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 G

ESERCIZIO 1 (12 punti)
Un comune vuole calcolare la velocità media che percorrono i veicoli in una strada nell'arco di una giornata. In particolare, vuole ottenere la velocità media di tutte le autovetture la cui targa, descritta con un numero a sei cifre, è inferiore o uguale a 900000 e la velocità e la targa del mezzo più lento, includendo anche gli autoveicoli la cui targa è superiore a 900000. A tale scopo si realizzi:

1) una funzione

```
float mediaVel(int vel[], long targhe[], int length, int* menoVeloce, long* menoVeloceTarga)
```

che noto il numero di autoveicoli **`length`**, le velocità **`vel[]`** e le targhe **`targhe[]`**, restituisca la velocità media come **`float`** (escludendo i veicoli la cui targa è superiore a 900000). Inoltre, tramite i parametri **`menoVeloce`** e **`menoVeloceTarga`** la funzione deve restituire rispettivamente la velocità e la targa dell'autoveicolo meno veloce, includendo anche i veicoli la cui targa è superiore a 900000. Si assuma che la velocità in **`vel[0]`** corrisponda al veicolo con targa **`targhe[0]`**, la velocità in **`vel[1]`** al veicolo con targa **`targhe[1]`** e così via; **(7 punti)**

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

a) chieda all'utente il numero di veicoli **`V`** presi in esame e controlli che **`V`** abbia valore tra 10 e 100 compresi. In caso contrario, si richieda nuovamente il numero di veicoli;

b) chieda all'utente di inserire **`V`** velocità e targhe, controllando che le velocità inserite abbiano valori tra 0 e 120 compresi e le targhe valori tra 0 e 1000000 (estremi non inclusi). In caso contrario, si richieda nuovamente all'utente di inserire velocità e/o targa,

c) richiami opportunamente la funzione **`mediaVel(...)`**;

d) stampi la velocità media restituita da **`mediaVel(...)`**, la velocità e la targa dell'autoveicolo meno veloce. **(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+dim-1-i)<=*(x2+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 **`str2`** in **`str1`**. Ad esempio, la chiamata **`fun("Casale", "Sala")`** deve restituire 5 (2 occorrenze di 'a' per 2 volte, 1 occorrenza di 'l').
Si proponga una possibile funzione chiamante.

ESERCIZIO 4 (4 punti)

Data la funzione:

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

e la funzione chiamante:

```
int main(){
	int r=0;
	fun(6,&r,17);
	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,x,y,z}

```
S::= AA | EBC
A::= CF | ECB
B::= FBE | E
C::= DC | DF
D::= 1 | 2 | 3 | 4
E::= a | b | c
F::= x | y | z
```

La stringa “1yzb34xzaa” appartiene alla grammatica? Se sì se ne mostri la derivazione left-most.

ESERCIZIO 6 (2 punti)

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

```
x2 = x1;
```

Soluzioni

Esercizio 1

```
#define DIM 100
float mediaVel(int vel[], long targhe[], int length, int* menoVeloce, long*
menoVeloceTarga){
	int i, somma=0, macchineInSomma=0;
	*menoVeloce=130;
	*menoVeloceTarga=-1;

	for(i=0; i<length; i++) {
		if (targhe[i]<=900000) {
			somma=somma+vel[i];
			macchineInSomma++;
		}
		If (vel[i]<*menoVeloce) {
			*menoVeloce=vel[i];
			*menoVeloceTarga=targhe[i];
		}
	}
	return ((float)somma)/((float)macchineInSomma);
}
int main(){
	int vel[DIM];
	long targhe[DIM];
	int menoVeloce, V, i;
	long menoVeloceTarga;
	float media;

	do {  printf("Numero veicoli?\n");
		scanf("%d",&V);
	} while(V<10||V>100);

	for(i=0;i<V;i++){
		do {  printf("Velocita' veicolo %d?\n",i);
			scanf("%d",&vel[i]);
		} while((vel[i]<0)||(vel[i]>120));
		do {  printf("Targa veicolo %d?\n",i);
			scanf("%l",&targhe[i]);
		} while((targhe[i]<=0)||(targhe[i]>=1000000));
	}
	media=mediaVel(vel, targhe, V, &menoVeloce, &menoVeloceTarga);
	printf("media %f; meno veloce %d con targa %l\n", media, menoVeloce,
menoVeloceTarga);
	return 0;
}
```

Esercizio 2

15.0 3.0 2.1 -17.0
0.2 4.0 8.0 0.3

La funzione fun() scorre gli array x1 e x2; il primo dalla fine all'inizio, il secondo dall'inizio alla fine. Mentre scorre i due array, fun() confronta i valori di x1 ed x2; se il valore di x1 è minore o uguale di quello di x2, inverte i valori di x1 in posizione i e di x2 in posizione DIM-1-i. Si noti che i valori scambiati non coincidono con quelli utilizzati nel confronto.
La funzione main() stampa l'array f1 modificato dall'inizio alla fine e l'array f2 modificato dalla fine all'inizio.

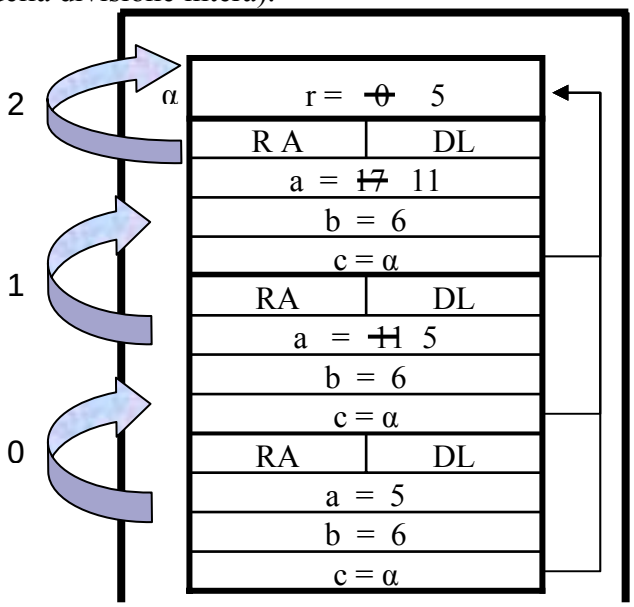
Esercizio 3

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

	while(str2[i]!='\0'){
```

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

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

Esercizio 4

La funzione effettua una divisione; come risultato restituisce 2 (risultato della divisione intera), **r** assume il valore 5 (resto della divisione intera).



Esercizio 5

S -> AA -> CFA -> DFFA -> 1FFA -> 1yFA -> 1yzA -> 1yzECB -> 1yzbCB -> 1yzbDCB -> 1yzb3CB -> 1yzb3DFB -> 1yzb34FB -> 1yzb34xB -> 1yzb34xFBE -> 1yzb34xzBE -> 1yzb34xzEE -> 1yzb34xzaE -> 1yzb34xzaa

Esercizio 6

La variabile `x1` alloca lo spazio necessario a contenere 5 `int`.
La variabile `x2` alloca lo spazio necessario a contenere l'indirizzo di un `int`.
L'assegnamento `x2 = x1` è corretto: il nuovo valore di `x2` è l'indirizzo del primo elemento di `x1`; dopo l'assegnamento, le due variabili puntano allo stesso indirizzo di memoria.