

Esercizi su

Visibilità e tempo di vita

Operatore condizionale

Caratteri

Numeri reali

Enumerati

Conversioni e limiti

Visibilità e tempo di vita

- *mostra_visibilita.cc*

Operatore condizionale

- *oper_cond.cc*
- Per casa:
 - *oper_cond2.cc*
 - Esempio di soluzione molto utilizzata nelle prove di programmazione

Input/output caratteri

- Sintassi stampa di una variabile/costante *char* su *stdout*
 - Come per gli *int*
- Sintassi lettura di un carattere da *stdin*
 - Come per gli *int*
 - Attenzione, con il tipo **char** si può leggere anche il carattere *newline* (come vedremo)

- *leggi_stampa_char.cc*

- Conversione codice -> carattere
 - *codice_car.cc*

- Conversione carattere -> codice
 - *car_codice.cc*

- Supponendo che sullo *stdin* ci sia il carattere '1'
 - Che differenza c'è tra il contenuto delle variabili *char c* e *int i*, dopo
cin >> c ;
oppure
cin >> i ;

Domanda

- Nel primo caso nella variabile finisce il codice del carattere '1', nell'altro caso vi finisce il valore 1

- Siete riusciti a leggere/stampare il codice del carattere spazio?
- E del carattere *newline*?
- Se no, come mai?

- Perché, come sappiamo, lo spazio ed il *newline* (insieme ad altri caratteri) sono considerati *spazi bianchi*
 - e l'oggetto *cin* è configurato per saltare gli spazi bianchi

Manipolatore noskipws

- Per configurare l'oggetto *cin* per non saltare gli spazi bianchi bisogna passargli il manipolatore **noskipws**
- Viceversa, per tornare a saltare gli spazi bianchi si utilizza il manipolatore **skipws**

- Un programma che legge e stampa i codici di tutti i caratteri letti in ingresso finché non viene immesso il carattere – è il seguente:

Esempio

```
main()
{
    char carattere;

    do {
        cout<<"Inserisci un carattere " ;
        cin>>noskipws>>carattere ;
        int i = static_cast<int>(carattere) ;
        cout<<"\nIl codice ASCII del carattere e' "
            <<i<<endl ;
    } while(carattere != '-') ;
}
```

Domande 1/2

- Il programma stampa anche il codice ASCII del carattere *newline*?
 - Cosa stampa se si preme invio senza immettere nessun (altro) carattere?
- Se si preme un tasto diverso da invio e corrispondente ad un carattere stampabile, il programma riesce a leggerlo (e quindi a ristamparlo) anche se non si preme invio dopo di tale tasto?

Domande 2/2

- Come mai la risposta all'ultima domanda è no?

Modalità terminale 1/2

- La risposta alla domanda è legata al comportamento del terminale
- Un terminale può essere configurato per funzionare in modi molto diversi tra loro
- La configurazione utilizzata tipicamente è quella cosiddetta **canonica**
 - In tale configurazione il terminale manda i caratteri allo *stdin* del programma solo dopo che è stato premuto invio

Modalità terminale 2/2

- Si può però configurare il terminale in modo **non canonico** e farsi inviare ogni carattere non appena viene immesso dall'utente
- Non entreremo in ulteriori dettagli tecnici
 - Non hanno nessuna rilevanza per il corso e l'esame
 - Servirebbero conoscenze che ancora non abbiamo
 - Per gli interessati, un punto di partenza può essere **man termios**

- Svolgere l'esercizio
traccia_car_codici_immediato.cc
 - La soluzione è nel file
sol_car_codici_immediato.cc

- *leggi_inc_stampa_char.cc*

Esercizio per casa

- Stampa tabella codici ASCII
 - *tabella_ascii.cc*

The father of the ASCII table

- "We had over 60 different ways to represent characters in computers.

It was a real Tower of Babel," says Bob Bemer, who was instrumental in ASCII's development and is widely known as

"the father of ASCII."

