

```
/*
* Politecnico di Milano
* Corso Informatica B - Prof. Loiacono
* Esercizi di: Alessandro A. Nacci
* Per contatti: alessandro.nacci@mail.polimi.it
* -----
* Esercitazione 2
* ESERCIZIO 1: IF ELSE
* -----
* scrivere un programma che dato un intero
* stampa 1 se il numero e' strettamente positivo,
* dispari e minore di 50 o strettamente positivo, pari e maggiore di 100
* stampa 0 altrimenti
*/

#include <stdio.h>
#include <stdlib.h>

int main(){

    int n;

    printf("inserire un numero intero: \n");
    scanf("%d", &n);

    if( (n > 0) && ( ( n%2 != 0 && n <= 50 ) || ( n%2 == 0 && n >= 100 ) ) ){
        printf("1\n");
    }else{
        printf("0\n");
    }

    return 0;
}
```

```
/*
 * Politecnico di Milano
 * Corso Informatica B - Prof. Loiacono
 * Esercizi di: Alessandro A. Nacci
 * Per contatti: alessandro.nacci@mail.polimi.it
 * -----
 * Esercitazione 2
 * ESERCIZIO 2: ANNO BISESTILE
 * -----
 * Scrivere un programma che:
 * dato un numero intero che rappresenta un anno, verificare che l'anno inserito
 * sia bisestile. Ricorda che un anno è bisestile se il suo numero è
 * divisibile per 4, * con l'eccezione degli anni secolari (quelli divisibili
 * per 100) che non sono divisibili per 400.
 */

#include <stdlib.h>
#include <stdio.h>

int main(){

    int anno;

    printf("inserire l'anno: ");
    scanf("%d", &anno);

    if (anno % 4 == 0 && (anno % 100 != 0 || anno % 400 == 0)){
        printf("\nL'anno %d e' bisestile!\n", anno);
    } else {
        printf("\nL'anno %d NON e' bisestile!\n", anno);
    }

    return 0;
}
```

```

/*
* Politecnico di Milano
* Corso Informatica B - Prof. Loiacono
* Esercizi di: Alessandro A. Nacci
* Per contatti: alessandro.nacci@mail.polimi.it
* -----
* Esercitazione 2
* ESERCIZIO 3: SINTASSI C
* -----
* Dire quale è l'output a schermo dei seguenti pezzi di codice C.
*/

//=====  ESEMPIO A  =====

#include <stdlib.h>
#include <stdio.h>

int main(){

    int valore = 0;
    printf("\nInserisci il valore: ");
    scanf("%d", &valore);

    if (valore){
        printf("Mi piace il colore rosso.\n");
        return 0;
    }

    printf("Mi piace il colore blu.\n");

    return 0;

}
// NON ESISTONO I VALORI Boolean True e False!
// Una condizione assume un valore intero pari a:
//     - se la condizione è falsa
//     - 1 se la condizione è vera
//=====

//=====  ESEMPIO B  =====

#include <stdlib.h>
#include <stdio.h>

int main(){

    int anni = 0;
    printf("\nInserisci anni: ");
    scanf("%d", &anni);

    if (anni > 17);           // ERRORE DI SINTASSI!!!!
                            // --> QUI NON CI VUOLE IL PUNTO E VIRGOLA!
    {
        printf("Sei maggiorenne.\n");
        return 0;
    } else {
        printf("Non sei maggiorenne.\n");
    }

    return 0;

}

//=====

```

```
//===== ESEMPIO C =====

#include <stdlib.h>
#include <stdio.h>

int main(){

    int ann1, ann2;

    printf("\nInserisci ann1: ");
    scanf("%d", &ann1);
    printf("\nInserisci ann2: ");
    scanf("%d", &ann2);

    if (ann1 = ann2) // --> IL CONFRONTO DI FA CON '=='!!!!
