

Esercizi su

---

Memoria dinamica  
IO non formattato

- E' arrivato il momento di commentare le modalità d'esame ...
- Per potersi iscrivere all'esame è necessario compilare il questionario di valutazione della didattica

# Esercizi sullo heap

---

- *crea\_riempi\_distruggi\_array.cc*
- *percorso\_semplificato.cc*
  - Soluzione intermedia:  
*percorso\_semplificato\_solo\_main.cc*
- *percorso.cc*

- Come si scrive sullo *stderr*?

*cerr*<<...

# Esercizio su gestione memoria

---

- *pila.cc*
  - Le funzioni riportate nella soluzione fanno riferimento ad un solo oggetto globale?
  - Oppure permettono di fatto di operare con qualsiasi pila?
  - Nel secondo caso, si è definito un **tipo di dato** pila

# Altre domande prove scritte

---

- Con le vostre attuali conoscenze potete ora rispondere ad ulteriori domande, riportate di sotto, delle prove scritte precedentemente suggerite
  - 17 Dicembre 2008, anche 11
    - Ossia tranne 3, 5 e 6
  - 10 Febbraio 2009, anche 3
    - Ossia tranne 1, 9
  - 30 Giugno 2009, svolgendo questa volta la 11 in modo corretto
  - 22 Settembre 2009, anche 11

# Prove scritte

---

- Potete poi svolgere le seguenti prove scritte, a meno delle domande riportate a fianco
  - 11 Giugno 2008
  - 2 Luglio 2008, tranne 2
  - 2 Settembre 2008, tranne 4

# Prove di programmazione

---

- 3 Settembre 2008
- 19 Dicembre 2008
- 11 Febbraio 2009
- 10 Giugno 2009



# Compiti per casa 1/2

---

- Fare pila di *struct* con due campi, uno di tipo *int* ed uno di tipo *stringa*
- Farne poi una versione con inserimento ordinato ed una con ordinamento a posteriori
  - In base alla *stringa* ed al valore numerico
  - Eventualmente riguardare la soluzione di *lista esami ordinata*

# Esercizi sui file binari

---

- *file/conta\_linee.cc*
- *file/scrivi\_leggi\_array.cc*

# Memorizzazione informazione

---

- Un file può contenere testo, immagini, audio, ...
- Ma in generale, cos'è?
- Proviamo a scoprirlo ...

- Compilare ed eseguire il programma  
*file/scrivi\_leggi\_array.cc*
  - Se l'avete con voi, utilizzate pure la vostra soluzione
- Il programma memorizza l'array in due file, uno di testo (*dati.txt*) e l'altro binario (*dati.dat*)

# Stampa di un file 1/2

---

- Visualizzazione di un file
  - *cat*
  - *more, less* (per uscire: 'q')
- Visualizziamo entrambi i file con un editor di testo, o col comando *cat* oppure *less*

# Stampa di un file 2/2

---

- Visualizzatore esadecimale: *hd*, *dump*, *hexdump*
- *hd nomefile*
- Proviamo ad utilizzarlo per visualizzare di nuovo entrambi i file

# Struttura di un file

---

- Come sono memorizzate le informazioni?
  - Mediante rappresentazioni numeriche
- Un file è semplicemente:
  - una sequenza di numeri,
  - a cui è associato un nome

# Struttura di un file di testo

---

- Anche un file di testo non è altro che una sequenza di numeri
- Ciascun numero rappresenta il codice ASCII di un carattere



# Altro esempio di file binario

---

- Il file *.bmp* contengono immagini sotto forma di sequenze di bit
  - Nel caso di immagine in bianco e nero, un bit a 1 può rappresentare per esempio un pixel nero, mentre un bit a 0 un pixel bianco

# Tipo di un file e nome del file

---

- E' importante capire che il nome di un file, tipicamente il suo suffisso, ci da solo un suggerimento su come vanno interpretati i byte contenuti nel file
- L'effettivo tipo del file dipende dal modo in cui vi sono state inserite le informazioni all'atto della scrittura

# Esercizi sui file binari

---

- *file\_binario.cc*
  - Tentare di visualizzare il file creato dal programma
    - Provare anche con *hd* o *dump*
- Per casa
  - *copia\_car\_num.cc*

# Altri esercizi sui file

---

- *lunghezza\_file.cc*
- *pila\_file.cc*

# Altri compiti per casa

---

- Scrivere versioni con lettura/scrittura da/su file degli esercizi con inserimento e stampa di array
  - Versione testuale
  - Versione binaria
- Assicurarsi di aver capito **inserimento ordinato** e **riordinamento** di un array