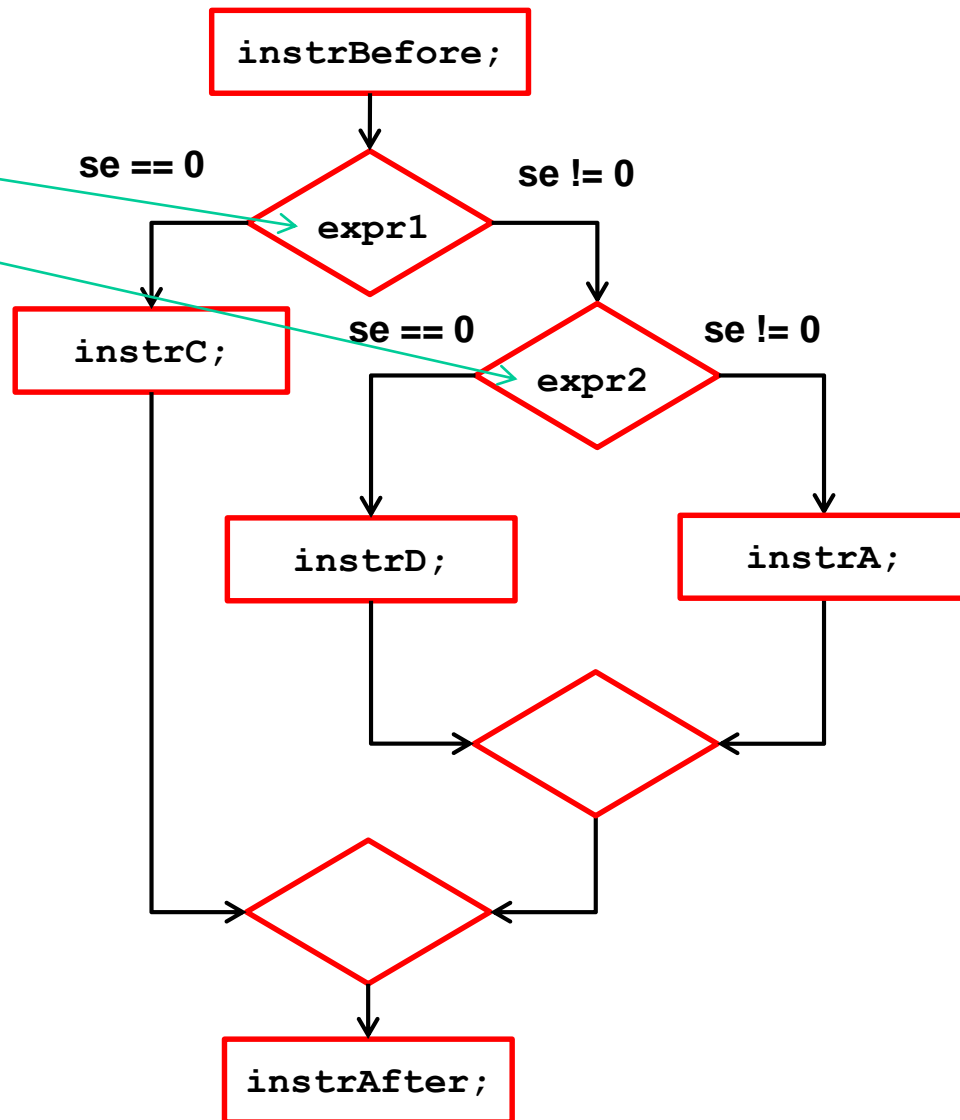


```
instrBefore;  
if (expression)  
{  
    instrA;  
    instrB;  
}  
instrAfter;
```



if Annidati

```
instrBefore;  
if (expr1) {  
  if (expr2) {  
    instrA;  
  }  
  else {  
    instrD;  
  }  
} else {  
  instrC;  
}  
instrAfter;
```





if - else if - else

Nel caso le condizioni da verificare siano più di una è possibile specificare catene di **if**

```
if (expr1)
    instr1;
else if (expr2)
    instr2;
else if (expr3)
    instr3;
else
    instr4;
```



```
if (expr1)
    instr1;
else {
    if (expr2)
        instr2;
    else {
        if (expr3)
            instr3;
        else
            instr4;
    }
}
```



switch-case

- **switch, case, default** keywords
- **int_expr** espressione a valori integral (char o int)
- **constant-expr1** numero o carattere
- **default** opzionale

```
switch (int_expr) {  
    case constant-expr1:  
        statement1  
    case constant-expr2:  
        statement2  
        ...  
    case constant-exprN:  
        statementN  
    default : statement  
}
```




switch-case

- **NB:** `int_expr` può contenere variabili
- **NB:** `constant-expr1` non può contenere una variabile,
- **NB:** a differenza di `if`, `while` e `for`,
 - `int_expr` non è un'espressione booleana

```
switch (int_expr) {  
    case constant-expr1:  
        statement1  
    case constant-expr2:  
        statement2  
        ...  
    case constant-exprN:  
        statementN  
    default : statement  
}
```