    {
        printf("I due utenti sono coetanei.\n");
        return 0;
    } else {
        printf("I due utenti non sono coetanei.\n");
    }

    return 0;

}
```

```
//===== ESEMPIO D =====

#include <stdlib.h>
#include <stdio.h>

int main(){

    int a,b,c,d;

    printf("\nInserisci a: ");
    scanf("%d", &a);
    printf("\nInserisci b: ");
    scanf("%d", &b);

    c = 3;
    d = 0;

    if ( a+b == 20 )
        c = a-b;
        d = c * 2;
    // else {           --> variante dell'esercizio!
    // c = 100;         che da errore di compilazione
    //}                 poiché senza parentesi
    //                  il compilatore non sa a chi
    //                  attribuire il ramo ELSE!

    printf("c = %d\n",c);
    printf("d = %d\n",d);

}
```

```
return 0;
```

```
}
```

```
// PROVARE CON I SEGUENTI VALORI IN INGRESSO:
```

```
// a = 10 | b = 5 ==> c = 3 | d = 6;
```

```
// a = 17 | b = 3 ==> c = 14 | d = 28;
```

```
//
```

```
// SE HO PIU ISTRUZIONI DA ESEGUIRE DEVO
```

```
// METTERE LE PARENTESI GRAFFE DOPO L'IF
```

```
//=====
```

```

/*
* Politecnico di Milano
* Corso Informatica B - Prof. Loiacono
* Esercizi di: Alessandro A. Nacci
* Per contatti: alessandro.nacci@mail.polimi.it
* -----
* Esercitazione 2
* ESERCIZIO 4: ELSE-IF
* -----
* scrivere un programma che prenda in ingresso un numero intero
* e svolga le seguenti operazioni: se il numero inserito  $\sqrt{\textcircled{}}$  pari scriva
* a schermo "NUMERO PARI". Altrimenti, se il numero inserito  $\sqrt{\textcircled{}}$ 
* dispari ed  $\sqrt{\textcircled{}}$  divisibile per 17 scriva "NUMERO DISPARI DIVISIBILE PER 17".
* Negli altri casi far scrivere "NUMERO NON RICONOSCIUTO".
*/

#include <stdio.h>
#include <stdlib.h>

int main(){

    int n;

    printf("inserire un numero ");
    scanf("%d", &n);
    printf("\n");

    if (n % 2 == 0){
        printf("IL NUMERO E' PARI\n");
    } else if (n % 17 == 0){
        printf("NUMERO DISPARI DIVISIBILE PER 17\n");
    } else {
        printf("NUMERO NON RICONOSCIUTO\n");
    }

    return 0;
}

```

```
/*
 * Politecnico di Milano
 * Corso Informatica B - Prof. Loiacono
 * Esercizi di: Alessandro A. Nacci
 * Per contatti: alessandro.nacci@mail.polimi.it
 * -----
 * Esercitazione 2
 * ESERCIZIO 5: SWITCH
 * -----
 * scrivere un programma che prenda in ingresso un numero intero
 * e svolga le seguenti operazioni: se il numero inserito  $\sqrt{x}$  compreso
 * tra 1 e 3 stampa il numero inserito altrimenti se il numero inserito
 *  $\sqrt{x}$  tra 4 e 7 stampa il numero e i successivi fino a 7
 * altrimenti indica che il numero inserito non  $\sqrt{x}$  valido
 */
```

```
/* */
```

```
#include <stdio.h>
```

```
int main(){
```

```
    int n;
```

```
    printf("inserire un numero\n");
    scanf("%d", &n);
```

```
    switch (n){
        case 1: printf("numero 1\n");
                break;
        case 2: printf("numero 2\n");
                break;
        case 3: printf("numero 3\n");
                break;
        case 4: printf("numero 4\n");
        case 5: printf("numero 5\n");
        case 6: printf("numero 6\n");
        case 7: printf("numero 7\n");
                break;
        default: printf("errore\n");
    }
```

```
    return 0;
```

```
}
```

```
/*
* Politecnico di Milano
* Corso Informatica B - Prof. Loiacono
* Esercizi di: Alessandro A. Nacci
* Per contatti: alessandro.nacci@mail.polimi.it
* -----
* Esercitazione 2
* ESERCIZIO 6: FOR-FIBONACCI
* -----
* stampare i primi n numeri della serie d fibonacci
