

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 H

ESERCIZIO 1 (12 punti)

Un comune vuole calcolare l'altezza media dei suoi residenti. In particolare, vuole ottenere l'altezza media di tutti i residenti la cui età è compresa tra 18 e 65 anni, ma anche l'altezza e l'età della persona più alta senza vincoli d'età. A tale scopo si realizzi:

1) una funzione

```
float mediaAlt(float alt[], int eta[], int length, float* piuAlto,
int* piuAltoEta)
```

che noto il numero di persone **length**, le altezze **alt[]** e le età **eta[]**, restituisca l'altezza media come **float** (escludendo le persone la cui età non è compresa tra 18 e 65). Inoltre, tramite i parametri **piuAlto** e **piuAltoEta** la funzione deve restituire rispettivamente l'altezza e l'età della persona più alta senza vincoli d'età. 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 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 inserite abbiano valori tra 0.3 e 2.3 compresi e le età valori tra 0 e 120, sempre estremi inclusi. In caso contrario, si richieda nuovamente all'utente di inserire l'altezza e/o l'età,

c) richiami opportunamente la funzione **mediaAlt(...)**;

d) stampi l'altezza media restituita da **mediaAlt(...)**, l'altezza della persona più alta e l'età corrispondente. **(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+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;
    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[i-1]);
    printf("\n");
    return 0;
}
```

ESERCIZIO 3 (6 punti)

Si scriva una funzione iterativa `char fun(char *str1, char *str2)` che, ricevuti come parametri in ingresso due stringhe ben formate `str1` e `str2`, restituisca come valore il carattere di `str1` che compare più volte in `str2` (si assuma che in `str1` un carattere possa comparire al più una volta). Ad esempio, la chiamata `fun("cane", "casale")` deve restituire 'a'.

Si proponga anche il codice di una possibile funzione chiamante.

ESERCIZIO 4 (4 punti)

Data la funzione:

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

e la funzione chiamante:

```
int main(){
    int r=0;
    fun(&r,5,42);
    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 traslandolo poi in decimale per la verifica:

25-71

ESERCIZIO 6 (2 punti)

Date le definizioni: `char s1[]="pippo", *s2;`

le seguenti istruzioni sono corrette? In caso affermativo, si dica che cosa viene stampato a video, motivando opportunamente la risposta.

```
s2 = s1; *(s2+1) = 'z'; printf("%s\n", s1);
```

Soluzioni

Esercizio 1

```
#include <stdio.h>
#define DIM 100

float mediaAlt(float alt[], int eta[], int length, float* piuAlto, int*
piuAltoEta){
    int i;
    float somma=0, altInSomma=0;
    *piuAlto=0.1;
    *piuAltoEta=-1;

    for(i=0;i<length;i++){
        if((eta[i]>=18) && (eta[i]<=65)){
            somma=somma+alt[i];
            altInSomma++;
        }
        if(alt[i]>*piuAlto){
            *piuAlto=alt[i];
            *piuAltoEta=eta[i];
        }
    }
    return somma/((float)altInSomma);
}

int main(){
    int eta[DIM];
    float alt[DIM];
    float piuAlto, media;
    int piuAltoEta, 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=mediaAlt(alt, eta, P, &piuAlto, &piuAltoEta);
    printf("media %f; più alto %f di età %d\n", media, piuAlto, piuAltoEta);
    return 0;
}
```

Esercizio 2

a a c b

c b d d

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 maggiore o uguale a quello di x2, inverte i valori di x1 ed x2.

La funzione main() stampa l'array c1 modificato dall'inizio alla fine e l'array c2 modificato dalla fine all'inizio.

Esercizio 3

```
char fun(char* str1, char* str2){
    int occCharCorrente, occCharC=0, i=0, j;
    char charC='';
```

```

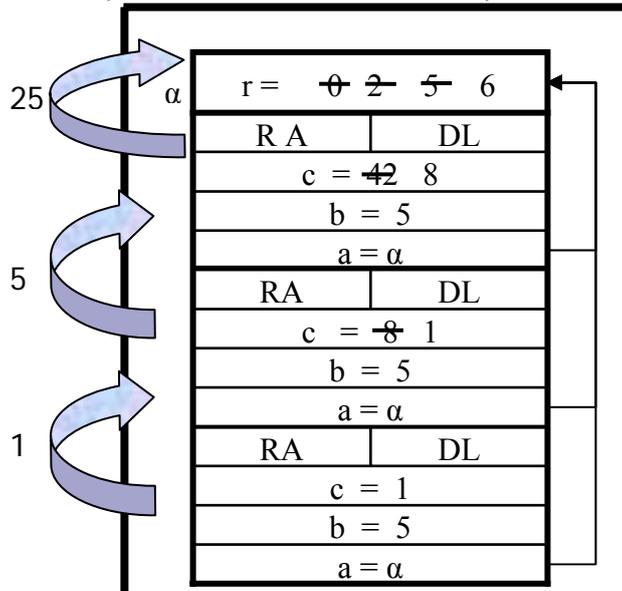
while(str1[i]!='\0'){
    j=0;
    occCharCorrente=0;
    while(str2[j]!='\0'){
        if(str1[i]==str2[j]) occCharCorrente++;
        j++;
    }
    if(occCharCorrente>occCharC){
        occCharC=occCharCorrente;
        charC=str1[i];
    }
    i++;
}
return charC;
}

int main(){
    char piuFrequente;
    char str1[]="abcdefghilmno";
    char str2[]="aabbdbdeh";
    piuFrequente=fun(str1,str2);
    printf("piuFrequente %c\n",piuFrequente);
    return 0;
}

```

Esercizio 4

La funzione effettua una divisione; come risultato restituisce 25, r assume il valore 6



Esercizio 5

Sottrazione dei moduli (numero maggiore meno numero minore)

```

1000111-   71-
0011001=   25=
-----
0101110    46

```

Aggiunta del segno
10101110

Esercizio 6

Le istruzioni sono corrette. s2 = s1 assegna a s2 il valore dell'indirizzo del primo elemento di s1. *(s2+1)='z' assegna al secondo elemento di s2 il valore 'z'. Infine la printf() stampa su stdout pzpoo, visto che dopo il primo assegnamento, le due variabili puntano allo stesso indirizzo di memoria.