switch-case

1. Viene valutata `int_expr` (eventualmente convertita)
2. Si controlla se `int_expr` è **uguale** a `constant-expr1`
3. Se sono **uguali** eseguo **statement1**, ed in **cascata**, **tutti gli statement** dei **case** seguenti (senza verifiche, incluso lo **statement** di **default**)
4. Altrimenti controllo se **expression** è **uguale** a `constant-expr2` ...
5. Eseguo lo **statement** di **default** [se presente]

```
switch (int_expr) {  
    case constant-expr1:  
        statement1  
    case constant-expr2:  
        statement2  
    ...  
    case constant-exprN:  
        statementN  
    default : statement  
}
```



switch-case

Esempio di utilizzo di **switch**

```
.....  
scanf ("%c" , &a) ;  
switch (a) {  
    case 'A' : nA++ ;  
    case 'E' : nE++ ;  
    case 'O' : nO++ ;  
    default : nCons++ ;  
  
}
```

- ```
.....
```
- Se `a == 'A'` , verranno incrementate `nA` , `nE` , `nO` , `nCons` ;
  - Se `a == 'E'` , verranno incrementate `nE` , `nO` , `nCons` ;
  - Se `a == 'O'` , verranno incrementate `nO` , `nCons` ;
  - Se `a == 'K'` , verranno incrementa `nCons` ;



## switch-case

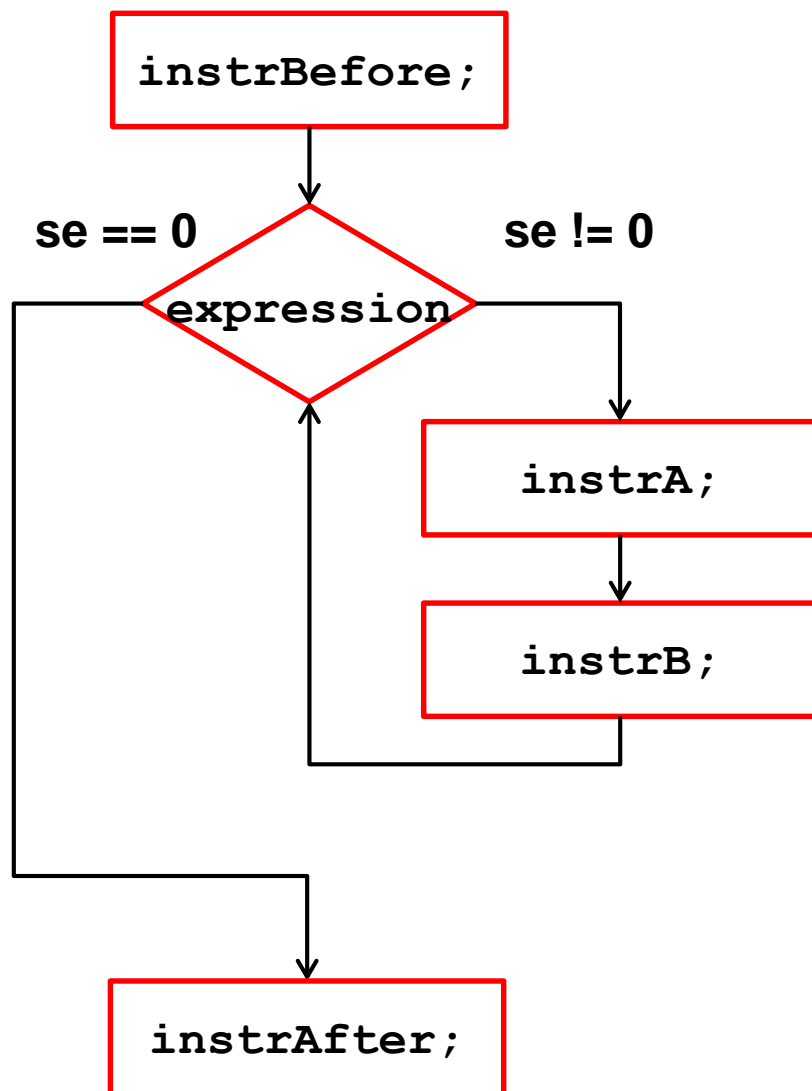
Per evitare l'**esecuzione in cascata** alla prima corrispondenza trovata, occorre inserire negli statements opportuni la keyword **break**

```
scanf ("%c" , &a) ;
switch (a) {
 case 'A' : nA++; break ;
 case 'E' : nE++; break ;
 case 'O' : nO++; break ;
 default : nCons++;
}
```

- Se `a == 'A'` , verrà incrementata `nA` ;
- Se `a == 'E'` , verrà incrementata `nE` ;
- Se `a == 'O'` , verrà incrementata `nO` ;
- Se `a == 'K'` , verrà incrementa `nCons` ;



