

Elenco comandi:

oggetto	codice	utilizzo
funzione main	int main (void) { ... return 0; }	void specifica che main non prende alcun argomento - int specifica che main restituisce un valore intero
librerie	#include <stdio.h>	Funzionalità base
	#include <math.h>	Funzioni matematiche base
	#include <time.h>	Per data e ora
	#include <string.h>	Manipolare le stringhe
	#include <unistd.h>	Per usare usleep
	#include <stdlib.h>	Per lo srand (valori casuali)
identificatori	int	Valore numerico standard
	float	Con decimale
	double	Numero intero grande
	char	Una lettera o un numero
stampa	printf ("_testo %specificatore_", identificatore); printf ("inserire %d", i);	Stampa a schermo
specificatori di formato per printf	%c	char (carattere singolo)
	%d - %i	int (intero decimale)
	%e	float o doble (esponenziale)
	%f	float o double (reale)
	%o	int - n base 8 senza segno
	%p	pointer - indica variabile
	%s	array di char - stringa di caratteri
inserire	scanf ("specificatore di formato", &nome_identificatore);	Inserisce un carattere da tastiera e lo da all'int
	if (condizione per il loop) {_azione da compiere_}; else	Un loop a due sole possibilità NOTA: possono essere

controllori di flusso	<code>{_cosa fare altrimenti_};</code>	incatenati più if: if{if...
	<code>switch (nome_variabile) { case n: _comando per caso n_; break; };</code>	Un elenco di conseguenze in base alle variazioni della variabile
	<code>for (inizializzatore; verifica; incremento/decremento) {_comando da eseguire_}; es: for (i=0; i < 10; i++) {printf ("inserisci %d:", i+1); scanf ("%d" &n);</code>	Un comando di inizializzazione (un int o un'equazione) che viene verificata ($i < 10$) e se vera Esegue l'eventuale incremento o decremento eseguendo il comando
	<code>do {elenco di comandi da eseguire} while {condizione}; es: do { printf ("%d", n); n++} while {n < 11};</code>	Esegue ciclicamente la stringa di comandi presente nel do fintanto che il while è vero. Di solito contiene un incremento o decremento di una variabile
array	<code>int nome_variabile[n]; float float_array[n]; char char_array[n];</code>	Collezione organizzata di elementi simili. Genera una tabella di n colonne e un riga
	<code>float_array[4] = 87.43</code>	Nella casella n.5 dell'array di tipo float si inserisce il valore decimale 87.43
	<code>float _variabile_ = float_array[1]</code>	Assegna ad una variabile float il valore della seconda casella dell'array float
Comandi Shell linux	<code>cd "nome"</code>	passa da una directory corrente a "nome"
	<code>mkdir</code>	Crea una directory
	<code>rmdir</code>	Rimuove una directory
	<code>ls</code>	Mostra l'elenco delle directory e dei file

	<code>pwd</code>	mostra percorso file o directory
	<code>Gedit file.c &</code>	Apre un file dove si può programmare
	<code>gcc file.c -o file</code>	Converte il file in un programma eseguibile
	<code>./file</code>	Esegue il programma "file"

esercizi programmazione per esame:

Programma	consegna
<pre>#include <stdio.h> int main(void) { printf("hello world!\n\n"); return 0; }</pre>	Crea un programma che stampa la frase "hell world".
<pre>#include <stdio.h> int main(void) { int w; int v = 42; printf("La variabile v contiene %i\n", v); w = 13; w = 2 + 3 * 5; printf("La variabile w contiene %i\n", w); return 0; }</pre>	Un programma che mostra il valore di v e dopo averne assegnato uno a w ne modifica il numero stampandone il risultato finale
<pre>#include <stdio.h> int main(void) { int lato; lato = 13; int volume; volume = lato * lato * lato; printf("Il volume del cubo di lato %i e' %i\n", lato, volume); return 0; }</pre>	Crea un programma che calcola il volume di un cubo di dato lato e stampa il risultato.
<pre>#include <stdio.h> int main(void) { int ore1 = 16;</pre>	Crea un programma che calcola il tempo passato espresso in ore minuti e secondi, già dati.

```

int minuti1 = 5;
int secondi1 = 58;
int ore2 = 8;
int minuti2 = 20;
int secondi2 = 5;

int totale1 = secondi1 + minuti1 * 60 + ore1 *
60 * 60;
int totale2 = secondi2 + minuti2 * 60 + ore2 *
60 * 60;
int passati = totale2 + 24 * 60 * 60 - totale1;

printf("sono passati %i secondi\n", passati);

return 0;
}