*/
```

```
#include <stdio.h>
```

```
int main(){
```

```
    int n, i, temp;
```

```
    int fib = 1, fibprec = 0;
```

```
    printf("inserire un numero");
```

```
    scanf("%d", &n);
```

```
    for(i =0; i<n; i++){
```

```
        temp = fib;
```

```
        fib = fib + fibprec;
```

```
        fibprec = temp;
```

```
        printf("%d\n", fib);
```

```
    }
```

```
    return 0;
```

```
}
```

```
/*
 * Politecnico di Milano
 * Corso Informatica B - Prof. Loiacono
 * Esercizi di: Alessandro A. Nacci
 * Per contatti: alessandro.nacci@mail.polimi.it
 * -----
 * Esercitazione 2
 * ESERCIZIO 7: FOR-MEDIA
 * -----
 * dati n numeri (dove n è specificato dall'utente)
 * calcolarne la media
 */
```

```
#include <stdio.h>
```

```
int main(){

    int n, i;
    float sum = 0, media, add;

    printf("inserire il numero n: ");
    scanf("%d", &n);

    for(i = 0; i < n; i++){

        printf("inserire un numero: ");
        scanf("%f", &add);
        sum = sum + add;

    }

    media = sum/n;

    printf("la media e': %f \n    ", media);

    return 0;
}
```

```

/*
* Politecnico di Milano
* Corso Informatica B - Prof. Loiacono
* Esercizi di: Alessandro A. Nacci
* Per contatti: alessandro.nacci@mail.polimi.it
* -----
* Esercitazione 2
* ESERCIZIO 8: DOWHILE-MEDIA
* -----
* calcolarne la media di n numeri interi positivi
* n non è inserito dall'utente.
* si calcola la media dei numeri inseriti
* fino a che non viene inserito un numero negativo
* da parte dell'utente
*/

#include <stdio.h>

int main(){

    int cont = 0;
    float sum = 0, media, numero;

    // in questo caso conviene usare il do-while
    // e non il costrutto while perché la
    // richiesta di inserimento deve essere sempre
    // fatta almeno una volta!

    // FAR NOTARE LA DIFFERENZA TRA DO WHILE
    // E MAGARI MOSTRARE COME SI FA QUESTO CODICE
    // CON IL WHILE!!

    do {

        printf("inserire un numero: ");
        scanf("%f", &numero);

        if (numero >= 0){
            sum = sum + numero;
            cont++;
        }

    } while(numero >= 0);

    media = sum/cont;

    printf("Hai inserito %d numeri\n", cont);
    printf("La media e': %f \n    ", media);

    return 0;
}

```

```
/*
* Politecnico di Milano
* Corso Informatica B - Prof. Loiacono
* Esercizi di: Alessandro A. Nacci
* Per contatti: alessandro.nacci@mail.polimi.it
* -----
* Esercitazione 2
* ESERCIZIO 9: CICLI-BREAK-CONTINUE
* -----
* mostrare quale è l'output del seguente
* frammento di codice
*/

#include <stdio.h>

int main(){

    int i = 0;

    for (i = 0; i < 10; i++){

        if (i == 8)
            break;

        if (i % 2 == 0)
            continue;

        printf("%d\n",i);

    }

    return 0;
}

// L'OUTPUT E' 1 - 3 - 5 - 7
// NUMERI DISPARI TRA 1 e 7
```

```

/*
* Politecnico di Milano
* Corso Informatica B - Prof. Loiacono
* Esercizi di: Alessandro A. Nacci
* Per contatti: alessandro.nacci@mail.polimi.it
* -----
* Esercitazione 2
* ESERCIZIO 10: ARRAY-MAX-MIN
* -----
* dati 10 numeri interi, trovare quale è
* il numero massimo ed il numero minimo
* tra questi. I numeri devono essere
* salvati in un array
*/

#include <stdio.h>

int main(){

    int numeri[10];
    int i = 0;

    for (i = 0; i < 10; i++){
        printf("\nInserisci il numero %d-esimo: ", i);
        scanf("%d", &numeri[i]);
    }

    int max = 0;
    int min = 2147483647; //2^(n-1) - 1 :::: CPL2 max value
                        //n = 32 poiché int è rappresentato
                        //con 32 bit.

