

Fondamenti di Informatica L-A (A.A. 2004/2005) - Ingegneria Informatica
Prof.ssa Mello & Prof. Bellavista – I Prova Intermedia del 11/11/2004 - durata 2h - COMPITO B

▪ **Esercizio 1 (punti 12)**

Una associazione di consumatori vuole svolgere una indagine di mercato sulle offerte per computer portatili. A tal scopo vuole realizzare un programma che chieda, per un numero finito di modelli, il nome del modello ed il suo prezzo in euro, e restituisca come output il nome del modello più economico e il suo prezzo vantaggioso.

A tal scopo si definisca inizialmente una struttura di nome *offerta*, composta dal nome del modello (al più 127 caratteri) e dal prezzo (rappresentato tramite *float*). Si realizzi poi:

- a) una procedura
`void seleziona(offerta lista[], int length, char affare[], float *prezzo)`

che riceve in ingresso un array *lista* di tipo *offerta*, e la lunghezza *length* di tale array. La funzione deve determinare il modello più economico e copiarne il nome nella stringa *affare*; il costo in euro deve poi essere restituito tramite la variabile *prezzo* (passata per riferimento). Al fine di copiare il nome del modello più economico, si utilizzi la funzione di libreria `strcpy(char *s, char *ct)`, che copia la stringa *ct* nella stringa *s*, terminatore compreso. (punti 6)

- b) un programma `main()` che chieda all'utente di inserire, alternativamente, il nome di un modello e il suo prezzo, per un totale di *DIM* offerte (si controlli che il prezzo sia maggiore o uguale a 0; in caso contrario si ripeta la richiesta di inserire il prezzo). Il programma deve provvedere a memorizzare tali informazioni in un array di tipo *offerta* precedentemente definito, di dimensione fissa *DIM* (si supponga per semplicità che *DIM* valga 10). Una volta raccolte le informazioni, si utilizzi la procedura `seleziona()` definita al punto precedente per ottenere il modello e il prezzo del prodotto più vantaggioso, e si stampino a video tali informazioni. (punti 6)

▪ **Esercizio 2 (punti 7)**

Dato il seguente programma C:

```
#include <stdio.h>

int swap(int * nome, int length);

int main() {
    int temp[] = {1, 2, 3, 4, 5, 6};
    int i= 0;
    int result;

    result = swap(temp, 4);

    for (i=0; i<result; i++)
        printf("%d\n", temp[i]);

    return 0;
}

int swap(int * nome, int length) {
    int i = 0;
    int temp;

    while (i<length-1) {
        temp = *nome;
        *nome = *(nome+1);
        *(nome+1) = temp;
        i = i + 2;
        nome = nome + 2;
    }
    return length;
}
```

Che cosa fa la funzione `swap()`? Che cosa stampa il programma? Si motivi opportunamente la risposta.

▪ **Esercizio 3 (punti 5)**

Si consideri la seguente funzione:

```
int F(int a, int b) {
    if (a > b )
        return a/b;
    else {
        b-=a;
        a++;
        return F(a, b);
    }
}
```

Si scriva il risultato della funzione quando invocata come **F(1, 8)** e si mostrino i record di attivazione. La funzione è tail-ricorsiva?

▪ **Esercizio 4 (punti 6)**

Si scriva una funzione ricorsiva **int prod(int x)** che, ricevuto come parametro un numero intero positivo (pari o dispari), calcoli ricorsivamente il prodotto di tutti i numeri dispari compresi tra 1 e quel numero (compreso).

▪ **Esercizio 5 (punti 2)**

Si consideri la grammatica **G** con scopo **S** e simboli terminali {**qui, quo, qua, zio, paperone, paperino, pippo**}

```
S ::= C D C | D A | A
D ::= C A B | A
C ::= A B | A B B | A A
A ::= qui | qua | pippo | zio
B ::= quo | paperone | paperino
```

Si dica se la stringa "**qui quo qua zio paperone**" è sintatticamente corretta rispetto a tale grammatica e se ne mostri la derivazione *left most*.

