Compito A
Sintesi (main)

a) chieda all'utente il numero di giorni G del mese preso in esame e che controlli che G abbia valore tra 28 e 31 compresi...

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...

```
#define DIM 31;
/* 31 è il massimo numero di giorni */
int main(){
        int temp[DIM];
        int piuCaldoTemp, piuCaldoGiorno;
        int i, giorni; float media;

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

La dimensione di un array deve essere nota al compilatore, quindi deve essere una costante
Errori gravi (ma non rari) da NON fare:

```
int n;                  int n[ ];
scanf("%d",&n);         …
int temp[n];
…
```

|| or logico
&& and logico

Bisogna richiedere *giorni* valori (non *DIM* valori)
NB: il vettore *temp* ha dimensione fisica *DIM* e dimensione logica *giorni*

Compito A
Sintesi (main)

a) chieda all'utente il numero di giorni G del mese preso in esame e che controlli che G abbia valore tra 28 e 31 compresi. In caso contrario...

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...

```
        ...
        int temp[DIM];
        int i, giorni;
        ...
        do {
                printf("Numero 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) );
        }
        …
```

Alla scanf() bisogna passare l'indirizzo della variabile, non il suo valore.
scanf("%d", **giorni**); // ERRATO
scanf("%d", **&giorni**); // Corretto

|| or logico
&& and logico

Compito A
Sintesi (main)

c) richiami opportunamente la funzione mediaTemp(...) e infine

d) stampi la temperatura media restituita da mediaTemp(...), i gradi e la data del giorno più caldo

Il primo parametro della funzione mediaTemp è *int temp[ ]*, ovvero un puntatore ad un array di int. Di conseguenza il main() deve passare alla funzione un indirizzo che punti ad un array di int. Ciò è possibile con ***temp*** oppure ***&temp[0]***: **entrambe rappresentano l'indirizzo del primo elemento del vettore *temp*.**
NB: passare *temp[ ]* produce un errore in fase di compilazione (errore sintattico)

```
        …
        int temp[DIM];
        …
        media=mediaTemp(temp, giorni, &piuCaldoTemp, &piuCaldoGiorno);
        printf("media %f; più caldo %d il %d\n",media, piuCaldoTemp, piuCaldoGiorno);
        return 0;
}
```

Compito A
Sintesi (main)

c) richiami opportunamente la funzione mediaTemp(...) e infine

d) stampi la temperatura media restituita da mediaTemp(...), i gradi e la data del giorno più caldo

```
int piuCaldoTemp, piuCaldoGiorno;
…
media=mediaTemp(temp, giorni, &piuCaldoTemp, &piuCaldoGiorno);
```

Alcuni studenti hanno scritto

```
int *piuCaldoTemp,*piuCaldoGiorno;
printf("media %f; più caldo %d il %d\n",media, piuCaldoTemp,
    piuCaldoGiorno);
```

Il compilatore non trova errori, poiché la funzione mediaTemp viene chiamata in modo sintatticamente corretto.
Ma qual è il valore di **piuCaldoTemp* e **piuCaldoGiorno*? A che cosa puntano?
Puntano ad una cella di memoria **casuale**, che potrebbe essere anche all'esterno dello spazio di indirizzamento del processo in esecuzione. **Leggere** il valore di una cella casuale **non ha senso**, **scrivere** un valore in una cella casuale **è pericoloso** (potrebbe essere utilizzata per la memorizzazione di un'altra variabile, il cui valore viene sovrascritto!)

Compito A
Sintesi (funzione)

[...] noto il numero di giorni presi in esame length e le temperature temp, restituisca la temperatura media (escludendo dal calcolo il giorno più caldo) come float e tramite i parametri piuCaldoTemp e piuCaldoGiorno rispettivamente la temperatura e la data del giorno più caldo

```
float mediaTemp(int temp[], int length, int* piuCaldoTemp, int* piuCaldoGiorno){
        int i;
        int somma=0;
        *piuCaldoGiorno=0;
        *piuCaldoTemp=temp[0];

        ...
}
```

Una variabile dichiarata ma non inizializzata possiede un valore casuale
Esempio di codice errato:

```
int somma;
somma = somma + 5;
```

Qual è il valore di somma?

Compito A
Sintesi (funzione)

[...] noto il numero di giorni presi in esame length e le temperature temp, restituisca la temperatura media (escludendo dal calcolo il giorno più caldo) come float e tramite i parametri piuCaldoTemp e piuCaldoGiorno rispettivamente la temperatura e la data del giorno più caldo