```

```

#include <stdio.h>

int main(void) {
    int i, j;
    printf("Inserisci i: ");
    scanf("%i", &i);
    printf("Inserisci j: ");
    scanf("%i", &j);
    double media;
    media = (i + j) / 2.0;

    printf("La media di %i e %i è %.2f\n", i, j,
media);

    return 0;
}

```

```

#include <stdio.h>

int main(void) {
    int centesimi;
    printf("Inserisci i centesimi da tradurre: ");
    scanf("%i", &centesimi);

```

Crea un programma che calcola la media tra due valori inseriti da tastiera con precisione di due decimali.

Crea un programma che chiede di inserire un numero x di centesimi cui verrà stampato l'equivalente in monete da 2, 1, 0.50, 0.20, 0.10, 0.05, 0.02 e 0.01 euro.

```
printf("%i centesimi si dividono come segue:\n", centesimi);

printf("%i monete da 2 euro\n", centesimi / 200);
centesimi %= 200;
printf("%i monete da 1 euro\n", centesimi / 100);
centesimi %= 100;
printf("%i monete da 50 centesimi\n",
centesimi / 50);
centesimi %= 50;
printf("%i monete da 20 centesimi\n",
centesimi / 20);
centesimi %= 20;
printf("%i monete da 10 centesimi\n",
centesimi / 10);
centesimi %= 10;
printf("%i monete da 5 centesimi\n",
centesimi / 5);
centesimi %= 5;
printf("%i monete da 2 centesimi\n",
centesimi / 2);
centesimi %= 2;
printf("%i monete da 1 centesimo\n",
centesimi);

return 0;
}
```

```
#include <stdio.h>

int main(void) {
    char v;
    //v = 65; // per nerd
    v = 'A'; // per esseri umani
```

Programma che utilizza l'indicatore Char per stampare a video dei caratteri dell'alfabeto.

<pre> printf("v = %i\n", v); printf("v = %c\n", v); v += 2; printf("v = %i\n", v); printf("v = %c\n", v); return 0; } </pre>	
<pre> // calcola e stampa la somma dei primi n numeri interi #include <stdio.h> int main(void) { int triangular = 0; int n; printf("Inserisci n: "); scanf("%i", &n); //triangular = 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 // + n; // NO for (int i = 1; i <= n; i = i + 1) { triangular = triangular + i; } // indentato o nested printf("La somma dei primi %i numeri interi e': %i\n", n, triangular); return 0; } </pre>	<p>Crea un programma che esegue la somma dei primi n numeri interi a partire da 0 fino a n.</p>
<pre> #include <stdio.h> int main(void) { int n; printf("Inserisci n: "); scanf("%i", &n); </pre>	<p>Crea un programma che stampa una piramide di asterischi con un numero, da inserire, di righe.</p>

```

int spazi = n - 1;
int asterischi = 1;
for (int riga = 1; riga <= n; riga = riga + 1) {
    // stampo spazi
    for (int i = 1; i <= spazi; i = i + 1)
        printf(" ");

    // stampo asterischi
    for (int i = 1; i <= asterischi; i = i +
1)
        printf("*");

    spazi = spazi - 1;
    asterischi = asterischi + 2;
    printf("\n");
}

return 0;
}

```

```

#include <stdio.h>

int main(void) {
    // char c = '\n'; // andata a capo
    // char c = '\t'; // tabulazione orizzontale
    // char c = '\''; // carattere apice singolo
    // char c = '\"'; // carattere apice doppio
    char c = '\\'; // carattere slash
    printf("%c\t\"hello\"\n", c);
    return 0;
}

```

```

#include <stdio.h>

int main(void) {
    int n;
    printf("Inserisci n: ");
    scanf("%i", &n);

    int factorial = 1;

```

Crea un programma che stampa il valore di char.

