



Nacionalni centar
za vanjsko vrednovanje
obrazovanja

Adesivo per l'identificazione

INCOLLARE ATTENTAMENTE

INF

INFORMATICA

Tabelle ausiliarie

INF T D

INF.35.IT.R.T1.04



33264



12

Tabelle ausiliarie

Tabella 1.
Estratto dalla tabella ASCII (ISO-7)

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|----|---|----|---|---|---|---|---|
| 0 | | | SP | 0 | Ž | P | ž | p |
| 1 | | | ! | 1 | A | Q | a | q |
| 2 | | | “ | 2 | B | R | b | r |
| 3 | | | # | 3 | C | S | c | s |
| 4 | | | \$ | 4 | D | T | d | t |
| 5 | | | % | 5 | E | U | e | u |
| 6 | | | & | 6 | F | V | f | v |
| 7 | | | ‘ | 7 | G | W | g | w |
| 8 | | | (| 8 | H | X | h | x |
| 9 | | |) | 9 | I | Y | i | y |
| A | LF | | * | : | J | Z | j | z |
| B | | | + | ; | K | Š | k | š |
| C | CR | | , | < | L | Đ | l | đ |
| D | | | - | = | M | Ć | m | ć |
| E | | | . | > | N | Č | n | č |
| F | | | / | ? | O | | o | |

La coordinata della colonna corrisponde al carattere esadecimale del nibble (4 bit) sinistro del byte, mentre la coordinata della riga corrisponde al carattere esadecimale del nibble (4 bit) destro del byte.

Esempio:

20_{16} = 00100000 codice per *SP* (ingl. *space*), spazio nel testo

$0A_{16}$ = 00001010 codice per *LF* (ingl. *line feed*), vai a capo riga

$0C_{16}$ = 00001100 codice per *CR* (ingl. *carriage return*), inizio di riga

41_{16} = 01000001 codice per *A* (*A* maiuscola)

$6B_{16}$ = 01101011 codice per *k* (*k* minuscola).

Tabella 2.
Operatori aritmetici

| Descrizione | Pseudocodice | Pascal | C/C++ | Python |
|--------------------------------|--------------|--------|-------|--------|
| Somma | + | + | + | + |
| Sottrazione | - | - | - | - |
| Moltiplicazione | * | * | * | * |
| Divisione | / | / | / | / |
| Divisione a quoziente intero | <u>div</u> | div | / | // |
| Resto della divisione (modulo) | <u>mod</u> | mod | % | % |



Informatica

Tabelle ausiliarie

Tabella 3.
Operatori logici

| Descrizione | Pseudocodice | Pascal | C/C++ | Python |
|---------------------|--------------|--------|-------|--------|
| Invertitore | NE | not | ! | not |
| Congiunzione logica | I | and | && | and |
| Disgiunzione logica | ILI | or | | or |

Tabella 4.
Operatori di relazione

| Descrizione | Pseudocodice | Pascal | C/C++ | Python |
|---------------------|--------------|--------|-------|--------|
| Minore | < | < | < | < |
| Minore o uguale a | <= | <= | <= | <= |
| Maggiore | > | > | > | > |
| Maggiore o uguale a | >= | >= | >= | >= |
| Uguale | = | = | == | == |
| Diverso | <> | <> | != | != |

Tabella 5.
Funzioni predefinite

| Descrizione | Pseudocodice | Pascal | C/C++ | Python |
|--|--------------|-----------|------------|-----------|
| Valore assoluto di x | abs (x) | abs (x) | abs (x) | abs (x) |
| Quadrato del valore di x | sqr (x) | sqr (x) | pow (x, 2) | x ** 2 |
| Radice quadrata di x | sqrt (x) | sqrt (x) | sqrt (x) | sqrt (x) |
| Arrotondamento al valore intero più vicino a x | round (x) | round (x) | round (x) | round (x) |
| Parte intera del numero x | trunc (x) | trunc (x) | trunc (x) | trunc (x) |

Tabella 6.
Priorità degli operatori

| Nr. progressivo | Operatori |
|-----------------|-----------------------------------|
| 1. | () |
| 2. | NE |
| 3. | *, /, <u>div</u> , <u>mod</u> , I |
| 4. | +, -, ILI |
| 5. | <, >, <=, >=, <>, = |



Tabella 7.
Istruzioni elementari

| Descrizione | Pseudocodice | Pascal | C/C++ | Python |
|--|--|---|---|---|
| Blocchi di istruzioni | { } } | begin end | { } | rientro |
| Letture | <u>leggi</u> | read | scanf | input |
| Scrittura | <u>stampa</u> | write | printf | print |
| Assegnazione | <u>:</u> = | <u>:</u> = | = | = |
| Selezione | se <u>condizione</u> allora istruzione1 oppure istruzione2; | if condizione then istruzione1 else istruzione2; | if (condizione) istruzione1; else istruzione2; | if condizione: istruzione1 else: istruzione2 |
| Iterazione con successione finita di istruzioni | per b := p <u>fino a</u> k <u>esegui</u> istruzione; | for b := p to k do istruzione; | for (b = p; b <= k; b++) istruzione; | for i in range(p, k + 1): istruzione |
| Ciclo infinito: condizione di inizio sempre vera | <u>finchè</u> condizione periste <u>esegui</u> istruzione; | while condizione do istruzione; | while (condizione) istruzione; | while condizione: istruzione |

