

Corso di Laurea in Matematica

Insegnamento di: Laboratorio di Programmazione, gr.2, prof. **Simoncelli**

MODALITA' DI ESAME E DI PRENOTAZIONE per l' anno acc.co 2012/2013

A) MODALITA' DI ESAME per l'anno accademico 2012/2013, valide per tutte le sessioni

L'esame si articola in due parti :

1) **Verifica in laboratorio**: prova individuale, svolta presso il Laboratorio di Calcolo del D.M.A.. Si richiede di scrivere, compilare ed eseguire un semplice programma assegnato e di consegnare i risultati. **Il superamento di questa prova è condizione per l'accesso all'esame orale.**

N.B.: Dalla prova di laboratorio sono **esonerati** gli studenti che hanno **frequentato assiduamente** le esercitazioni in laboratorio, **superando la verifica individuale durante tali esercitazioni.**

L' elenco nominativo di tali studenti è pubblicato sulla [pagina web del docente](#).

2) **Esame orale**. Consiste nella verifica della conoscenza di tutti gli argomenti trattati durante le lezioni, partendo dalla discussione degli elaborati consegnati all'atto della prenotazione. Il programma svolto è disponibile sulla [pagina web del docente](#)

La prenotazione d'esame va fatta per posta elettronica (oppure di persona presso il docente) entro la data stabilita nell' avviso. Se nell'avviso non è fissata una scadenza, la prenotazione va effettuata almeno una settimana prima della data fissata per l'orale. Alla prenotazione vanno allegati gli elaborati da discutere, preparati seguendo rigorosamente le modalità previste (regole di impaginazione ed elenco delle tracce sono più avanti). Gli studenti in debito della prova di laboratorio debbono segnalarlo al docente nella loro e-mail di prenotazione.

Le date per le sedute di esami orali: sono comunicate di volta in volta con avvisi specifici pubblicati sulla pagina web. Per la sessione estiva dell'anno acc.co 2012/2013 le date sono le seguenti (salvo eventuali piccoli spostamenti dell'ultimo momento, se inevitabili)

- Inizio 28 maggio 2013 (prenotazione entro il 25 maggio ore 24)
- Inizio 11 giugno 2013 (prenotazione entro il 4 giugno ore 24)
- Inizio 25 giugno 2013 (prenotazione entro il 18 giugno ore 24)
- Inizio 9 luglio 2013 (prenotazione entro il 2 luglio ore 24)
- Inizio 19 luglio 2013 (prenotazione entro il 12 luglio ore 24)
- Inizio 20 settembre 2013 (prenotazione entro il 13 settembre ore 24)

B) modalità per la PRENOTAZIONE dell'esame orale

Gli studenti intenzionati a sostenere l'esame in una specifica seduta debbono prenotarsi segnalandolo per posta elettronica al docente (all'indirizzo simoncel@unina.it) entro la data di scadenza prevista, allegando alla prenotazione il loro elaborato, consistente nello svolgimento degli esercizi che compaiono nell'elenco specifico per la seduta prescelta. Gli studenti impossibilitati a prenotarsi per posta elettronica possono prenotarsi di persona, contattando il docente durante l'orario di ricevimento e consegnando direttamente gli elaborati, in forma cartacea e/o digitale.

MOLTO IMPORTANTE: ELABORATI CHE NON RISPETTINO RIGOROSAMENTE LE REGOLE FORMALI FISSATE IN QUESTA CIRCOLARE NON VERRANNO ACCETTATI (E VERRANNO VALUTATI ZERO AI FINI DEL CALCOLO DEL VOTO D'ESAME). In caso di dubbi (anche relativamente ai testi degli esercizi) chiedere tempestivamente chiarimenti al docente, anche per posta elettronica.

C) MODALITA' DI PREPARAZIONE DELL' ELABORATO, ELENCO DELLE TRACCE

Fare riferimento all' **elenco specifico per il mese** in cui si intende sostenere l'esame .

Per ogni esercizio nell'elenco, consegnare, in allegato al messaggio di prenotazione, una directory/cartella chiamata "CognomeEsercizio" ad es. SimoncelliM1, SimoncelliM3...) contenente esclusivamente i seguenti files :

- 1) il programma sorgente in linguaggio Fortran (main + procedure, in uno o più files di tipo ASCII puro con estensione *.f90*)
- 2) un file in cui sia stata salvata (senza apportare alcuna modifica !) la sessione di lavoro durante la quale il programma è stato compilato con successo, linkato, ed eseguito. (N.B. questo file deve essere di tipo *.txt*, *.doc*, o *.pdf*, NON deve essere una immagine e NON deve avere uno sfondo scuro. Se non sapete come salvare la sessione di lavoro chiedete consiglio al docente, anche per *e-mail*)
- 3) eventuali files di dati contenenti input per il programma e/o output di questo, se richiesti dal testo dell'esercizio.

