

Corso di Laurea in Ingegneria Informatica (NOD)
Fondamenti di Informatica B
Prof. Maurizio Vincini

Compito Scritto - 07/09/2016

ESERCIZIO 1 (7 punti)

Dato il seguente listato:

```
#include <stdio.h>
#include <stdlib.h>

void main() {
    int a=3, *b, d=4;

    b = &a; *b = d;
    d++; a--;

    printf("%d %d %d\n", a, *b, d);
    {
        int b, d=1;

        b=(a+d)/2;
        a=(b++)*(++d);
        printf("%d %d %d\n", a, b, d);
    }
    printf("%d %d %d\n", a*2, (*b)+1, d-1);
    {
        d=(a>5)*3;
        printf("%d %d %d\n", a, *b, d);
        {
            int *d=b;
            a=7;
            printf("%d %d %d\n", a+3, *b, (*d)-3);
        }
        printf("%d %d %d\n", a+3, *b, (d)-3);
    }
}
```

Scrivere la corretta sequenza della stampa a video e mostrare lo sviluppo dei record di attivazione ignorando le chiamate alle funzioni di I/O.

ESERCIZIO 2 (9 punti)

Il Prof. Vincini dispone di un archivio di esercizi d'esame, costruito in quindici anni di appelli.

Ogni anno si sono svolti 6 appelli, ed ogni appello contiene 4 diverse domande, ciascuna riferita ad un argomento d'esame.

Ogni gruppo di domande è numerata in modo progressivo da 1 a N, ed ogni compito è costituito esattamente da 4 domande, una per ciascuna tipologia.

Scrivere una procedura che mostri a video tutti le possibili prove d'esame distinte che si possono costruire inserendo una domanda per ciascuna delle 4 categorie.

ESERCIZIO 3 (8 punti)

Data una lista dinamica di interi ed un valore indicato da utente, scrivere una procedura che elimini la prima coppia di elementi adiacenti la cui somma è pari al valore indicato dall'utente.

ESERCIZIO 4 (9 punti)

Dato un albero binario contenente valori interi, scrivere la funzione che esplori l'albero e mostri, per ogni foglia, il valore medio dei valori nel cammino per raggiungerla.

ESERCIZIO 3 (8 punti)

Data una lista dinamica di interi ed un valore indicato da utente, scrivere una procedura che elimina la prima coppia di elementi adiacenti la cui somma è pari al valore indicato dall'utente.

```
void main(void) {
    list l = emptylist();
    int el;
    int scelta;
    char invio;
    FILE *f;
    int n=35;
    int k=4;

    f = fopen("../Valori2.txt", "r");
    if (f == NULL) {
        printf("Errore Apertura file");
        return;
    }

    while (fscanf(f, "%d", &el)>0) {
        l = cons(el, l);
    }

    fclose(f);

    showlist(l);

    n=9;
    k=2;
    printf("Tolgo %d su %d elementi consecutivi\n", n,k);
    l=EliminaFisso(l, n, k);
    showlist(l);

}
```

```

list EliminaFisso(list l, int n, int k) {

    list t=l, f, s = emptylist();
    int i, val=0;

    while(!empty(t)){
        val=0;
        f=t;
        for(i=0; i < k && !empty(f); i++){
            val += head(f);
            f = tail(f);
        }
        if(i==k && val == n){
            if(empty(s))
                return f;
            else{
                s->next = f;
                return l;
            }
        }
        s=t;
        t = tail(t);
    }

    return l;
}

```

ESERCIZIO 4 (9 punti)

Dato un albero binario contenente valori interi, scrivere la funzione che esplori l'albero e mostri, per ogni foglia, il valore medio dei valori nel cammino per raggiungerla.

```
void main () {
    tree t1,t2;
    FILE *f;
    list l = NULL;
    element el;
    int ris = -1;

    f = fopen("../Valori3.txt", "r");
    if (f == NULL) {
        printf("Errore Apertura file");
        return;
    }
    t1 = emptyTree();

    while (fscanf(f, "%d", &el)>0) {
        t1 = insertBinOrd(el, t1);
    }

    fclose(f);

    f = fopen("../Valori4.txt", "r");
    if (f == NULL) {
        printf("Errore Apertura file");
        return;
    }
    t2 = emptyTree();

    while (fscanf(f, "%d", &el)>0) {
        t2 = insertBinOrd(el, t2);
    }

    fclose(f);

    printf("\nVisita in ordine t1\n");
    inOrder(t1);
    printf("\nVisita in ordine t2\n");
    inOrder(t2);

    VisualizzaLunghezzaCammino(t2,0, 0);

}
```

```

int VisualizzaLunghezzaCammino(tree t, int l, element sum) {

    if (!empty(t)) {

        VisualizzaLunghezzaCammino(left(t), l+1, sum + root(t));
        VisualizzaLunghezzaCammino(right(t), l+1, sum + root(t));

    }
    else {

        printf("\nLunghezza cammino %d, media %f", l, (float)
sum/l);
        return l;
    }
}

```