

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 B

ESERCIZIO 1 (12 punti)

Un centro di meteorologia vuole effettuare in modo automatizzato statistiche sull'umidità registrata quotidianamente nell'arco di un mese. In particolare, vuole ottenere il tasso di umidità media (non considerando il giorno meno umido del mese) e la data del giorno meno umido col relativo tasso di umidità. A tale scopo si realizzi:

1) una funzione

```
int    mediaUmidita(float    umidita[],    int    length,    float*  
minUmiditaTasso, int* minUmiditaGiorno)
```

che noto il numero di giorni presi in esame **length** e i tassi di umidità **umidita**, restituisca l'umidità media (escludendo dal calcolo il giorno meno umido) come **int**. Inoltre, tramite i parametri **minUmiditaTasso** e **minUmiditaGiorno** la funzione deve restituire rispettivamente il tasso di umidità e la data del giorno meno umido, assumendo che il giorno 0 corrisponda al primo giorno del mese, il giorno 1 al secondo, e così via. In relazione al parametro **umidita** si noti che l'umidità presente in **umidita[0]** corrisponde al primo giorno del mese, l'umidità in **umidita[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. Si controlli che **G** abbia valore tra 28 e 31 compresi; in caso contrario, si richieda nuovamente il numero di giorni;
- b) chieda all'utente di inserire **G** tassi di umidità in ordine dal primo all'ultimo giorno del mese, controllando che i tassi di umidità inseriti abbiano valori tra 0 e 100 compresi. In caso contrario, si richieda nuovamente l'umidità per quel giorno;
- c) richiami opportunamente la funzione **mediaUmidita(...)**;
- d) stampi l'umidità media restituita da **mediaUmidita (...)**, il tasso di umidità e la data del giorno meno umido. **(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 *str1, char *str2)` che, ricevuti come parametri in ingresso due stringhe ben formate `str1` e `str2`, restituisca come valore il carattere di `str2` che compare più volte in `str1` (si assuma che in `str2` un carattere possa comparire al più una volta). Ad esempio, la chiamata `fun("Informatica", "Ciao")` deve restituire `'a'`, che compare due volte. Si proponga una possibile funzione chiamante.

ESERCIZIO 4 (4 punti)

Data la funzione:

```

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

```

e la funzione chiamante:

```

int main(){
    int r=0;
    fun(16,3,&r);
    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:

59-73

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

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

int mediaUmidita(float umidita[], int length, float* minUmiditaTasso, int*
minUmiditaGiorno){
    int i; float somma=0;
    *minUmiditaGiorno=0;
    *minUmiditaTasso=umidita[0];

    for(i=1;i<length;i++){
        if(umidita[i]<*minUmiditaTasso){
            somma=somma+*minUmiditaTasso;
            *minUmiditaTasso=umidita[i];
            *minUmiditaGiorno=i;
        }
        else somma=somma+umidita[i];
    }
    return (int)(somma/((float)(length-1)));
}

int main(){
    float umidita[DIM], minUmiditaTasso;
    int minUmiditaGiorno, media, giorni, i;

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

    for(i=0;i<giorni;i++){
        do { printf("Umidita' giorno %d?\n",i);
            scanf("%f",&umidita[i]);
        } while(umidita[i]<0||umidita[i]>100);
    }
    media=mediaUmidita(umidita, giorni, &minUmiditaTasso, &minUmiditaGiorno);
    printf("media %d; meno umido %f il %d\n",media,minUmiditaTasso,minUmiditaGiorno);
    return 0;
}
```

Esercizio 2

c a c b

d d 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 minore 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* str1, char* str2){
    int occCharCorrente, occCharC=0, i=0, j;
    char charC='';

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

```

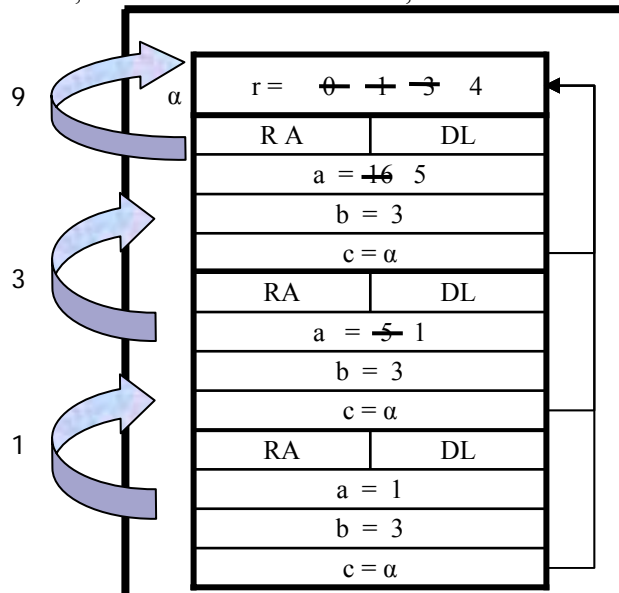
        occCharC=occCharCorrente;
        charC=str2[i];
    }
    i++;
}
return charC;
}

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

```

Esercizio 4

La funzione effettua una divisione; come risultato restituisce 9, r assume il valore 4.



Esercizio 5

Sottrazione dei moduli (numero maggiore meno numero minore)

```

1001001-    73-
0111011=    59=
-----
0001110    14

```

Aggiunta del segno
10001110

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)