

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 D

ESERCIZIO 1 (12 punti)

Un comune vuole calcolare l'altezza media dei suoi residenti. In particolare, vuole ottenere l'età media di tutti i residenti la cui altezza è compresa tra 1.2 e 2.1 metri, e l'altezza e l'età della persona più bassa senza vincoli d'altezza. A tale scopo si realizzi:

1) una funzione

```
float mediaEta(float alt[], int eta[], int length, float* piuBasso,
int* piuBassoEta)
```

che noto il numero di persone **length**, le altezze **alt[]** e le età **eta[]**, restituisca l'età media come **float** (escludendo le persone la cui altezza non è compresa tra 1.2 e 2.1 metri). Inoltre, tramite i parametri **piuBasso** e **piuBassoEta** la funzione deve restituire rispettivamente l'altezza e l'età della persona più bassa senza vincoli d'altezza. Si assuma che l'altezza in **alt[0]** corrisponda alla persona con età **eta[0]**, l'altezza in **alt[1]** alla persona con età **eta[1]** e così via; **(7 punti)**

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

- a) chieda all'utente il numero di persone **P** prese in esame e che controlli che **P** abbia valore tra 10 e 100 compresi. In caso contrario, si richieda nuovamente il numero di persone;
- b) chieda all'utente di inserire **P** altezze ed età, controllando che le altezze abbiano valori tra 0.3 e 2.3 compresi e le età valori tra 0 e 120 compresi. In caso contrario, si richieda nuovamente all'utente di inserire l'altezza e/o l'età,
- c) richiami opportunamente la funzione **mediaEta(...)**;
- d) stampi l'età media restituita da **mediaEta(...)**, l'altezza e l'età della persona più bassa.

(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(char *x1, char* x2, int dim){
    int i; char temp;
    for(i=dim-1;i>=0;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;
    char c1[DIM]={'a','b','c','d'};
    char c2[DIM]={'b','d','a','c'};

    fun(c1,c2,DIM);

    for(;i<DIM;i++) printf("%c ",c1[i]);
    printf("\n");
    for(;i>0;i--) printf("%c ",c2[DIM-i]);
    printf("\n");
    return 0;
}
```

ESERCIZIO 3 (6 punti)

Si scriva una funzione iterativa `char fun(char *a, char *b)` che, ricevuti come parametri in ingresso due stringhe ben formate **a** e **b**, restituisca come valore il carattere di **b** che compare più volte in **a** (si assuma che in **b** un carattere possa comparire al più una volta). Ad esempio, la chiamata `fun("zuzzurellone", "lazo")` deve restituire 'l'.

Si proponga una possibile funzione chiamante.

ESERCIZIO 4 (4 punti)

Data la funzione:

```
int fun(int x, int* y, int z){
    *y=*y+z%x;
    if(z<x) return 1;
    else { z=z/x;
          return x*fun(x,y,z);
        }
}
```

e la funzione chiamante:

```
int main(){
    int r=0;
    fun(6,&r,59);
    return 0;
}
```

mostrare la sequenza dei record di attivazione.

ESERCIZIO 5 (2 punti)

Un elaboratore rappresenta i numeri interi su 8 bit dei quali 7 sono dedicati alla rappresentazione del modulo del numero e uno al suo segno. Indicare come viene svolta la seguente operazione aritmetica e determinarne il risultato trasladolo poi in decimale per la verifica:

29-75

ESERCIZIO 6 (2 punti)

Si illustrino, eventualmente tramite esempi, le differenze nel tempo di vita delle variabili quando definite come globali o come locali a una funzione nel linguaggio C.

Soluzioni

Esercizio 1