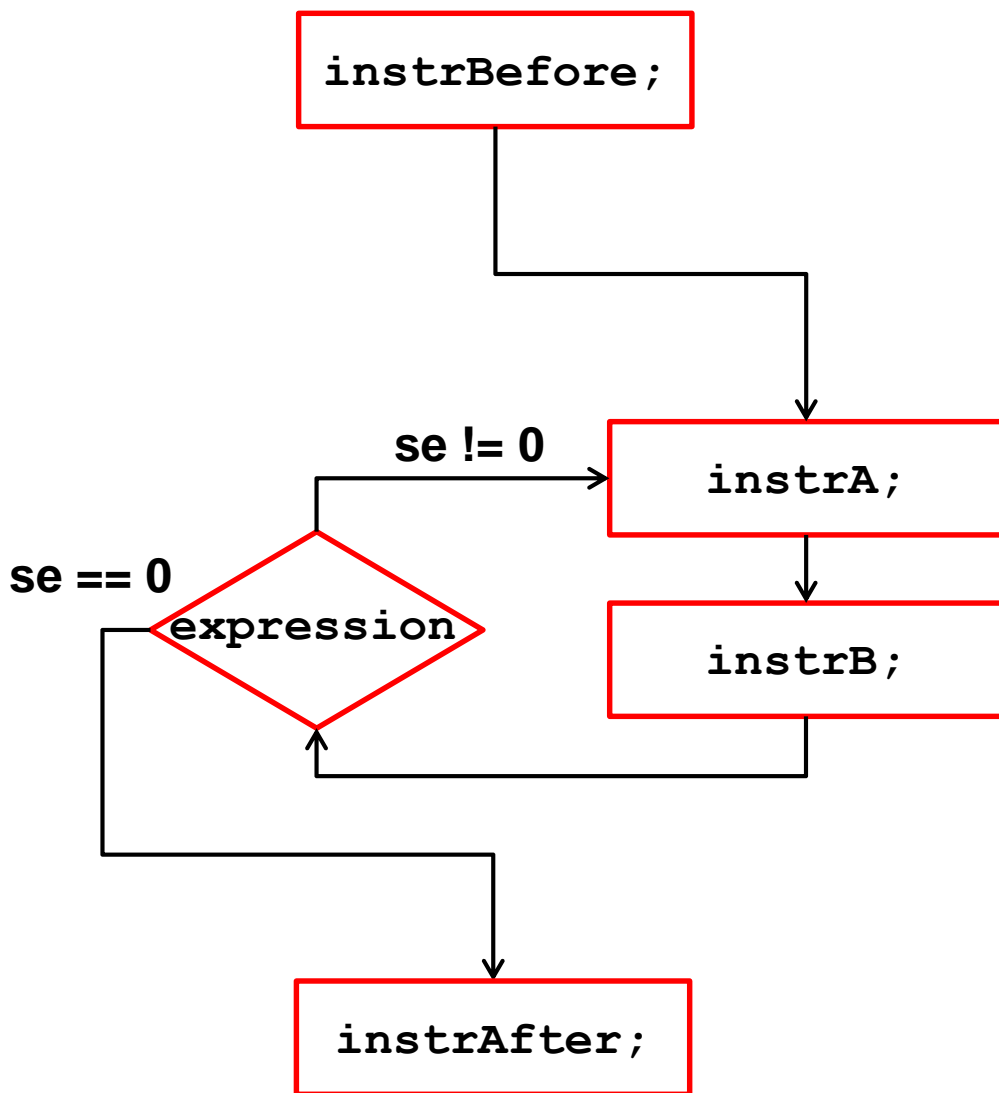
# while



```
instrBefore;
while (expression) {
 instrA;
 instrB;
}
instrAfter;
```



# do-while



```
instrBefore;
```

```
do {
 instrA;
 instrB;
}
```

```
while
(expression);
```

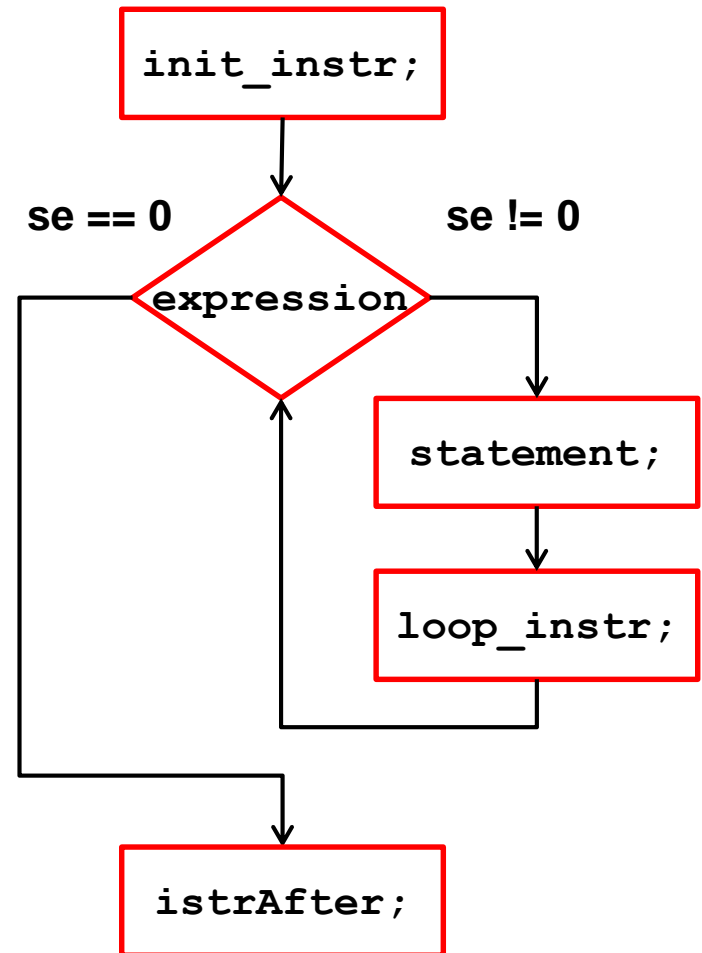
```
instrAfter;
```



# for

```
for (init_instr; expression; loop_instr)
 statement
```

1. Esegue `init_instr`
2. Valuta `expression`
3. Se vera, esegue `statement`, se falsa, termina il loop.
4. Al termine di `statement` esegue `loop_instr`
5. Valuta `expression`





