Fondamenti di Informatica L-A (A.A. 2005/2006) - CdS Ingegneria Informatica
Prof.ssa Mello & Prof. Bellavista – I Prova Intermedia del 02/11/2005 - durata 2h30m
COMPITO A

ESERCIZIO 1 (12 punti)

Un centro di meteorologia vuole effettuare in modo automatizzato statistiche sulla temperatura registrata quotidianamente nell'arco di un mese. In particolare, vuole ottenere la temperatura media (non considerando il giorno più caldo del mese) e la data del giorno più caldo con la relativa temperatura. A tale scopo si realizzi:

1) una funzione

```
float mediaTemp(int temp[], int length, int* piuCaldoTemp, int* piuCaldoGiorno)
```

che noto il numero di giorni presi in esame **`length`** e le temperature **`temp`**, restituisca la temperatura media (escludendo dal calcolo il giorno più caldo) come **`float`** e tramite i parametri **`piuCaldoTemp`** e **`piuCaldoGiorno`** rispettivamente la temperatura e la data del giorno più caldo. Si assuma che il giorno 0 corrisponda al primo giorno del mese, il giorno 1 al secondo, e così via; in relazione al parametro **`temp`** si noti che la temperatura presente in **`temp[0]`** corrisponde al primo giorno del mese, la temperatura in **`temp[1]`** al secondo, e così via; **(7 punti)**

2) un programma **`main()`** che

a) chieda all'utente il numero di giorni **`G`** del mese preso in esame e che controlli che **`G`** abbia valore tra 28 e 31 compresi. In caso contrario, si deve richiedere nuovamente il numero di giorni;

b) chieda all'utente di inserire **`G`** temperature in ordine dal primo all'ultimo giorno del mese, controllando che le temperature inserite abbiano valori tra -30 e +50 compresi. In caso contrario, si deve richiedere nuovamente la temperatura;

c) richiami opportunamente la funzione **`mediaTemp(...)`** e infine

d) stampi la temperatura media restituita da **`mediaTemp(...)`**,i gradi e la data del giorno più caldo. **(5 punti)**

ESERCIZIO 2 (6 punti)

Il seguente programma C compila correttamente? In caso affermativo, quali sono i valori stampati a tempo di esecuzione? (Si motivi opportunamente la risposta data)

```
#include <stdio.h>

#define DIM 4

void fun(float *x1, float* x2, int dim){
      int i;
      float temp;
      for(i=0;i<dim;i++){
            if (*(x1+i)>=*(x2+dim-1-i)) {
                  temp=x1[i];
                  x1[i]=x2[dim-1-i];
                  x2[dim-1-i]=temp;
            }
      } return;
}

int main(){
      int i=0;
      float f1[DIM]={0.2,3.0,8.0,-17.0};
      float f2[DIM]={0.3,2.1,4.0,15};

      fun(f1,f2,DIM);
      for(;i<DIM;i++) printf("%f ",f1[i]);
      printf("\n");
```

```
      for(;i>0;i--) printf("%f ",f2[i-1]);
      printf("\n");
      return 0;
}
```

ESERCIZIO 3 (6 punti)

Si scriva una funzione iterativa **`int fun(char *str1, char *str2)`** che, ricevuti come parametri in ingresso due stringhe ben formate **`str1`** e **`str2`**, restituisca come valore di ritorno un **`int`** rappresentante la somma totale di occorrenze di ogni carattere di **`str1`** in **`str2`**. Ad esempio, la chiamata **`fun("Pippo", "Poporono")`** dovrà restituire 7 (1 occorrenza di 'P', 1 di 'p' per 2 volte, 4 di 'o').
Si proponga una possibile funzione chiamante.

ESERCIZIO 4 (4 punti)

Data la funzione:

```
int fun(int a, int b, int* c){
      if(a>=b){
            a=a-b;
            return 1+fun(a,b,c);
      }
      else {
            *c=a;
            return 0;
      }
}
```

e la funzione chiamante:

```
int main(){
      int r=0;
      fun(13,5,&r);
      return 0;
}
```

mostrare la sequenza dei record di attivazione.

ESERCIZIO 5 (2 punti)

Si consideri la grammatica G con scopo S e simboli terminali {1,2,3,4,a,b,c}

```
S::= A | BC
A::= DC | DCB
B::= EB | DE
C::= DCE | E
D::= 1 | 2 | 3 | 4
E::= a | b| c
```

La stringa “ba4c1ab” appartiene alla grammatica? In caso affermativo, se ne mostri la derivazione left-most.

