Fondamenti di Informatica L-A (A.A. 2003/2004) - Ingegneria Informatica
Prof.ssa Mello & Prof. Bellavista – Seconda Prova Intermedia del 09/12/2003 - durata 2h
COMPITO A

ESERCIZIO 1 (12 punti)

Si scriva una funzione **`lettura()`** che riceva come parametri di ingresso il nome di un file binario **`nome_file`**, un vettore **`prezzi`** di strutture **`item`** (definite nel seguito), la dimensione fisica **`max`** del vettore `prezzi`, e un puntatore a intero **`num`** che rappresenta la dimensione logica del vettore. La funzione deve aprire il file binario `nome_file` (in caso di errore nell'apertura di tale file la funzione deve restituire immediatamente il valore –1); la funzione deve poi leggere dal file binario al più `max` strutture di tipo `item`, che devono essere memorizzate nel vettore `prezzi`; la dimensione logica del vettore deve essere restituita tramite il parametro `num` (passato appunto per riferimento). Al termine della lettura dei dati la funzione deve restituire il valore 0. **(6 punti)**.

```
int lettura (char nome_file[], item prezzi[], int max, int * num);
```

È dato un file binario **`dati.dat`**, contenente strutture di tipo **`item`**, composte da una stringa **`nome`** (di al più 20 caratteri) e due campi **`old_price`** e **`new_price`**, entrambi di tipo `float`. Il file contiene al più 100 strutture di tipo `item` (potrebbero essere anche in numero minore).
Si scriva un programma C che inserisca in un vettore **`V`** (supposto di dimensione massima **`DIM 100`**) tutti i dati presenti nel file `dati.dat`; il programma quindi allochi dinamicamente un secondo vettore **`infl`** di `float`, la cui dimensione fisica deve essere pari ala dimensione logica di `V`. Il programma assegni ad ogni elemento di `infl` il valore dell'inflazione calcolata sul corrispondente elemento di `V`, e ne stampi anche il valore a video. Si ricorda che l'inflazione dell'elemento i-esimo è così calcolata: **(6 punti)**

$$\text{infl}[i] = \left(\frac{new\_price_i}{old\_price_i} - 1 \right) * 100$$

ESERCIZIO 2 (9 punti)

Si scriva una funzione ricorsiva **`select()`** che data in ingresso una lista di interi (non ordinati), restituisca in uscita una lista contenente solo gli elementi pari della lista di origine, ordinati in maniera crescente. A tal scopo si supponga di possedere il tipo di dato astratto **`list`** e le sole operazioni PRIMITIVE (che quindi possono NON essere riportate nella soluzione): si consideri quindi attentamente il fatto che funzioni più elaborate come `member()` o `insord()` invece NON sono disponibili.
Si scriva poi una seconda funzione **`max()`** che, data in ingresso una lista di interi, determini il valore massimo contenuto nella lista, utilizzando la notazione a puntatori. La funzione può essere realizzata in maniera sia iterativa che ricorsiva.

ESERCIZIO 3 (8 punti)

Nel caso in cui il programma sorgente C seguente compili ed esegua correttamente, se ne indichino i valori stampati a tempo di esecuzione, motivando la risposta data. In caso di errori di compilazione o errori runtime, si descriva invece nel dettaglio la motivazione di tali errori.
Si indichino inoltre quali sono i blocchi in cui sono visibili le variabili **N** e **L**, sempre motivando la risposta.

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

int L = 4;

void F(int v[], int pos) {
    int N=L;
    N--;
    *(v+pos+1) = *(v+pos+1) + *(v+pos);
    return;
}

main () {
    int N=L;
    int * v;
    int  i;
    {
      ++N;
    }
    v = (int *) malloc((N) * sizeof(int));
    for (i=0; i<N; i++)
      v[i]=2*i+1;

    for (i=0; i<L; i++)
      if (v[i]) F(v, i);

    printf("Adesso N vale: %d\n", N);

    for (i=1; i++ < (i?N:0); )
      printf("%d\n", v[i-1]);
}
```

Esercizio 4 (3 punti)

Dire se il seguente codice viene compilato correttamente. In caso positivo spiegare anche che cosa fa e se è semanticamente corretto; in caso contrario spiegare perché il codice non viene compilato.

```
      int V[5] = {1,2,3,4,5};
      printf("%d\n", *(V+6));
```

Soluzione Compito A

Esercizio 1

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

#define DIM 21
#define MAX_ITEM 100

typedef struct {
    char nome[DIM];
    float old_price;
    float new_price;
} item;

int lettura (char nome_file[], item prezzi[], int max, int * num) {
    FILE * f;
    int i;

    f = fopen(nome_file, "rb");
    if (f == NULL)
      return(-1);

    *num = fread(prezzi, sizeof(item), max, f);
    fclose(f);
    return 0;
}

int main() {
    item V[MAX_ITEM];
    float * infl;
    int num, i, result;

    result = lettura("dati.dat", V, MAX_ITEM, &num);

    if (result!=0) {
      printf("Problemi durante la lettura...\n");
      exit(-1);
    }

    infl = (float *) malloc(sizeof(float) * num);

    for (i=0; i < num; i++) {
      infl[i] = (V[i].new_price/V[i].old_price -1)*100;
      printf("%6.2f\n", infl[i]);
    }
    return 0;
}
```

Esercizio 2

```
list  select(list l) {
      if empty(l)
            return emptylist();
      else
            if (!(head(l)%2))
                  return insord(head(l), select(tail(l)));
            else
                  return select(tail(l));
}

list insord(element el, list l) {
      if (empty(l)) return cons(el,l);
      else if (el <= head(l)) return cons(el,l);
      else return cons(head(l), insord(el,tail(l)));
}

int max(list l) {
      int temp;
      if (l==NULL) abort();   // fallimento in caso di lista vuota
      else temp = l->value;
      while (l!=NULL) {
            if ((l->value)>temp) temp = l->value;
            l = l->next;
            }
      return temp;
}
```

Esercizio 3

Il programma stampa:

Adesso N vale: 5
4
9
16
25

Innanzitutto N viene pre-incrementato all'interno del blocco, e quindi vale 5. Viene poi allocato dinamicamente un vettore di interi di dimensione N, cioè 5, e viene riempito coi primi 5 numeri dispari. Per ognuno di questi elementi è invocata la funzione F, che sovrascrive il valore in posizione pos+1. Il programma principale quindi stampa a video il valore di N, e poi stampa 4 elementi del vettore v, da quello di posizione 1 fino all'ultimo.

La variabile N è dichiarata sia nel blocco main, sia nella funzione `F()`. N non è visibile globalmente, ma solo nei singoli blocchi. L invece è dichiarata globalmente, e quindi visibile in ogni blocco.

Esercizio 4

Il codice viene compilato correttamente, ma non è corretto. Il vettore V infatti è di dimensione 5, mentre la seconda istruzione stampa l'elemento di indice 6. Il codice presentato stamperebbe quindi a video una zona di memoria non allocata, il cui contenuto non è noto a priori.