# Let's rock!







## Massimo Comun Divisore

- Scrivere un programma che richieda all'utente di inserire due interi e ne calcola il massimo comune divisore.
- Suggerimento:  $x$  è divisore di  $y$  se e solo se  $y \% x == 0$



# Massimo Comun Divisore

```
#include <stdio.h>
void main() {
```

```
}
```



# Massimo Comun Divisore

```
#include <stdio.h>
void main() {
 int a, b, maxValue, i = 1, MCD;

}
}
```



# Massimo Comun Divisore

```
#include <stdio.h>
void main() {
 int a, b, maxValue, i = 1, MCD;
 printf("\n\nInserire primo numero: ");
 scanf("%d", &a); scanf("%*c");
 printf("\n\nInserire secondo numero numero: ");
 scanf("%d", &b); scanf("%*c");

}
```



# Massimo Comun Divisore

```
#include <stdio.h>
void main() {
 int a, b, maxValue, i = 1, MCD;
 printf("\n\nInserire primo numero: ");
 scanf("%d", &a); scanf("%*c");
 printf("\n\nInserire secondo numero numero: ");
 scanf("%d", &b); scanf("%*c");
 if(a < b)
 maxValue = a;
 else
 maxValue = b;
}
```



# Massimo Comun Divisore

```
#include <stdio.h>
void main() {
 int a, b, maxValue, i = 1, MCD;
 printf("\n\nInserire primo numero: ");
 scanf("%d", &a); scanf("%*c");
 printf("\n\nInserire secondo numero numero: ");
 scanf("%d", &b); scanf("%*c");
 if(a < b)
 maxValue = a;
 else
 maxValue = b;
 while(i <= maxValue) {

 }
}
```



# Massimo Comun Divisore

```
#include <stdio.h>
void main() {
 int a, b, maxValue, i = 1, MCD;
 printf("\n\nInserire primo numero: ");
 scanf("%d", &a); scanf("%*c");
 printf("\n\nInserire secondo numero numero: ");
 scanf("%d", &b); scanf("%*c");
 if(a < b)
 maxValue = a;
 else
 maxValue = b;
 while(i <= maxValue) {
 if((a % i == 0) && (b % i == 0))
 MCD = i;
 i++;
 }
}
```



# Massimo Comun Divisore

```
#include <stdio.h>
void main() {
 int a, b, maxValue, i = 1, MCD;
 printf("\n\nInserire primo numero: ");
 scanf("%d", &a); scanf("%*c");
 printf("\n\nInserire secondo numero numero: ");
 scanf("%d", &b); scanf("%*c");
 if(a < b)
 maxValue = a;
 else
 maxValue = b;
 while(i <= maxValue) {
 if((a % i == 0) && (b % i == 0))
 MCD = i;
 i++;
 }
 printf("mcd = %d\n", MCD);
}
```





# Soluzione Alternativa





# Massimo Comun Divisore (Soluzione Alternativa)

```
#include <stdio.h>
void main() {
 int a, b, maxValue, i, MCD, found=0;
 printf("\n\nInserire primo numero: ");
 scanf("%d", &a); scanf("%*c");
 printf("\n\nInserire secondo numero numero: ");
 scanf("%d", &b); scanf("%*c");
 if(a < b)
 maxValue = a;
 else
 maxValue = b;

 printf("mcd = %d\n", MCD);
}
```



# Massimo Comun Divisore (Soluzione Alternativa)

```
#include <stdio.h>
void main() {
 int a, b, maxValue, i, MCD, found=0;
 printf("\n\nInserire primo numero: ");
 scanf("%d", &a); scanf("%*c");
 printf("\n\nInserire secondo numero numero: ");
 scanf("%d", &b); scanf("%*c");
 if(a < b)
 maxValue = a;
 else
 maxValue = b;
 i = maxValue;

 printf("mcd = %d\n", MCD);
}
```



# Massimo Comun Divisore (Soluzione Alternativa)

```
#include <stdio.h>
void main() {
 int a, b, maxValue, i, MCD, found=0;
 printf("\n\nInserire primo numero: ");
 scanf("%d", &a); scanf("%*c");
 printf("\n\nInserire secondo numero numero: ");
 scanf("%d", &b); scanf("%*c");
 if(a < b)
 maxValue = a;
 else
 maxValue = b;
 i = maxValue;
 while(i > 0 && found != 1) {

 }
 printf("mcd = %d\n", MCD);
}
```