ESERCIZIO 6 (2 punti)

Date le definizioni: **`int x1[]={1,7,0,-3}, * x2;`**
indicare la quantità di memoria allocata dalle due variabili sullo stack. Inoltre, spiegare se e perché il seguente assegnamento risulta corretto/scorretto:

```
x1 = x2;
```

Soluzioni

Esercizio 1

```
#define DIM 31
#include <stdio.h>

float mediaTemp(int temp[], int length, int* piuCaldoTemp, int* piuCaldoGiorno){
	int i; int somma=0;
	*piuCaldoGiorno=0;
	*piuCaldoTemp=temp[0];

	for(i=1;i<length;i++){
		if(temp[i]>*piuCaldoTemp){
			somma=somma+*piuCaldoTemp;
			*piuCaldoTemp=temp[i];
			*piuCaldoGiorno=i;
		}
		else somma=somma+temp[i];
	}
	return somma/((float)(length-1));
}

int main(){
	int temp[DIM];
	int piuCaldoTemp, piuCaldoGiorno, i, giorni;
	float media;

	do {	printf("Numeri giorni?\n");
		scanf("%d",&giorni);
	} while((giorni<28)||(giorni>31));

	for(i=0;i<giorni;i++){
		do {	printf("Temp giorno %d?\n",i);
			scanf("%d",&temp[i]);
		} while((temp[i]<-30)||(temp[i]>50));
	}

	media=mediaTemp(temp, giorni, &piuCaldoTemp, &piuCaldoGiorno);
	printf("media %f; più caldo %d il %d\n",media, piuCaldoTemp, piuCaldoGiorno);
	return 0;
}
```

Esercizio 2

0.2 3.0 2.1 -17.0
15.0 4.0 8.0 0.3

La funzione fun() scorre gli array x1 e x2; il primo dall'inizio alla fine, il secondo dalla fine all'inizio. Mentre scorre i due array, fun() confronta i valori di x1 e x2; nel caso in cui il valore di x1 sia maggiore o uguale a quello di x2, i due valori vengono invertiti.
La funzione main() stampa l'array f1 modificato dall'inizio alla fine e l'array f2 modificato dalla fine all'inizio.

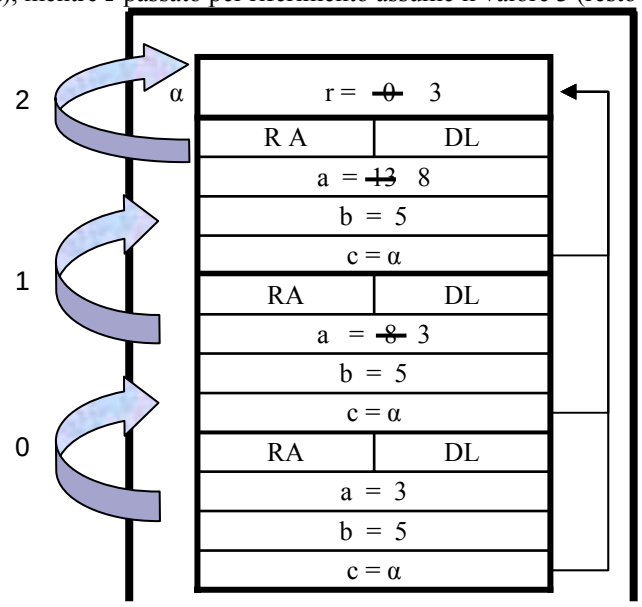
Esercizio 3

```
int fun(char* str1, char* str2){
	int totale=0, i=0, j;

	while(str1[i]!='\0'){
		j=0;
		while(str2[j]!='\0'){
			if(str1[i]==str2[j]) totale++;
			j++;
		}
		i++;
	}
```

```
        return totale;
}

int main(){
        int occorrenze=0;
        char str1[]="abcdefghilmno";
        char str2[]="aabbbde";
        occorrenze=fun(str1,str2);
        printf("Occorrenze %d\n",occorrenze);
        return 0;
}
```

Esercizio 4

La funzione effettua una divisione fra il primo e il secondo parametro di ingresso; come valore di ritorno restituisce 2 (risultato della divisione intera), mentre **r** passato per riferimento assume il valore 3 (resto della divisione intera).



Esercizio 5

S -> BC -> EBC -> bBC -> bEBC -> baBC -> baDEC -> ba4EC -> ba4cC -> ba4cDCE -> ba4c1CE -> ba4c1EE -> ba4c1aE -> ba4c1ab

Esercizio 6

La variabile `x1` alloca lo spazio necessario a contenere 4 `int`.
La variabile `x2` alloca lo spazio necessario a contenere l'indirizzo di un `int`.
L'assegnamento `x1 = x2` è scorretto: anche se il tipo di dato associato a `x1` è uguale al tipo di dato associato a `x2` (per come gli array sono implementati in C), non è consentito cercare di variare il valore di `x1` (l'indirizzo denotato dal nome di un array è costante e non modificabile).