Soluzione Compito B

Esercizio 1

```
#include <string.h>

#define MAX 128
#define DIM 2

typedef struct {
    char modello[MAX];
    float prezzo;
} offerta;

void seleziona(offerta lista[], int length, char affare[], float * prezzo) {
    int i;
    int pos = 0;

    if (length <= 0) {
        strcpy(affare, "NESSUN AFFARE!");
        *prezzo = 0;
        return;
    }
    for (i=1; i<length; i++)
        if (lista[i].prezzo < lista[pos].prezzo) {
            pos = i;
        }

    strcpy(affare, lista[pos].modello);
    *prezzo = lista[pos].prezzo;
}

int main() {
    int i;
    offerta lista[DIM];
    char affare[MAX];
    float prezzo = 0;

    for (i=0; i<DIM; i++) {
        printf("Inserire nome modello: ");
        scanf("%s", lista[i].modello);

        do {
            printf("Inserire prezzo: ");
            scanf("%f%c", &lista[i].prezzo);
        } while (lista[i].prezzo < 0);
    }

    seleziona(lista, i, affare, &prezzo);
    printf("Il modello %s e' il piu' conveniente, al costo di %6.2f euro.\n", affare,
prezzo);

    return 0;
}
```

Esercizio 2

Il programma stampa a video la sequenza di interi, ciascuno su una linea diversa:

```
2    1    4    3
```

All'inizio del `main` sono dichiarate tre variabili, `temp` come array di interi, `result` e `i` come interi. Subito dopo questa fase a `result` è assegnato il valore di ritorno della funzione `swap`. Viene eseguito un ciclo con `i` come indice che va da 0 fino al valore indicato da `swap` (escluso); questo ciclo si limita a stampare a video i primi interi contenuti in `temp`, e poi termina. Dunque, al fine di

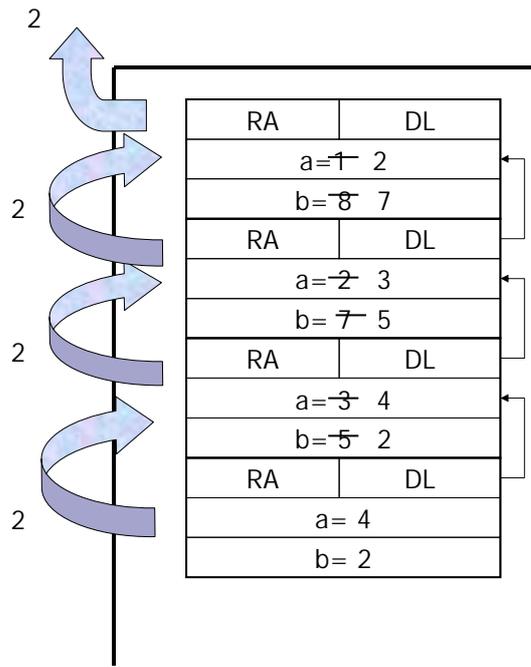
comprendere che cosa stampi il programma, è necessario comprendere quali operazioni esegue la funzione `swap()`, e quale valore ritorna.

La funzione `swap()` riceve in ingresso un puntatore ad intero chiamato `nome`, ed un intero `length`. Dopo aver dichiarato le variabili `i` e `temp` (ed inizializzato `i` al valore 0) la funzione `swap` esegue un ciclo `while`; in tale ciclo viene effettuato lo scambio (swap) dei valori interi puntati rispettivamente da `nome` e da `(nome+1)`, utilizzando come variabile di appoggio `temp`. All'interno del ciclo poi si provvede anche ad incrementare di 2 sia l'indice `i` che il puntatore ad intero `nome`. Poiché `swap` viene invocata con `length` pari a 4, il ciclo viene eseguito 2 volte, e sortisce l'effetto di scambiare il primo elemento dell'array con il secondo alla prima iterazione, ed il terzo elemento dell'array con il quarto alla seconda iterazione. La funzione restituisce poi il valore `length`.

Quindi, dopo l'invocazione della funzione `swap()`, alcuni elementi dell'array (i primi 4) sono stati scambiati (il primo con il secondo, il terzo con il quarto). La funzione `main()` si limita a stampare i "result" elementi, dove `result` viene a valere 4. Quindi stampa, su quattro righe distinte, i valori 2, 1, 4, 3.

Esercizio 3

Il risultato è 2.



La funzione è tail-ricorsiva.

Esercizio 4

```
int prod(int x) {
    if (x == 1)
        return 1;
    else {
        if (x%2 == 1)
            return x * prod(x-2);
        else
            return prod(x-1);
    }
}
```

Esercizio 5

La frase è sintatticamente corretta. Derivazione left-most:

S -> C D C -> A B D C -> qui B D C -> qui quo D C -> qui quo A C -> qui
quo qua C -> qui quo qua A B -> qui quo qua zio B -> qui quo qua zio paperone