# Massimo Comun Divisore (Soluzione Alternativa)

```
#include <stdio.h>
void main() {
 int a, b, maxValue, i, MCD, found=0;
 printf("\n\nInserire primo numero: ");
 scanf("%d", &a); scanf("%*c");
 printf("\n\nInserire secondo numero numero: ");
 scanf("%d", &b); scanf("%*c");
 if(a < b)
 maxValue = a;
 else
 maxValue = b;
 i = maxValue;
 while(i > 0 && found != 1) {
 if((a % i == 0) && (b % i == 0)) {
 MCD = i;
 }
 }
 printf("mcd = %d\n", MCD);
}
```



# Massimo Comun Divisore (Soluzione Alternativa)

```
#include <stdio.h>
void main() {
 int a, b, maxValue, i, MCD, found=0;
 printf("\n\nInserire primo numero: ");
 scanf("%d", &a); scanf("%*c");
 printf("\n\nInserire secondo numero numero: ");
 scanf("%d", &b); scanf("%*c");
 if(a < b)
 maxValue = a;
 else
 maxValue = b;
 i = maxValue;
 while(i > 0 && found != 1) {
 if((a % i == 0) && (b % i == 0)) {
 MCD = i;
 found = 1;
 }
 }
 printf("mcd = %d\n", MCD);
}
```



# Massimo Comun Divisore (Soluzione Alternativa)

```
#include <stdio.h>
void main() {
 int a, b, maxValue, i, MCD, found=0;
 printf("\n\nInserire primo numero: ");
 scanf("%d", &a); scanf("%*c");
 printf("\n\nInserire secondo numero numero: ");
 scanf("%d", &b); scanf("%*c");
 if(a < b)
 maxValue = a;
 else
 maxValue = b;
 i = maxValue;
 while(i > 0 && found != 1) {
 if((a % i == 0) && (b % i == 0)) {
 MCD = i;
 found = 1;
 }
 i--;
 }
 printf("mcd = %d\n", MCD);
}
```



# Tabelline Pitagoriche

- Scrivere un programma che stampa la tabella pitagorica
  - Modificarlo per stampare solo la parte triangolare alta/  
solo la parte triangolare bassa / solo la diagonale

## TABELLINE

| x  | 0 | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10  |
|----|---|----|----|----|----|----|----|----|----|----|-----|
| 0  | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   |
| 1  | 0 | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10  |
| 2  | 0 | 2  | 4  | 6  | 8  | 10 | 12 | 14 | 16 | 18 | 20  |
| 3  | 0 | 3  | 6  | 9  | 12 | 15 | 18 | 21 | 24 | 27 | 30  |
| 4  | 0 | 4  | 8  | 12 | 16 | 20 | 24 | 28 | 32 | 36 | 40  |
| 5  | 0 | 5  | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50  |
| 6  | 0 | 6  | 12 | 18 | 24 | 30 | 36 | 42 | 48 | 54 | 60  |
| 7  | 0 | 7  | 14 | 21 | 28 | 35 | 42 | 49 | 56 | 63 | 70  |
| 8  | 0 | 8  | 16 | 24 | 32 | 40 | 48 | 56 | 64 | 72 | 80  |
| 9  | 0 | 9  | 18 | 27 | 36 | 45 | 54 | 63 | 72 | 81 | 90  |
| 10 | 0 | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |







# Tabelline Pitagoriche

```
#include <stdio.h>
void main() {

}
}
```



# Tabelline Pitagoriche

```
#include <stdio.h>
void main() {
 int i, j;

}
```



# Tabelline Pitagoriche

```
#include <stdio.h>
void main() {
 int i, j;
 for(i = 0; i <= 10; ++i) {

 }
 }
}
```



# Tabelline Pitagoriche

```
#include <stdio.h>
void main() {
 int i, j;
 for(i = 0; i <= 10; ++i) {
 for(j = 0; j <= 10; ++j) {

 }

 }
}
```



# Tabelline Pitagoriche

```
#include <stdio.h>
void main() {
 int i, j;
 for(i = 0; i <= 10; ++i) {
 for(j = 0; j <= 10; ++j) {
 printf("%3d ", i*j);
 }
 }
}
```



# Tabelline Pitagoriche

```
#include <stdio.h>
void main() {
 int i, j;
 for(i = 0; i <= 10; ++i) {
 for(j = 0; j <= 10; ++j) {
 printf("%3d ", i*j);
 }
 printf("\n");
 }
}
```