```
#define DIM 100
float mediaEta(float alt[], int eta[], int length, float* piuBasso, int*
piuBassoEta){
    int i, somma=0, etaInSomma=0;
    *piuBasso=3.0;
    *piuBassoEta=-1;

    for(i=0;i<length;i++){
        if(alt[i]>=1.2&&alt[i]<=2.1){
            somma=somma+eta[i];
            etaInSomma++;
        }
        if(alt[i]<*piuBasso){
            *piuBasso=alt[i];
            *piuBassoEta=eta[i];
        }
    }
    return ((float)somma)/((float)etaInSomma);
}

int main(){
    int eta[DIM];
    float alt[DIM];
    float piuBasso, media;
    int piuBassoEta, P, i;

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

    for(i=0;i<P;i++){
        do { printf("Altezza persona %d?\n",i);
            scanf("%f",&alt[i]);
        } while(alt[i]<0.3 || alt[i]>2.3);
        do { printf("Eta' persona %d?\n",i);
            scanf("%d",&eta[i]);
        } while(eta[i]<0 || eta[i]>120);
    }
    media=mediaEta(alt, eta, P, &piuBasso, &piuBassoEta);
    printf("media %f; più basso %f di età %d", media, piuBasso, piuBassoEta);
    return 0;
}
```

Esercizio 2

c a d d

b c b a

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 ed x2; se il valore di x1 è maggiore o uguale a 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 gli array c1 e c2 modificati dall'inizio alla fine.

Esercizio 3

```
char fun(char* a, char* b){
    int occCharCorrente, occCharC=0, i=0, j;
    char charC='';

    while(b[i]!='\0'){
        j=0;
        occCharCorrente=0;
```

```

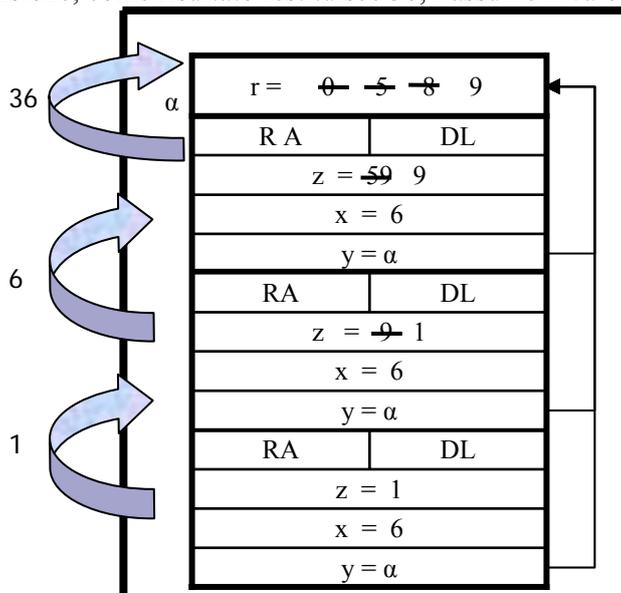
while(a[j]!='\0'){
    if(b[i]==a[j]){
        occCharCorrente++;
    }
    j++;
}
if(occCharCorrente>occCharC){
    occCharC=occCharCorrente;
    charC=b[i];
}
i++;
}
return charC;
}

int main(){
    char piuFrequente;
    char b[]="abcdefghilmno";
    char a[]="aabbbddeh";
    piuFrequente=fun(a,b);
    printf("piuFrequente %c\n",piuFrequente);
    return 0;
}

```

Esercizio 4

La funzione effettua una divisione; come risultato restituisce 36, r assume il valore 9.



Esercizio 5

Sottrazione dei moduli (numero maggiore meno numero minore)

```

1001011-    75-
0011101=    29=
-----
0101110    46

```

Aggiunta del segno

```
10101110
```

Esercizio 6

Ogni variabile viene creata quando se ne incontra la definizione. Una volta creata:

- una variabile globale viene mantenuta in memoria fino alla terminazione del programma in esecuzione
- una variabile locale ad una funzione viene eliminata appena termina la funzione che l'ha creata (e viene deallocato il corrispondente record di attivazione)