Crea un programma che calcola il fattoriale dato un numero

```

for (int i = 1; i <= n; i++) {
    factorial = factorial * i;
    printf("%4i    %7i\n", i, factorial);
}

return 0;
}

```

```

#include <stdio.h>

int main(void) {
    int n;
    printf("Inserisci n: ");
    scanf("%i", &n);

    int abs = (n < 0) ? -n : n;

    printf("Il valore assoluto di %i e' %i\n",
n, abs);

    return 0;
}

```

```

#include <stdio.h>

int main(void) {
    float n1, n2;
    char op;
    printf("Inserisci l'espressione: ");
    scanf("%f %c %f", &n1, &op, &n2);

    switch (op) {
        case '+':
            printf("%f + %f = %f\n", n1, n2, n1
+ n2);
            break;
        case '-':
            printf("%f - %f = %f\n", n1, n2, n1 - n2);
            break;
        case '*':
        case 'x':

```

Crea un programma che restituisce il valore assoluto di un valore inserito.

Crea una calcolatrice capace di fare le quattro operazioni con due valori da inserire.

```

        printf("%f * %f = %f\n", n1, n2, n1 *
n2);
        break;
    case '/':
        if (n2 != 0)
            printf("%f / %f = %f\n", n1, n2,
n1 / n2);
        else
            printf("Non posso dividere per 0!\n");
        break;
    default:
        printf("Operatore sconosciuto\n");
        break;
    }

    return 0;
}

```

```

#include <stdio.h>

int main(void) {
    int n;

    do {
        printf("Inserisci n: ");
        scanf("%i", &n);
    }
    while (n < 0);

    int divisori = 0;
    for (int d = 1; d <= n; d++)
        if (n % d == 0) // se d divide n
            divisori++; // ho trovato un altro
divisore, me lo segno

    printf("%i ha %i divisori\n", n, divisori);

    return 0;
}

```

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

Crea un programma che trova i divisori di un numero da inserire.

Crea un programma che dice se un numero inserito è un numero primo

```

int main(void) {
    int n;

    do {
        printf("Inserisci n: ");
        scanf("%i", &n);
    }
    while (n < 0);

    int found_divisor = 0; // 0 significa false,
    e' un "flag"
    for (int d = 2; d * d <= n; d++)
        if (n % d == 0) { // se d divide n
            found_divisor = 1; // ho trovato
un divisore tra 2 ed n escluso
        }

    if (n >= 2 && found_divisor == 0) // se
non ho trovato un divisore
        printf("%i e' primo\n", n);
    else
        printf("%i non e' primo\n", n);

    return 0;
}

```

o meno

```

#include <stdio.h>
#include <unistd.h>
#include <math.h>

int main(void) {
    int s = 0;
    double c = 0.0;

    while (1) {
        s = (sin(c) + 1) * 40;
        c += 0.1;

        for (int i = 1; i <= s; i++)
            printf(" ");

        printf("\n");
    }
}

```

Crea un programma che stampa una cometa con i *.

<pre> // dormi per 0.1 secondi usleep(100000); // 100000 millonesimi di secondo } return 0; } </pre>	
<pre> #include <stdio.h> #include <stdlib.h> #include <time.h> int main(void) { //int t = time(NULL); //srand(t); srand(time(NULL)); // mescolare la carte int r = rand() % 6 + 1; printf("r = %i\n", r); r = rand() % 6 + 1; printf("r = %i\n", r); r = rand() % 6 + 1; printf("r = %i\n", r); return 0; } </pre>	<p>Crea un programma che simula i risultati possibili dati dal lancio di due dadi a sei facce.</p>
<pre> #include <stdio.h> int main(void) { int x, y; printf("Inserisci x: "); scanf("%i", &x); printf("Inserisci y: "); scanf("%i", &y); if (x == y + 1 y == x + 1) printf("%i e %i sono consecutivi\n", x, y); else printf("%i e %i non sono consecutivi\n", x, y); return 0; } </pre>	<p>Crea un programma che, inseriti due numeri, determina se sono o meno consecutivi.</p>

```
#include <stdio.h>

int main(void) {
    int n;
    printf("Inserisci n: ");
    scanf("%i", &n);

    int riga = 1;
    while (riga <= n) {
        // stampo riga - 1 spazi
        int i = 1;
        while (i <= riga - 1) {
            printf(" ");
            i++;
        }

        // stampo un asterisco e vado a capo
        printf("*\n");

        riga++;
    }

    return 0;
}
```

Crea un programma che stampa una diagonale di n asterischi.

```
#include <stdio.h>

int main(void) {
    int n;
    do {
        printf("Inserisci n: ");
        scanf("%i", &n);
    }
    while(n < 0);

    int perfetto = 0; // falso
    for (int i = 0; i * i <= n; i++)
        if (i * i == n) {
            perfetto = 1; // vero
            break;
    }
}
```

Crea un programma che dato un n<0 dice se è un numero perfetto (= n. naturale, uguale alla somma dei suoi divisori positivi, escludendo il numero stesso).

