

C Reference Card (ANSI) - versione semplificata

corso Informatica B Politecnico di Milano

1 Struttura del programma

struttura del programma principale

```
int main(void) {  
    declarations  
    statements  
}
```

Dichiarazione di variabile: `type name;`

2 Comandi per il preprocessore

inclusione di un file di libreria `#include <filename>`

definizione di costante `#define name value`

3 Dichiarazioni

carattere (1 byte)	<code>char</code>
intero	<code>int</code>
short (16 bit integer)	<code>short</code>
long (32 bit integer)	<code>long</code>
numero senza segno	<code>unsigned</code>
costante	<code>type const name;</code>
numero reale	<code>float, double</code>
type enumerativo	<code>typedef enum name1, name2, ... typeName;</code>
struct	<code>typedef struct ... typeName;</code>
ridefinizione di tipo	<code>typedef oldTypeName newName;</code>
tipo array	<code>typedef elementType arrayType[dim];</code>
tipo matrice	<code>typedef elementType matrixType[dim1][dim2];</code>

4 Operatori

accesso al campo di una struct	<code>structName.field</code>
accesso all'i-esimo elemento di un array	<code>arrayName[i]</code>
operazioni aritmetiche per interi	<code>+, -, *, /, %, ++, --</code>
operazioni aritmetiche per reali	<code>+, -, *, /</code>
operatori di confronto	<code>==, !=, >, <, >=, <=</code>
operatore di assegnamento	<code>=</code>
operatori logici	<code>&&, , !</code>

5 Flusso di controllo

terminatore di istruzione	<code>;</code>
delimitatore blocco	<code>{ ... }</code>
istruzione if	<code>if (espr) istruzione1 else istruzione2</code>
istruzione while	<code>while (espr) istruzione</code>
istruzione for	<code>for (istr1; espr; istr3) istr2</code>
istruzione do	<code>do istruzione while (espr);</code>
istruzione switch	<code>switch (espr) {case val1: istr1; break; ... default istr}</code>

6 Libreria ANSI C

`<assert.h>` `<ctype.h>` `<errno.h>` `<float.h>` `<limits.h>` `<locale.h>`
`<math.h>` `<setjmp.h>` `<signal.h>` `<stdarg.h>` `<stddef.h>` `<stdio.h>`
`<stdlib.h>` `<string.h>` `<time.h>`

7 Input/Output <stdio.h>

stampa formattata `printf("formato", arg1, arg2, ...);`
acquisizione formattata `scanf("formato", arg1, arg2, ...);`

8 Operazioni tra stringhe <string.h>

lunghezza di s	<code>strlen(s)</code>
copia di s2 in s1	<code>strcpy(s1, s2)</code>
concatenazione di s2 dopo s1	<code>strcat(s1, s2)</code>
confronto tra s1 e s2	<code>strcmp(s1, s2)</code> - restituisce 0 se uguali, < 0 se s1 precede s2, > 0 altrimenti

9 Mathematical Functions <math.h>

funzioni trigonometriche	<code>sin(x), cos(x), tan(x)</code>
funzioni trigonometriche inverse	<code>asin(x), acos(x), atan(x)</code>
arctan(y/x)	<code>atan2(y,x)</code>
funzioni trigonometriche iperboliche	<code>sinh(x), cosh(x), tanh(x)</code>
esponenziali & logaritmi	<code>exp(x), log(x), log10(x)</code>
elevamento a potenza	<code>pow(x,y), sqrt(x)</code>