N.B. : Non consegnare files contenenti programmi eseguibili o rilocabili !

N.N.B. (MOLTO IMPORTANTE, OBBLIGATORIO !)

- 1) Inserire cognome, nome, numero di matricola dello studente nelle liste Fortran (anche nelle *subroutines* e nelle *functions* se sono in files separati) sotto forma di commenti.
- 2) Inserire nel programma opportune frasi di scrittura per fare in modo che nome cognome e numero di matricola dello studente appaiano anche nell'output del programma.
- 3) Inserire nel codice Fortran e nell'output del programma anche il numero con cui il testo dell'esercizio compare nell'elenco, ed una breve descrizione dello scopo del programma (max. 5 parole).

ESEMPIO:

```
PROGRAM binario
```

```
!
```

```
!           Studente: Alba C. Simoncelli, matricola N87/xxx
```

```
!           Esercizio  G1 – Cambiamento di base
```

```
!
```

```
WRITE (*,*) 'Simoncelli, Alba Chiara N87/XXX'
```

```
WRITE (*,*) 'Esercizio G1 – cambiamento di base'
```

=====

ELENCO DELLE TRACCE PER LA SEDUTA DI: MAGGIO 2013

M1) Come è noto, la successione di Fibonacci $(a_i)_{i \in \mathbb{N}}$ è definita dalle seguenti relazioni: (1) $a_0 = a_1 = 1$ e (2) $a_i = a_{i-2} + a_{i-1} \quad \forall i > 1$.

Scrivere in Fortran95, compilare ed eseguire un programma che prenda in input (da tastiera o da file) un valore intero *nn* compreso tra 20 e 30 e **costruisca un array i cui elementi sono i primi *nn* numeri di Fibonacci**, quindi scriva gli elementi di tale array in un **file su disco** (chiamato *Fibonacci*) nell'ordine in cui sono stati costruiti. Successivamente, il programma chieda in input da tastiera un valore *val* quindi **controlli** (e segnali in output) se il numero *val* è presente o meno tra i primi *nn* numeri di Fibonacci.

Per effettuare il controllo utilizzare una subroutine o (meglio) una function che implementi un **algoritmo di searching** (quello che si ritiene più idoneo allo scopo). Si esegua una sessione di lavoro in cui il programma viene eseguito almeno due volte: la prima fornendo come valore di *val* il numero 21, la seconda fornendo il valore 22.

M2) Scrivere in Fortran95, compilare ed eseguire un programma che costruisca e fornisca in output sullo schermo le seguenti **matrici** : (1) la matrice A (con dieci righe e sei colonne) i cui elementi sono dati dalla formula $a_{ij} = 10i + j$, (2) la matrice B (con sei righe e dodici colonne) per la quale $b_{ij} = i$, per ogni i e per ogni j. Dopo aver costruito le matrici A e B, il programma calcola e fornisce in output anche la matrice $C = A \times B$ (**prodotto** righe per colonne) e la matrice C_t , **trasposta** di C. In fine, scriva in un unico file su disco (chiamato *matrici*) le tre matrici A, B, C, ed in un altro file (chiamato *trasposta*) la matrice C_t , facendo precedere gli elementi di ciascuna matrice da un record contenente il numero di righe ed il numero di colonne della matrice stessa. Per il calcolo del prodotto di matrici e della trasposta di una matrice si scrivano e si utilizzino opportune subroutines Fortran.

M3 (FACOLTATIVO) Scrivere in Fortran95, compilare ed eseguire un programma che, letto da un file su disco il cui nome è fornito dall'utilizzatore (NON dal programmatore !) un array di dati numerici non ordinato (almeno cinquanta elementi) permetta all'utilizzatore di **scegliere l'algoritmo di ordinamento** da utilizzare per disporre in ordine crescente i dati letti, e richiami la subroutine corrispondente all'algoritmo prescelto. Il programma dovrà anche scrivere i dati ordinati in un diverso file su disco (chiamato *fileordinato.dat*) nel primo record del quale verrà scritto il numero di elementi dell'array appena ordinato.