```
float mediaTemp(int temp[], int length, int* piuCaldoTemp, int* piuCaldoGiorno){
        ...
        for(i=1;i<length;i++){
                if(temp[i]>*piuCaldoTemp){
                        somma=somma+*piuCaldoTemp;
                        *piuCaldoTemp=temp[i];
                        *piuCaldoGiorno=i;
                }
                else somma=somma+temp[i];
        }
        return somma/((float)(length-1));
}
```

Non necessari due o più cicli. È sufficiente un unico ciclo in cui si effettua la sommatoria e si ricerca il giorno più caldo del mese

Compito A
Analisi

```
#include <stdio.h>
#define DIM 4

int main(){
        int i=0;
        …
        for(;i<DIM;i++) printf("%f ",f1[i]);
        printf("\n");

        for(;i>0;i--) printf("%f ",f2[i-1]);
        printf("\n");
        return 0;
}
```

La variabile **i** non viene inizializzata nel **for**: ciò non è un problema dato che è già stata inizializzata al momento della dichiarazione

Alla fine del primo ciclo la variabile **i** ha valore 4
All'inizio del secondo ciclo la variabile **i** ha ancora valore 4

Alla fine del secondo ciclo la variabile **i** ha valore **0**

Compito A
Sintesi (stringhe)

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 **str1** in **str2**.

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

Stringa ben formata equivale a dire array di char con **ultimo elemento** il carattere di terminazione **'\0'**

Compito A
Sintesi (stringhe)

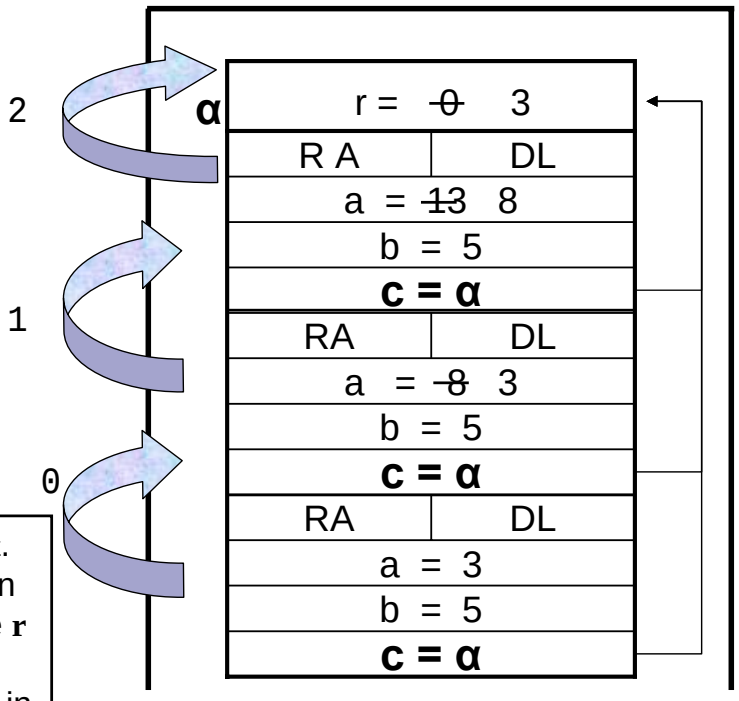
Si scriva una funzione iterativa **int fun(char *str1, char *str2)** che, ricevuti come parametri in ingresso due **stringhe ben formate** **str1** e **str2**, restituisca come valore di ritorno un **int** rappresentante la somma totale di occorrenze di ogni carattere di **str1** in **str2**.

> La funzione **fun()** si aspetta come parametri due variabili di tipo puntatore a char, ovvero l'indirizzo di due variabili di tipo char
> **str1** e **str2** del main sono due variabili di tipo puntatore a char
> Passare **&str1** e **&str2** vuol dire passare l'indirizzo di due variabili di tipo puntatore a char, e NON l'indirizzo di due variabili di tipo char

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

Compito A
Record di Attivazione

```
int fun(int a, int b, int* c){
        if(a>=b){
                a=a-b;
                return 1+fun(a,b,c);
        }
        else {
                *c=a; return 0;
        }
}
int main(){
        int r=0; fun(13,5,&r);
        return 0;
}
```



> La variabile **c** è un puntatore ad **int**.
> Il suo valore è l'indirizzo di un **int**, in particolare l'indirizzo della variabile **r** dichiarata nel main.
> Assegnare a **c** il valore di un intero in un record di attivazione (ad esempio ***c=4**) non è corretto