Bob Bemer



Bob Bemer

*E la targa
della sua
auto ...*



- ANSI (American National Standards Institute)
- ISO (International Organization for Standardization)
 - ISO 8859-1 Latin-1:
Estensione della codifica
ASCII

- Codici carattere multilingue
 - TRON Character Code
 - Unicode e ISO 10646-1:1993
 - Struttura rigida e imposizioni sulle lingue orientali
 - UNIX e Mule

- QBPMP
- HAL
- Crittografia ai tempi dei romani ...

Prova di programmazione: RSA

- Traccia in *traccia_RSA_int_char.txt*
 - Solo menù *int*: soluzione in *RSA_solo_menu.cc*
 - Soluzione in *RSA_intero.cc*
 - Soluzione in *RSA_carattere.cc*
- E' molto molto importante che lo svolgiate per intero (tranne eventualmente la parte di generazione delle chiavi) a casa

- Come prova di programmazione, svolgere tutto, tranne la generazione delle chiavi
- Tempo massimo per svolgere la prova: 1h30min
- Infine, vi lancio le seguenti due sfide
 - Per ciascuna sfida, vincerà chi per primo invierà la risposta
 - Nominativo comunicato poi a lezione

Prima sfida

- Decifrate il seguente numero criptato con la chiave (5, 173, 323):
 - 178
- Per dimostrare di esserci veramente riusciti bisogna fornire sia la risposta che il programma utilizzato per ottenerla

Seconda sfida

- Decifrate il carattere il cui codice criptato, con la stessa chiave utilizzata per il precedente numero, è 53
- Per dimostrare di esserci veramente riusciti bisogna fornire sia la risposta che il programma utilizzato per ottenerla

Mini-prova scritta

- *traccia-mini-scritto_23Nov16*
 - Verrà messa online la prossima settimana

- *stampa_enum.cc*
- Per casa
 - *giorni_lavoro.cc*

- *divis_reale.cc*
- *reale_int.cc*
- *divis_reale2.cc*
- *ascensore.cc*

Uscita forzata

- Per terminare forzatamente l'esecuzione di un programma, ovunque ci si trovi nel codice
- Funzione
void exit(int n) ;
- Il valore passato come parametro attuale corrisponde al valore che sarà ritornato a chi ha fatto partire il programma

Tipi e conversioni

- *limiti.cc*

Variabili e tipo 1/2

- Dato il seguente frammento di codice:

```
int i ;  
double d = 2.3 ;  
i = d ;
```

- Qual è il valore della variabile *i* dopo l'assegnamento?
- Scrivere un programma che stampi il valore di *i*

Variabili e tipo 2/2

- Morale: *i* e' di tipo intero e, come ogni variabile in C/C++, non cambia tipo

Assegnamento 1/2

- Dato il seguente frammento di codice:

```
int i = 5 ;  
int j = 2 ;  
double d = i / j ;
```

- A quale valore è inizializzata la variabile *d*?
- Scrivere un programma che stampi il valore di *d*

Assegnamento 2/2

- Anche se d e' di tipo *double*, la divisione tra i e j e' di tipo intero
- Il tipo del risultato di un assegnamento non influenza il tipo dell'espressione a destra dell'assegnamento,
 - né degli operandi/operatori in tale espressione

Compiti per casa: caratteri

- *codice_car2.cc*
- *tabella_ascii.cc*
- *funzioni_car.cc*
- Algoritmo di generazione chiavi RSA (traccia contenuta in *traccia_RSA_int_char.txt*)
 - Può essere un po' impegnativo

Compiti per casa: reali

- *funzioni_reali.cc*
- *numero_reale.cc*
- *radici.cc*
- *numero_razionale.cc*
- *newton_iterativo.cc*

Compiti per casa: conversioni

- *divis_reale3.cc*
- *int_reale_int.cc*