# Tabelline Pitagoriche

```
#include <stdio.h>
void main() {
 int i, j;
 for(i = 0; i <= 10; ++i) {
 for(j = 0; j <= 10; ++j) {
 printf("%3d ", i*j);
 }
 printf("\n");
 }
}
```

```
Terminale File Modifica Visualizza Cerca Terminale Aiuto
luca@luca:~/Scrivania/Condivisa_con_Windows/Didattica/InformaticaB/2017-2018/Esercitazioni$ gcc lab03_es01.c -o lab03_es01
luca@luca:~/Scrivania/Condivisa_con_Windows/Didattica/InformaticaB/2017-2018/Esercitazioni$./lab03_es01
0 0 0 0 0 0 0 0 0 0
0 1 2 3 4 5 6 7 8 9 10
0 2 4 6 8 10 12 14 16 18 20
0 3 6 9 12 15 18 21 24 27 30
0 4 8 12 16 20 24 28 32 36 40
0 5 10 15 20 25 30 35 40 45 50
0 6 12 18 24 30 36 42 48 54 60
0 7 14 21 28 35 42 49 56 63 70
0 8 16 24 32 40 48 56 64 72 80
0 9 18 27 36 45 54 63 72 81 90
0 10 20 30 40 50 60 70 80 90 100
luca@luca:~/Scrivania/Condivisa_con_Windows/Didattica/InformaticaB/2017-2018/Esercitazioni$ gcc lab03_es01.c -o lab03_es01
luca@luca:~/Scrivania/Condivisa_con_Windows/Didattica/InformaticaB/2017-2018/Esercitazioni$./lab03_es01
0 0 0 0 0 0 0 0 0 0
0 1 2 3 4 5 6 7 8 9 10
0 2 4 6 8 10 12 14 16 18 20
0 3 6 9 12 15 18 21 24 27 30
0 4 8 12 16 20 24 28 32 36 40
0 5 10 15 20 25 30 35 40 45 50
0 6 12 18 24 30 36 42 48 54 60
0 7 14 21 28 35 42 49 56 63 70
0 8 16 24 32 40 48 56 64 72 80
0 9 18 27 36 45 54 63 72 81 90
0 10 20 30 40 50 60 70 80 90 100
luca@luca:~/Scrivania/Condivisa_con_Windows/Didattica/InformaticaB/2017-2018/Esercitazioni$
```



# Tabelline Pitagoriche (Solo Diagonale Principale)

```
#include <stdio.h>
void main() {
 int i, j;
 for(i = 0; i <= 10; ++i) {
 for(j = 0; j <= 10; ++j) {
 if(i == j) {
 printf("%3d ", i*j);
 }
 else
 printf(" ");
 }
 printf("\n");
 }
}
```

```
Terminale File Modifica Visualizza Cerca Terminale Aiuto
luca@luca:~/Scrivania/Condivisa_con_Windows/Didattica/InformaticaB/2017-2018/Esercitazioni$./lab03_es01
0
 1
 4
 9
 16
 25
 36
 49
 64
 81
 100
luca@luca:~/Scrivania/Condivisa_con_Windows/Didattica/InformaticaB/2017-2018/Esercitazioni$
```





# Tabelline Pitagoriche (Solo Parte Triangola Alta)

```
#include <stdio.h>
void main() {
 int i, j;
 for(i = 0; i <= 10; ++i) {
 for(j = 0; j <= 10; ++j) {
 if(i <= j) {
 printf("%3d ", i*j);
 }
 else
 printf(" ");
 }
 printf("\n");
 }
}
```

```
Terminale File Modifica Visualizza Cerca Terminale Aiuto
luca@luca:~/Scrivania/Condivisa_con_Windows/Didattica/InformaticaB/2017-2018/Esercitazioni$./lab03_es01
 0 0 0 0 0 0 0 0 0 0
 1 2 3 4 5 6 7 8 9 10
 4 6 8 10 12 14 16 18 20
 9 12 15 18 21 24 27 30
16 20 24 28 32 36 40
25 30 35 40 45 50
36 42 48 54 60
49 56 63 70
64 72 80
81 90
100
luca@luca:~/Scrivania/Condivisa_con_Windows/Didattica/InformaticaB/2017-2018/Esercitazioni$
```



# Tabelline Pitagoriche (Solo Parte Triangola Bassa)

```
#include <stdio.h>
void main() {
 int i, j;
 for(i = 0; i <= 10; ++i) {
 for(j = 0; j <= 10; ++j) {
 if(i >= j) {
 printf("%3d ", i*j);
 }
 else
 printf(" ");
 }
 printf("\n");
 }
}
```

```
Terminale File Modifica Visualizza Cerca Terminale Aiuto
luca@luca:~/Scrivania/Condivisa_con_Windows/Didattica/InformaticaB/2017-2018/Esercitazioni$./lab03_es01
0
0 1
0 2 4
0 3 6 9
0 4 8 12 16
0 5 10 15 20 25
0 6 12 18 24 30 36
0 7 14 21 28 35 42 49
0 8 16 24 32 40 48 56 64
0 9 18 27 36 45 54 63 72 81
0 10 20 30 40 50 60 70 80 90 100
luca@luca:~/Scrivania/Condivisa_con_Windows/Didattica/InformaticaB/2017-2018/Esercitazioni$
```



# Calcolatrice

Scrivere un programma che opera come una calcolatrice: richiede due operandi ed un operatore `+` `-` `*` `/` `%` e restituisce il risultato a schermo



```
#include <stdio.h>
void main() {

}
```



```
#include <stdio.h>
void main() {
 int a, b;
 char op;

}
```



