C++ introduttivo Cheat Sheet

by Gray00 via cheatography.com/146241/cs/31604/

Tipi primitivi

int	Intero *(Es: -1, 1, 2, 3..)*
long	Intero "grande"
float	Numero con la virgola "piccolo"
double	Numero con la virgola "grande"
bool	Vero o Falso
char	Carattere *(Es: 'a', 'b', ...)*

Tipi di dato utilizzabili per le variabili

Input / Output

`#include <stdio.h>`	Necessario per i/o
`using namespase std;`	Evita di specificare std::
`cin>>variabile;`	Inserire a schermo un valore
`cout<<variabile;`	Stampa a schermo il contenuto di una variabile
`cout<<"frase";`	Mostra a schermo una frase
`cout<<"..."<<endl;`	Consente di andare a capo
`cout<<"...\n";`	Un altro modo per andare a capo

Comandi di base per gestire la stampa e l'inserimento a schermo.

Commenti

```
// commento su singola riga
/* commento multipla riga */
```

I commenti sono righe di codice utili per lasciare informazioni e rendere il codice più leggibile

For

```
for (int i = 0; i < n; i++){
       // ...
       // istruzioni eseguite n volte
       // ...
}
```

Struttura del ciclo for

While

```
while(condizione == true) {
       // ...
       // istruzioni eseguite fino a quando
       // la condizione è soddis fatta
       // ...
}
```

Struttura del ciclo while

Vettori (Array)

`tipo v[dim];`	crea un vettore di nome "v" di tipo "tipo" con "dim" elementi
`tipo v[] = {el1, el2, ... } ;`	Inizializza un vettore con gli elementi el1, el2 ecc ...
`v[indice] = valore;`	Inserisce "valore" dentro in posizione "indice"

- I vettori cominciano ad indice 0
- Es assegnare in **terza** posizione: `v[2] = 10;`

Gestione file

`#include <fstream.h>`	Importa la libreria file
`ifstream file;`	Crea un lettore di file
`ofstream file;`	Crea uno scrittore di file
`file.open("nome", [mode]);`	Apre i file in un determinata modalità
`fs.is_open()`	Funzione che restituisce se un file è aperto
`file >> var;`	Legge una linea da un file
`while(file.eof())`	"Finchè un file non termina"
`file.close();`	Chiude un file

Costanti per le modalità
ios::in // Apre il file in lettura
ios::out // Apre il file in scrittura
ios::app // Aggiunge al contenuto del file
ios::trunc // Elimina il contenuto precedente

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
http://crosswordcheats.com

Condizioni

A == B	Se A è uguale a B, restituisce vero.
A != B	Se A è diverso da B, restituisce vero.
A < B	Se A è minore di B, restituisce vero.
A > B	Se A è maggiore di B, restituisce vero.
A <= B	Se A è minore o uguale di B, restituisce vero.
A >= B	Se A è maggiore o uguale di B, restituisce vero.
A ! B	A not B
A && B	A and B, vero se entrambe le condizioni sono vere.
A || B	A or B, vero se almeno una condizione è vera.

Le espressioni booleane sono valutate da sinistra a destra!

Random

`#include <ct ime>`	Include la libreria "time"
`srand( tim e(N ULL);`	Permette la generazione di numeri casualli
`n = rand()%max + min;`	Trova un numero casuale tra max e min

Comandi per la generazione di numeri casuali

Errori comuni

Non si può mettere spazi nel nome delle variabili

Quando si vuole mostrare a schermo un testo bisogna inserire " "

if singola condizione, while *"fino a quando"* la condizione è soddisfatta

for quando si conosce il numero di iterazioni, while *"fino a quando"* la condizione è soddisfatta

Errori comuni da evitare

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
http://crosswordcheats.com

Selection Sort

```
void seleSort(int vet[], int dim){
    int  tmp, imin;
    for (int i = 0; i < dim - 1; i++){
        imin = i;
        for (int j = i+1; j < dim; j++){
            if(vet[j] < vet[imin])
                imin = j;
        }

        if (i != imin){
            tmp = vet[imin];
            vet[imin] = vet[i];
            vet[i] = tmp;
        }
    }
}
```

Algoritmo di ordinamento per selezione

Insert Sort

```
void insertSort(int vet[], int dim, int corrente){
   int dove=0;

    whi le( (ve t[dove] < corrente) && (dove <
dim)) {
dove++;
   }

    shi ftR (vet, dove, dim);
    vet [dove] = corrente;
}
// funzione di sposta mento
void shiftR(int vet[], int buco, int fine){
   int x;
   for (x = fine - 1; x >= buco; x--)
        vet[x + 1] = vet[x];
}
```

Algoritmo di ordinamento per inserimento

Bubble Sort

```
void bubbleSort(int vet[], int tanti){
   int x, y, min;
   int temp;
   bool scambia;
   for (x = 0; x < tanti-1; x++) {
scambi a=f alse;
for (y = 0; y < tanti-1-x; y++) {
if (vet[y] > vet[y+1]) {
temp = vet[y];
vet[y] = vet[y+1];
vet[y+1] = temp;
scambi a=true;
}
}
if(sca mbi a== false){
break;
}
   }
}
```

Algoritmo di ordinamento bubble sort

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
http://crosswordcheats.com

Quick Sort

```
void quick_sort ( int vett[], int estremo_sx, int
estremo_dx )
{
   int sx, dx;
   int q ;
   if ( estremo_sx < estremo_dx ) {
        vis ualizza ( vett, estrem o_sx,
estremo_dx );
       q = partiziona ( vett, estrem o_sx,
estremo_dx );
        qui ck_sort ( vett, estrem o_sx, q-1 );
        qui ck_sort ( vett, q+1, estremo_dx );
   }
}
int partiziona (int vett[], int sx, int dx){
   int pivot, ipivot;
    ipivot = sx;
    pivot = vett[i pivot];

    whi le(sx < dx) {
        while(( vett[sx] <= pivot ) && ( sx < dx
)) {
            sx++;
       }
        while( vett[dx] > pivot ){
            dx--;
       }
       if( sx < dx ){
            scambia ( vett, sx, dx) ;
       }
   }
    scambia ( vett, ipivot, dx ) ;
    return dx;
}
void scambia (int vet[], long x, long y){
   int tmp = vet[x];
    vet[x] = vet[y];
    vet[y] = tmp;
}
```

Algoritmo di ordinamento quick sort

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
http://crosswordcheats.com