Fondamenti di Programmazione – Prova di laboratorio
28-1-09 – AD – Andrea Sterbini

Svolgete le seguenti coppie di esercizi a seconda della postazione in cui vi trovate

Postazione	Es. 1	Es. 2		Post.	Es. 1	Es. 2
1, 17, 33	A	A		**9, 25, 41**	A	C
2, 18, 34	B	B		**10, 26, 42**	B	D
3, 19, 35	C	C		**11, 27, 43**	C	A
4, 20, 36	D	D		**12, 28, 44**	D	B
5, 21, 37	A	B		**13, 29, 45**	A	D
6, 22, 38	B	C		**14, 30, 46**	B	A
7, 23, 39	C	D		**15, 31, 47**	C	B
8, 24, 40	D	A		**16, 32, 48**	D	C

Esercizio 1 A

Si implementi l'algoritmo di **bubble-sort**. Si implementi il programma che legge una successione di numeri interi terminata dal valore -1 (al massimo 100 numeri), li ordina e li stampa ordinati.

Esercizio 1 B

Si implementi l'algoritmo di **insertion-sort**. Si implementi il programma che legge una successione di numeri interi terminata dal valore -1 (al massimo 100 numeri), li ordina e li stampa ordinati.

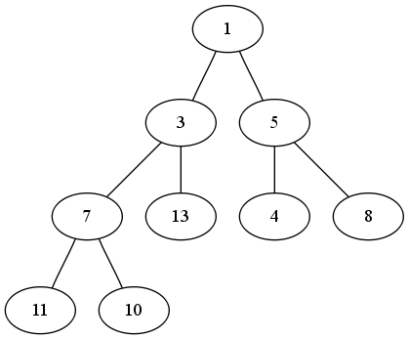
Esercizio 1 C

Si implementi l'algoritmo di **selection-sort**. Si implementi il programma che legge una successione di numeri interi terminata dal valore -1 (al massimo 100 numeri), li ordina e li stampa ordinati.

Esercizio 1 D

Si implementi l'algoritmo di **merge-sort**. Si implementi il programma che legge una successione di numeri interi terminata dal valore -1 (al massimo 100 numeri), li ordina e li stampa ordinati.

Esercizio 2 A



È possibile utilizzare un vettore per rappresentare un albero binario completo come segue:

Il figlio sinistro di un nodo con indice **i** è il nodo con indice **2i+1**; il figlio destro è il nodo con indice **2i+2**.

Esempio: il vettore contenente i numeri 1 3 5 7 13 4 8 11 10 corrisponde all'albero a fianco

Scrivere un programma che legge in input un intero n e quindi n interi, li memorizza in un vettore (che potete dichiarare staticamente di 100 elementi, o -meglio- allocare dinamicamente dopo la lettura del valore n), e quindi stampa i valori con una visita in **post-ordine** dell'albero.

Quindi in questo caso l'output dell'esempio sarebbe: 11 10 7 13 3 4 8 5 1

Non è necessario implementare gli alberi, usate direttamente il vettore.

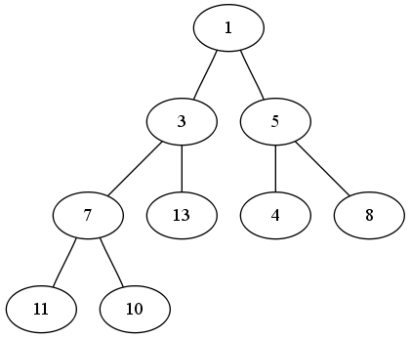
Esercizio 2 B

Scrivere un programma che legge una sequenza di interi non negativi, terminata da -1, e inserisce i valori letti in una **pila**.

Scrivere quindi una funzione che riceve come argomento la pila e un intero k e restituisce una nuova lista (potete usare una pila anche qui), i cui elementi hanno come valore la somma dei valori di k elementi successivi estratti dalla pila.

Gli elementi della pila originale devono essere correttamente deallocati.

Esercizio 2 C



È possibile utilizzare un vettore per rappresentare un albero binario completo come segue:

Il figlio sinistro di un nodo con indice **i** è il nodo con indice **2i+1**; il figlio destro è il nodo con indice **2i+2**.

Esempio: il vettore contenente i numeri 1 3 5 7 13 4 8 11 10 corrisponde all'albero a fianco

Scrivere un programma che legge in input un intero n e quindi n interi, li memorizza in un vettore (che potete dichiarare staticamente di 100 elementi, o -meglio- allocare dinamicamente dopo la lettura del valore n), e quindi stampa i valori con una visita in **pre-ordine** dell'albero.

Quindi in questo caso l'output dell'esempio sarebbe: 1 3 7 11 10 13 5 4 8

Non è necessario implementare gli alberi, usate direttamente il vettore.

Esercizio 2 D

Scrivere un programma che legge una sequenza di interi non negativi, terminata da -1, e inserisce i valori letti in una **coda**.

Scrivere quindi una funzione che riceve come argomento la coda e un intero k e restituisce una nuova lista (potete usare una coda anche qui), i cui elementi hanno come valore la somma dei valori di k elementi successivi estratti dalla coda.

Gli elementi della coda originale devono essere correttamente deallocati.

Pomeriggio

Esercizio 1

L'algoritmo di mergesort si basa sulla fusione di 2 liste ordinate.

Realizzate una versione del passo di merge che fonde 3 liste (o vettori) ordinate (chiamate la funzione merge3)

Realizzate quindi, usando questa versione di merge, un mergesort ricorsivo che suddivide le liste/vettori disordinati in 3 parti invece che 2.

Realizzate un programma che legge una sequenza di interi terminata da -1 di al massimo 100 elementi, la ordina e la stampa.

Esercizio 2

Si scriva una funzione che riceve una lista di parole ed elimina (disallocandoli opportunamente) tutti i nodi che NON contengono parole palindrome.

Si scriva un programma che legge una sequenza di parole (da leggere con %s) terminata dalla parola “STOP”, costruisce la lista, ne elimina i non palindromi con la funzione precedente, e stampa gli elementi rimasti.