# Calcolatrice

```
#include <stdio.h>
void main() {
 int a, b;
 char op;
 do {
 }
 }
 while (op != 'x');
}
```



# Calcolatrice

```
#include <stdio.h>
void main() {
 int a, b;
 char op;
 do {
 printf("Inserisci a: ");
 scanf("%d", &a); scanf("%*c");
 printf("Inserisci b: ");
 scanf("%d", &b); scanf("%*c");
 printf("Specifica l'operazione da eseguire: ");
 scanf("%c", &op); scanf("%*c");

 }
 while (op != 'x');
}
```



```
.....
.....
do {

.....
.....
 switch(op) {
 case '+':
 case '-':
 case '*':
 case '/':
 case '%':
 case 'x':
 default:
 }
}
while(op != 'x');
}
```





```
.....
.....
do {

.....
.....
 switch(op) {
 case '+': printf("%d\n", a + b); break;
 case '-':
 case '*':
 case '/':
 case '%':
 case 'x':
 default:
 }
}
while(op != 'x');
}
```



# Calcolatrice

```
.....
.....
do {

.....
.....
 switch(op) {
 case '+': printf("%d\n", a + b); break;
 case '-': printf("%d\n", a - b); break;
 case '*': printf("%d\n", a * b); break;
 case '/': printf("%d\n", a / b); break;
 case '%': printf("%d\n", a % b); break;
 case 'x': printf("La calcolatrice sta per
chiudersi\n"); break
 default: printf("il comando specificato e'
errato\n");
 }
}
while(op != 'x');
}
```



## Calcolatrice (alternative)

```
do {
```

```
.....
.....
.....
.....
if (op == '+') {printf("%d\n", a + b);}
else if (op == '-') {printf("%d\n", a - b);}
else if (op == '*') {printf("%d\n", a * b);}
else if (op == '/') {printf("%d\n", a / b);}
else if (op == '%') {printf("%d\n", a % b);}
else if (op == 'x') {printf(«La calcolatrice sta
```

```
per chiudersi\n");}
```

```
errato\n");} else {printf("il comando specificato e'
```

```
}
```

```
}
```

```
while (op != 'x');
```

```
}
```



## Calcolo di potenze

Scrivere un programma che legga da tastiera un numero reale  $n$  ed un numero intero  $m$  e calcoli  $n^m$ .

Fare attenzione a tutti i casi particolari: esponente 0, 1, intero negativo.



# Calcolo di potenze

```
#include <stdio.h>
void main() {
 float n, aux;
 int m, i, pos_m;


```



# Calcolo di potenze

```
#include <stdio.h>
void main() {
 float n, aux;
 int m, i, pos_m;
 printf("Inserisci n\n");
 scanf("%f", &n); scanf("%*c");
 printf("Inserisci m\n");
 scanf("%d", &m); scanf("%*c");
 printf("%f elevato alla %d vale: ", n, m);


```



# Calcolo di potenze

```
.....
.....
if(m == 0)
 printf("1\n");
else if(m == 1)
 printf("%f\n", n);
else if(m == -1)
 printf("%f\n", 1.0/n);
.....
.....
```



## Calcolo di potenze

```
else {
 i = 2;
 aux = n;
 if(m > 0)
 pos_m = m;
 else
 pos_m = -m;
 while(i <= pos_m) {
 aux *= n;
 ++i;
 }
 if(m > 0)
 printf("%f\n", aux);
 else
 printf("%f\n", 1.0/aux);
}
}
```





## Quadrato di asterischi

Scrivere un programma che chieda all'utente un numero intero  $n$  e visualizzi a monitor un quadrato costituito di asterischi.

Ad esempio, se l'utente inserisce il numero 3, il programma visualizza quanto segue:

```
* * *
* * *
* * *
```



## Quadrato di asterischi

```
#include <stdio.h>
void main() {

}
}
```



## Quadrato di asterischi

```
#include <stdio.h>
void main() {
 int n, i, j;

}
}
```



## Quadrato di asterischi

```
#include <stdio.h>
void main() {
 int n, i, j;
 printf("lato del quadrato? ");
 scanf("%d", &n); scanf("%*c");

}
}
```



## Quadrato di asterischi

```
#include <stdio.h>
void main() {
 int n, i, j;
 printf("lato del quadrato? ");
 scanf("%d", &n); scanf("%*c");
 // gestione delle righe
 for(i=0; i<n; i++) {

 }
}
```



## Quadrato di asterischi

```
#include <stdio.h>
void main() {
 int n, i, j;
 printf("lato del quadrato? ");
 scanf("%d", &n); scanf("%*c");
 // gestione delle righe
 for(i=0; i<n; i++) {
 // gestione delle colonne
 for(j=0; j<n; j++) {

 }

 }
}
```



## Quadrato di asterischi

```
#include <stdio.h>
void main() {
 int n, i, j;
 printf("lato del quadrato? ");
 scanf("%d", &n); scanf("%*c");
 // gestione delle righe
 for(i=0; i<n; i++) {
 // gestione delle colonne
 for(j=0; j<n; j++) {
 printf("* ");
 }
 }
}
```



## Quadrato di asterischi

```
#include <stdio.h>
void main() {
 int n, i, j;
 printf("lato del quadrato? ");
 scanf("%d", &n); scanf("%*c");
 // gestione delle righe
 for(i=0; i<n; i++) {
 // gestione delle colonne
 for(j=0; j<n; j++) {
 printf("* ");
 }
 printf("\n");
 }
}
```





## Triangolo di asterischi

Scrivere un programma che chieda all'utente un numero intero  $n$  e visualizzi a monitor un triangolo rettangolo che abbia per base  $n$  asterischi e altezza  $n$  asterischi.