<pre> } if (perfetto == 1) printf("Il numero %i e' perfetto\n", n); else printf("Il numero %i non e' perfetto\n", n); return 0; } </pre>	
<pre> #include <stdio.h> #include <math.h> int main(void) { int n, radice; do { printf("Inserisci n >= 0: "); scanf("%i", &n); } while (n < 0); // radice = sqrt(n); int i; for (i = 1; i * i <= n; i++); radice = i - 1; printf("La radice approssimata per difetto di %i e' %i\n", n, radice); return 0; } </pre>	<p>Crea un programma che calcola la radice di un numero da inserire.</p>
<pre> #include <stdio.h> int main(void) { int n; do { printf("Inserisci n: "); scanf("%i", &n); } while (n < 2); </pre>	<p>Crea un programma che calcola la serie di Fibonacci stampandola in un array di un numero n inserito.</p>

```

long fib[n];

fib[0] = fib[1] = 1;

for (int pos = 2; pos < n; pos++)
    fib[pos] = fib[pos - 1] + fib[pos -
2];

for (int pos = 0; pos < n; pos++)
    printf("%li ", fib[pos]);

printf("\n");

return 0;
}

```

```

#include<stdio.h>
#include<math.h>
int main(void){
    int n;
    printf("\nInserire un numero positivo: ");
    scanf("%i", &n);

    int nn=n;

    int cifre=(int)(log10(n) + 1);
    printf("num_cifre%i\n",cifre);

    int somma=0;
    int cifra=0;
    while(nn!=0){
        cifra=nn%10;
        somma+= pow(cifra, cifre);
        nn/=10;
    }
    if(somma==n){
        printf("\nIl numero %i è un numero
narcisista\n", n);
    } else {
        printf("\nIl numero %i non è un numero

```

Crea un programma che determina se un numero inserito è narcisista o meno (= se si eleva ciascuna cifra all'esponente pari al numero di cifre del numero stesso e si sommano i risultati di queste operazioni, si ottiene il numero stesso)

<pre>narcisista\n", n); } return 0; }</pre>	
<pre>#include <stdio.h> int mcd(int m, int n) { while (n != 0) { int temp = m % n; m = n; n = temp; } return m; } int main(void) { int m, n; printf("Inserisci m: "); scanf("%i", &m); printf("Inserisci n: "); scanf("%i", &n); int risultato = mcd(m, n); printf("L'mcd di %i e %i e' %i\n", m, n, risultato); return 0; }</pre>	<p>Crea un programma che utilizza l'algoritmo di Euclide per calcolare il massimo comune divisore tra due numeri m ed n.</p>
<pre>#include <stdio.h> #include <math.h> int main(void) { int n; do { printf("Dammi la dimensione dell'array: "); scanf("%i", &n); } while(n < 0); int array[n]; for (int i = 0; i < n; i++) {</pre>	<p>Crea un array di dimensione a scelta e lo stampa a schermo.</p>

<pre> array[i] = pow(2, i); } // stampare l'array for (int i = 0; i < n; i++) { printf("%i ", array[i]); } printf("\n"); } </pre>	
<pre> #include <stdio.h> // l'n-esimo numero di Fibonacci // fib(0) = 1 [caso base] // fib(1) = 1 [caso base] // fib(n) = fib(n - 1) + fib(n - 2) [se n > 1, caso ricorsivo] long fib(int n) { if (n == 0) return 1; else if (n == 1) return 1; else return fib(n - 1) + fib(n - 2); } int main(void) { int n; printf("Inserisci n: "); scanf("%i", &n); printf("fib(%i) = %li\n", n, fib(n)); return 0; } </pre>	<p>Crea un programma che stampa i primi n numeri della serie di Fibonacci.</p>
<pre> #include <stdio.h> int main(void) { int n; do { </pre>	<p>Crea un programma che calcola il massimo, il minimo e la media di n numeri.</p>

```
printf("Inserisci n: ");
scanf("%i", &n);
}

while (n <= 0);

int min, max, sum;

printf("Inserisci il numero #1: ");
int x;
scanf("%i", &x);
min = max = sum = x; // tutti e tre diventano x

for (int i = 2; i <= n; i++) {
    printf("Inserisci il numero #%-i: ", i);
    scanf("%i", &x);

    if (x > max)
        max = x;

    if (x < min)
        min = x;

    sum += x;
}

printf("Minimo = %i, massimo = %i\n", min,
max);
printf("Media = %.2f\n", sum / (float) n);

return 0;
}
```

```
#include <stdio.h>

int main(void) {
    int n;
    do {
        printf("Inserisci n: ");
        scanf("%i", &n);
    }
    while (n < 0);
```