TRACCE PER LE SEDUTE CHE CADONO NEL MESE DI: GIUGNO 2013

G1) Scrivere in Fortran95, compilare ed eseguire un programma che prenda in input, da un file su disco il cui nome abbia estensione ".num", cinque o più numeri positivi non interi rappresentati in base 10, e per ciascuno di essi: (1) ne calcoli e fornisca in output sullo schermo la **rappresentazione in base due**, (2) calcoli la rappresentazione in base dieci del numero in base due appena ottenuto e la **confronti** con il numero di partenza, per verificare la correttezza dei calcoli di conversione effettuati.

Le conversioni dalla rappresentazione in base dieci a quella in base due e viceversa debbono essere effettuate per mezzo di *subroutines* e/o *functions*.

Eseguire il programma due volte, per convertire due diversi set di dati nei quali siano presenti, insieme ad altri, i seguenti numeri : 4.125 ; 7.875 ; 1024.0 ; 0.0625 .

G2) Scrivere in Fortran95, compilare ed eseguire un programma che legga da due diversi files su disco (chiamati *mat1* e *mat2*) le due matrici M1 ed M2, quindi calcoli e fornisca in output sullo schermo : (1) il valore del più grande elemento di M1 ed il valore del più piccolo elemento di M2; (2) la matrice (**prodotto** righe per colonne) $P = M1 \times M2$ e (3) la matrice T, **trasposta** di P. Per il calcolo del massimo e del minimo si scrivano e si utilizzino opportune *functions*; per il prodotto di matrici e per il calcolo della trasposta di una matrice si scrivano e si utilizzino opportune *subroutines*.

(N.B. Ciascuna matrice in input ha almeno trenta elementi, e le sue dimensioni sono indicate nel primo record del rispettivo file, cioè ciascuna delle matrici in input è scritta nel rispettivo file secondo lo schema seguente: numero.di.righe numero.di.colonne **EOR** elementi della prima riga **EOR** elementi.della.seconda.riga **EOR** **EOR** elementi.della.ultima.riga **EOF**).

G3 (FACOLTATIVO) Scrivere in Fortran95, compilare ed eseguire un programma che, letti da due files su disco il cui nome è fornito dall'utilizzatore (NON dal programmatore !) due arrays di dati numerici non ordinati (ciascuno con almeno venti elementi), li **ordini** e quindi li **fonda** in un unico file su disco, ordinato in ordine non decrescente, il cui nome è *filefuso.dat*. Scrivere ed utilizzare una *subroutine* che effettui l'ordinamento con il metodo preferito.

TRACCE PER LE SEDUTE CHE CADONO NEL MESE DI: LUGLIO 2013

L1) Scrivere in Fortran95, compilare ed eseguire un programma che, letti da un file su disco chiamato *dati.inp* un insieme di valori numerici non interi (almeno trenta) dei quali né il programmatore né l'utilizzatore conoscono a priori la numerosità, calcoli e fornisca in output sullo schermo **il più grande** ed **il più piccolo** dei valori letti, nonché la **media aritmetica** e la **varianza** del campione costituito da tali valori. Successivamente il programma riscriva i dati su disco nell'ordine inverso a quello in cui sono stati letti, in un file chiamato *dati.out*, che conterrà nel primo record il numero di tali valori.

Per il calcolo della media e della varianza si utilizzino le seguenti formule, in cui $\mathbf{a} = [a_1 \dots a_n]$ rappresenta il campione considerato: $m(\mathbf{a}) = [\sum_1^n a_i] / n$; $\text{var}(\mathbf{a}) = [\sum_1^n (a_i)^2] / n - [m(\mathbf{a})]^2$.

Si scrivano e si utilizzino *functions* o *subroutines* almeno per il calcolo del massimo, del minimo, e della somma degli elementi di un vettore.

L2) Scrivere in Fortran95, compilare ed eseguire un programma che verifichi se una matrice quadrata A letta da un file su disco (chiamato *mat.inp*) è **simmetrica**, senza fare confronti diretti tra gli elementi della matrice.

Il programma può, ad esempio, calcolare la trasposta della matrice A per mezzo di una opportuna subroutine, quindi calcolare $C = A - A^t$ e verificare se C è la matrice nulla.