Ad esempio, se l'utente inserisce il numero 3, il programma visualizza quanto segue:

```
*

* *

* * *
```



# Triangolo di asterischi

```
#include <stdio.h>
void main() {

}
}
```



# Triangolo di asterischi

```
#include <stdio.h>
void main() {
 int n, i, j;

}
}
```



## Triangolo di asterischi

```
#include <stdio.h>
void main() {
 int n, i, j;
 printf("lato del triangolo? ");
 scanf("%d", &n); scanf("%*c");

}
}
```



# Triangolo di asterischi

```
#include <stdio.h>
void main() {
 int n, i, j;
 printf("lato del triangolo? ");
 scanf("%d", &n); scanf("%*c");
 // gestione delle righe
 for(i=0; i<n; i++) {

 }
}
```



## Triangolo di asterischi

```
#include <stdio.h>
void main() {
 int n, i, j;
 printf("lato del triangolo? ");
 scanf("%d", &n); scanf("%*c");
 // gestione delle righe
 for(i=0; i<n; i++) {
 //stampo i asterischi
 for(j=0; j<i+1; j++) {

 }

 }
}
```



# Triangolo di asterischi

```
#include <stdio.h>
void main() {
 int n, i, j;
 printf("lato del triangolo? ");
 scanf("%d", &n); scanf("%*c");
 // gestione delle righe
 for(i=0; i<n; i++) {
 //stampo i asterischi
 for(j=0; j<i+1; j++) {
 printf("* ");
 }
 }
}
```



# Triangolo di asterischi

```
#include <stdio.h>
void main() {
 int n, i, j;
 printf("lato del triangolo? ");
 scanf("%d", &n); scanf("%*c");
 // gestione delle righe
 for(i=0; i<n; i++) {
 //stampo i asterischi
 for(j=0; j<i+1; j++) {
 printf("* ");
 }
 printf("\n");
 }
}
```





## Triangolo di asterischi rovesciato

Scrivere un programma che chieda all'utente un numero intero  $n$  e visualizzi a monitor un triangolo rettangolo rovesciato che abbia per base  $n$  asterischi e altezza  $n$  asterischi.

Ad esempio, se l'utente inserisce il numero 3, il programma visualizza quanto segue:

```
* * *
 * *
 *
```



# Triangolo di asterischi rovesciato

```
#include <stdio.h>
void main() {
```

```
}
```



## Triangolo di asterischi rovesciato

```
#include <stdio.h>
void main() {
 int n, i, j;
```

```
}
```



## Triangolo di asterischi rovesciato

```
#include <stdio.h>
void main() {
 int n, i, j;
 printf("lato del triangolo? ");
 scanf("%d", &n); scanf("%*c");

}
}
```



## Triangolo di asterischi rovesciato

```
#include <stdio.h>
void main() {
 int n, i, j;
 printf("lato del triangolo? ");
 scanf("%d", &n); scanf("%*c");
 // gestione delle righe
 for(i=0; i<n; i++) {

 }
}
```



## Triangolo di asterischi rovesciato

```
#include <stdio.h>
void main() {
 int n, i, j;
 printf("lato del triangolo? ");
 scanf("%d", &n); scanf("%*c");
 // gestione delle righe
 for(i=0; i<n; i++) {
 //stampo i spazi bianchi
 for(j=0; j<i; j++) {

 }
 }
}
```



## Triangolo di asterischi rovesciato

```
#include <stdio.h>
void main() {
 int n, i, j;
 printf("lato del triangolo? ");
 scanf("%d", &n); scanf("%*c");
 // gestione delle righe
 for(i=0; i<n; i++) {
 //stampo i spazi bianchi
 for(j=0; j<i; j++) {
 printf(" ");
 }
 }
}
```



## Triangolo di asterischi rovesciato

```
#include <stdio.h>
void main() {
 int n, i, j;
 printf("lato del triangolo? ");
 scanf("%d", &n); scanf("%*c");
 // gestione delle righe
 for(i=0; i<n; i++) {
 //stampo i spazi bianchi
 for(j=0; j<i; j++) {
 printf(" ");
 }
 //stampo n-i asterischi
 for(j=0; j<n-i; j++) {

 }

 }
}
```





## Triangolo di asterischi rovesciato

```
#include <stdio.h>
void main() {
 int n, i, j;
 printf("lato del triangolo? ");
 scanf("%d", &n); scanf("%*c");
 // gestione delle righe
 for(i=0; i<n; i++) {
 //stampo i spazi bianchi
 for(j=0; j<i; j++) {
 printf(" ");
 }
 //stampo n-i asterischi
 for(j=0; j<n-i; j++) {
 printf("* ");
 }

 }
}
```



## Triangolo di asterischi rovesciato

```
#include <stdio.h>
void main() {
 int n, i, j;
 printf("lato del triangolo? ");
 scanf("%d", &n); scanf("%*c");
 // gestione delle righe
 for(i=0; i<n; i++) {
 //stampo i spazi bianchi
 for(j=0; j<i; j++) {
 printf(" ");
 }
 //stampo n-i asterischi
 for(j=0; j<n-i; j++) {
 printf("* ");
 }
 printf("\n");
 }
}
```