Crea un programma che crea un array di n lunghezza e poi ti fa inserire i valori nelle singole caselle dell'array e lo stampa a schermo spostando tutti i valori di una posizione.

```

int arr[n];
for (int pos = 0; pos < n; pos++) {
    printf("Inserisci il numero #%i: ", pos);
    scanf("%i", &arr[pos]);
}

for (int pos = 0; pos < n; pos++)
    printf("%i ", arr[pos]);
printf("\n");

// sposto a sinistra l'array
int temp = arr[0];
for (int pos = 1; pos < n; pos++)
    arr[pos - 1] = arr[pos];
arr[n - 1] = temp;

for (int pos = 0; pos < n; pos++)
    printf("%i ", arr[pos]);
printf("\n");

return 0;

```

```

#include <stdio.h>
#include "functions.h"

int main(void) {
    printf("%i\n", min(8, -2, -6));
    printf("%i %i\n", are_sorted(3, 8, 7, 9), are_sorted(3,
8, 8, 9));
    print_range('f', 'w');
    printf("%i\n", is_perfect(8128));
    print_binary(8128);
    printf("\n");
    char array[] = { 'a', 'd', 'd', 'a' };
    printf("%i\n", is_palindrome(array, 4));
    printf("%i\n", is_palindrome_rec(array, 4));

    return 0;
}

```

```

#include <stdio.h>

int main(void) {
    int n;

```

Crea un programma che esegue la sommatoria di due numeri uno dei quali decimale.

```

float x;

do {
    printf("Inserisci n >= 0: ");
    scanf("%i", &n);
}
while (n < 0);

printf("Inserisci x: ");
scanf("%g", &x);

float termine = 1;
float risultato = 1;
for (int i = 1; i <= n; i++) {
    termine = termine * x / i;
    risultato += termine;
}

printf("risultato = %g\n", risultato);

return 0;
}

#include <stdio.h>

int is_prime(int n) {
    if (n < 2)
        return 0;

    for (int d = 2; d < n; d++)
        if (n % d == 0)
            return 0;

    return 1;
}

void fill_with_primes(int arr[], int length) {
    for (int pos = 0, next = 2; pos < length; next++)
        if (is_prime(next))
            arr[pos++] = next;
}

int main(void) {
    int length;

    do {
        printf("Inserisci length >= 0: ");
        scanf("%i", &length);
    }
    while (length < 0);
}

```

Crea un programma che restituisce i primi n numeri primi.

<pre>int arr[length]; fill_with_primes(arr, length); for (int pos = 0; pos < length; pos++) printf("%i ", arr[pos]); printf("\n"); return 0; }</pre>	
<pre>#include <stdio.h> double e(int precision) { double approx = 0.0; int factorial = 1; int n; for (n = 0; n <= precision; n++) { approx += 1.0 / factorial; factorial *= n + 1; } return approx; } int main(void) { int precision; printf("precision: "); scanf("%i", &precision); printf("%.20g\n", e(precision)); return 0; }</pre>	Crea un programma che in base ad un numero n modifica la precisione dei decimali del numero di Eulero.
<pre>#include <stdio.h> int is_prime(int n) { int divisor; if (n < 2) return 0; for (divisor = 2; divisor < n; divisor++) if (n % divisor == 0) return 0; return 1; }</pre>	Crea un programma che calcola la quantità di numeri primi presenti tra 0 e n.

```
int count_primes(int max) {
    return max > 0 ? count_primes(max - 1) +
is_prime(max) : 0;
}

int main(void) {
    int max;

    printf("max: ");
    scanf("%i", &max);

    printf("Ci sono %i numeri primi tra 0 e %i\n",
        count_primes(max), max);

    return 0;
}
```