

# Corso di Laurea in Ingegneria Informatica (NOD)

## Fondamenti di Informatica II e Lab

Prof. Maurizio Vincini

Compito Scritto - 15/06/2011

### ESERCIZIO 1 (7 punti)

Dato il seguente listato:

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

void procl(int a, int b, int* c) {
    int d;
    d = a; a = b; *c = d; b = (*c) * a;
    return; }

char funz1 (int x, int y) {
    int* z ;
    z = (int*) malloc(sizeof(int));
    y = y -94;
    x = y - 3;
    procl(x, y, z);
    printf("valori x: %d, y:%d, z:%d\n", x, y, *z);
    return *z; }

char funz2 (int a1, char* a2) {
    int* a3 ;
    a3 = (int*) malloc(sizeof(int));
    a1 = (*a2) - 95;
    procl(a1, *(a2), a3);
    printf("valori a1: %d, a2:%d, a3:%d\n", a1, *a2, *a3);
    return *a3; }

void main() {
    char a = 'c', b = 'b', c;
    c = funz1(a-94,b);
    a = funz2(c, &a);
    printf("scrivi a:%d, b:%d, c:%d \n", a, b, c);
    return; }
```

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

Si ricorda che il valore del codice ASCII della lettera 'a' è 97.

### ESERCIZIO 2 (10 punti)

E' dato il file binario "azioni.dat" contenente dati relativi al mercato azionario, secondo la seguente struttura dati

```
typedef struct {
    int    Numero;
    float  CostoUnitario;
    float  Guadagno;
} Azione;
```

nella quale Numero indica il numero di azioni, CostoUnitario il costo della singola azione e Guadagno la quantità di denaro guadagnata acquistando tutte le azioni indicate da Numero.

Avendo a disposizione un valore massimo di budget inserito da utente, scrivere un programma che scelga le azioni da comprare (un pacchetto si deve comprare per intero senza frazionamenti) in modo da massimizzare il guadagno (lo studente scelga se implementare una soluzione esaustiva od euristica).

### ESERCIZIO 3 (8 punti)

Sia data una struttura

```
typedef struct nodo
{
    int val;
    struct nodo *sx;
    struct nodo *dx;
} item;
```

che rappresenta un nodo di un albero binario a valori interi.

Diciamo che un nodo è dominante se non è una foglia e contiene un valore maggiore della somma dei valori dei suoi figli.

Si scriva una procedura `int ContaDominanti(albero *t)` che, dato il puntatore `a` alla radice di un albero binario esistente, restituisca il numero di nodi dominanti contenuti nell'albero di radice `t`.

### ESERCIZIO 4 (8 punti)

Sia data una struttura

```
typedef struct elemento
{
    int val;
    struct elemento *next
} item;
```

che rappresenta un elemento di una lista di interi.

Si scriva una procedura `lista *SeekAndDestroy(lista *l, int k)` che data una lista `lis` di interi positivi e un intero positivo `k`, cerca all'interno di `lis` la prima sequenza di elementi consecutivi la cui somma sia esattamente `k` e elimina tali elementi dalla lista. La funzione deve restituire un puntatore al primo elemento della lista così modificata.

La valutazione terrà conto della semplicità ed efficienza della soluzione proposta: a tale proposito indicare la complessità asintotica in termini del numero `n` di elementi presenti nella lista.

[Nota: deve essere eliminata solo la prima sequenza di elementi con somma `k`; una tale sequenza potrebbe non esistere: in tal caso la lista non deve essere modificata.]