

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 E

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 freddo del mese in esame) e la data del giorno più freddo con la relativa temperatura. A tale scopo si realizzi:

1) una funzione

```
float mediaTemp(int temp[], int length, int* piuFreddoTemp, int* piuFreddoGiorno)
```

che noto il numero di giorni presi in esame **length** e le temperature **temp**, restituisca la temperatura media (escludendo dal calcolo il giorno più freddo) come **float**. Inoltre, tramite i parametri **piuFreddoTemp** e **piuFreddoGiorno** la funzione deve restituire rispettivamente la temperatura e la data del giorno più freddo, assumendo 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 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** 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 richieda nuovamente all'utente di inserire la temperatura,

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

d) stampi la temperatura media restituita da **mediaTemp(...)**, i gradi e la data del giorno più freddo. **(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 *a, char *b)` che, ricevuti come parametri in ingresso due stringhe ben formate `a` e `b`, restituisca come valore di ritorno un `int` rappresentante la somma totale di occorrenze di ogni carattere di `b` in `a`. Ad esempio, la chiamata `fun("Pollo", "Pluto")` deve restituire 5 (1 occorrenza di 'P', 2 occorrenze di 'l', 2 occorrenze di 'o').

Si proponga una possibile funzione chiamante.

ESERCIZIO 4 (4 punti)

Data la funzione:

```

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

```

e la funzione chiamante:

```

int main(){
    int r=0;
    fun(&r,9,4);
    return 0;
}

```

mostrare la sequenza dei record di attivazione.

ESERCIZIO 5 (4 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 ::= a | b | c
E ::= 1 | 2 | 3 | 4

```

La stringa "23a4c14" 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

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

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

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

int main(){
    int temp[DIM];
    int piuFreddoTemp, piuFreddoGiorno, giorni, i;
    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, &piuFreddoTemp, &piuFreddoGiorno);
    printf("media %f; più freddo %d il %d", media, piuFreddoTemp, piuFreddoGiorno);
    return 0;
}
```

Esercizio 2

0.2 4.0 2.1 -17.0
15.0 3.0 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 maggiore 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* a, char* b){
    int totale=0, i=0, j;

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

```

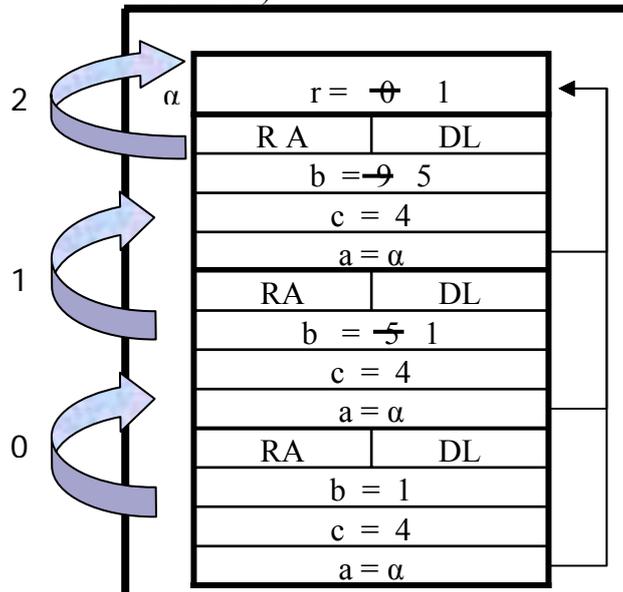
        }
        j++;
    }
    i++;
}
return totale;
}

int main(){
    int occorrenze=0;
    char a[]="abcdefghilmno";
    char b[]="aabbdde";
    occorrenze=fun(a,b);
    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 1 (resto della divisione intera).



Esercizio 5

S -> BC -> EBC -> 2BC -> 2EBC -> 23BC -> baDEC -> 23aEC -> 23a4C -> 23a4DCE -> 23a4cCE -> 23a4cEE -> 23a4c1E -> 23a4c14

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.