    for (i = 0; i < 10; i++){

        if (numeri[i] > max)
            max = numeri[i];

        if (numeri[i] < min)
            min = numeri[i];

    }

    printf("MIN = %d\n", min);
    printf("MAX = %d\n", max);

    return 0;
}

```

```

/*
* Politecnico di Milano
* Corso Informatica B - Prof. Loiacono
* Esercizi di: Alessandro A. Nacci
* Per contatti: alessandro.nacci@mail.polimi.it
* -----
* Esercitazione 2
* ESERCIZIO 11: ARRAY-OUT-OF-MEM
* -----
* mostrare quale è l'output del seguente
* frammento di codice
*/

//=====  ESEMPIO A  =====

#include <stdio.h>

int main(){

    int arr[3] = {100,200,300};

    printf("%d\n", arr[0]);
    printf("%d\n", arr[1]);
    printf("%d\n", arr[2]);
    printf("%d\n", arr[3]); // --> valori a caso!!

    return 0;
}

// L'OUTPUT E' 100 - 200 - 300 - valori_a_caso
//=====

//=====  ESEMPIO B  =====

#include <stdio.h>

int main(){

    int arr[3];

    // array non inizializzato
    printf("%d\n", arr[0]); // --> valori a caso!!
    printf("%d\n", arr[1]); // --> valori a caso!!
    printf("%d\n", arr[2]); // --> valori a caso!!

    arr[0] = 100;
    arr[1] = 200;
    arr[2] = 300;

    printf("%d\n", arr[0]); // --> stampa 100
    printf("%d\n", arr[1]); // --> stampa 200
    printf("%d\n", arr[2]); // --> stampa 300

    return 0;
}

//=====

```

```

/*
* Politecnico di Milano
* Corso Informatica B - Prof. Loiacono
* Esercizi di: Alessandro A. Nacci
* Per contatti: alessandro.nacci@mail.polimi.it
* -----
* Esercitazione 2
* ESERCIZIO 12: ENUM
* -----
* mostrare quale è l'output del seguente
* frammento di codice
*/

//===== ESEMPIO A =====

#include <stdio.h>

int main(){

    typedef enum{bianco,azzurro_chiario,giallo,rosso,verde_scuero,rosa,
        azzurro_scuero,verde_chiario,nero,marrone} colore;

    int auto_codice[3];          //ATTENZIONE QUI!
    colore auto_colore[3];      // tipo_di_dato nome_varibile[]

    auto_codice[0] = 1000;
    auto_colore[0] = giallo;

    auto_codice[1] = 4000;
    auto_colore[1] = rosa;

    auto_codice[2] = 8000;
    auto_colore[2] = nero;

    int i;

    for (i = 0; i < 3; i++ )
        printf("L'auto con codice %d e' di colore %d\n",
            auto_codice[i], auto_colore[i]);

    return 0;
}

//L'output del codice è il seguente
//-----
//L'auto con codice 1000 e' di colore 2
//L'auto con codice 4000 e' di colore 5
//L'auto con codice 8000 e' di colore 8
//-----
// Questo perché i valori specificati nelle ENUM in
// realtà sono gli interi corrispondenti
//=====

```