Per effettuare quest'ultima verifica, si potrà controllare ogni elemento della matrice, oppure, ad esempio, calcolare il massimo tra gli elementi della matrice C (c_{max}) ed il minimo tra gli elementi della stessa matrice (c_{min}) per mezzo di opportune *functions* e poi verificare se $c_{max} = c_{min} = 0$ (il che avviene se e solo se la matrice di partenza A è simmetrica).

Verificare il funzionamento del programma eseguendolo su opportune matrici quadrate di dimensione maggiore o uguale di sei. Nella sessione di lavoro si esegua il programma almeno due volte: una volta su una matrice simmetrica ed una volta su una matrice non simmetrica.

L3 (FACOLTATIVO) Scrivere in Fortran95, compilare ed eseguire un programma che, dopo aver letto da un file su disco un array \mathbf{a} di dati numerici (del quale si sa che: (1) gli elementi di \mathbf{a} sono almeno quaranta, (2) gli elementi di \mathbf{a} non sono ordinati; (3) i valori 10.14 e 85.85 sono presenti nell'array \mathbf{a}) e dopo aver ricevuto in input da tastiera un valore numerico x dello stesso tipo degli elementi di \mathbf{a} , verifichi se il valore x è presente nell'array \mathbf{a} , utilizzando l'algoritmo di **ricerca binaria**. Utilizzare una *subroutine* per effettuare l'ordinamento di \mathbf{a} con il metodo preferito.

TRACCE PER LE SEDUTE CHE CADONO NEL MESE DI: SETTEMBRE 2013

S1) Si consideri successione $(a_i)_{i \in \mathbb{N}}$ definita dalle relazioni: (1) $a_1 = a_2 = 1$ (2) $a_i = a_{i-2} + a_{i-1} \forall i > 2$. Scrivere in Fortran95, compilare ed eseguire un programma che **costruisca un array i cui elementi sono i primi trenta elementi della successione**, e li scriva in ordine decrescente in un **file su disco**, chiamato *farray*; poi chieda in input da tastiera un numero intero *vinc* e **controlli** (segnalandolo in output allo schermo) se il numero *vinc* è presente o meno nell'array appena costruito.

Per effettuare il controllo utilizzare una *subroutine* o (meglio) una *function* Fortran che implementi un **algoritmo di searching** (quello che si ritiene più idoneo allo scopo). Si esegua una sessione di lavoro in cui il programma viene eseguito almeno due volte: la prima fornendo come valore di *vinc* il numero 54, la seconda fornendo il valore 55.

S2) Scrivere in Fortran95, compilare ed eseguire un programma che legga una matrice quadrata A da un file su disco (chiamato *quadrata*) in cui la matrice è scritta per righe e che contiene nel primo record la dimensione della matrice *dima*. Il programma calcoli la trasposta di A, A^t , e calcoli poi il prodotto (righe per colonne) $P = A \times A^t$. Successivamente verifichi se P è o meno la matrice identica di ordine *dima* (cioè se : $p_{i,j} = 1$ per $i = j$ e $p_{i,j} = 0$ altrimenti). Si scrivano e si utilizzino *subroutines* per il calcolo della matrice trasposta e per il calcolo del prodotto di matrici. Verificare il funzionamento del programma eseguendolo su opportune matrici quadrate di dimensione due e tre. Nella sessione di lavoro da consegnare si esegua il programma almeno due volte: una volta sulla matrice A1 definita da: $a_{i,j} = 0$ se $i = j$; $a_{i,j} = 1$ altrimenti , per $i,j = 1,2$; ed una volta sulla matrice A2 definita da: $a_{i,j} = 1$ se $i = j$; $a_{i,j} = 0$ altrimenti , per $i,j = 1,2$.

S3 (FACOLTATIVO) Scrivere in Fortran95, compilare ed eseguire un programma che, letto da un file su disco il cui nome è fornito dall'utilizzatore (NON dal programmatore !) un insieme non ordinato di numeri interi (almeno trenta elementi) permetta all'utilizzatore di **scegliere l'algoritmo di ordinamento** da utilizzare per disporre in ordine decrescente i dati letti, e richiami la *subroutine* corrispondente all'algoritmo prescelto, quindi fornisca i dati ordinati in ordine decrescente sullo schermo e li scriva in un file su disco chiamato *farray.ord*, ordinati in ordine crescente, memorizzando nel primo record del file il numero di elementi dell'array stesso.

Napoli , 24 maggio 2013

Il docente del corso : Alba C. Simoncelli

