

Allocazione

L'allocazione statica: e' quella con cui sono memorizzate le variabili globali e le variabili statiche, queste vengono allocate all'avvio nel segmento dati da exec, con le distizioni che abbiamo visto.

L'allocazione automatica: argomenti di una funzione e variabili locali esistono solo per la durata della funzione. Lo spazio per queste variabili viene allocato nello stack quando viene eseguita la funzione e liberato quando si esce dalla medesima.

L'allocazione dinamica: non e' di per se prevista direttamente dal linguaggio C, ma la libc mette a disposizione alcune funzioni utilizzabili allo scopo. Di certo tutti conoscono malloc, calloc, realloc e free. Funzioni usate per allocare dinamicamente memoria (heap).

Variabili

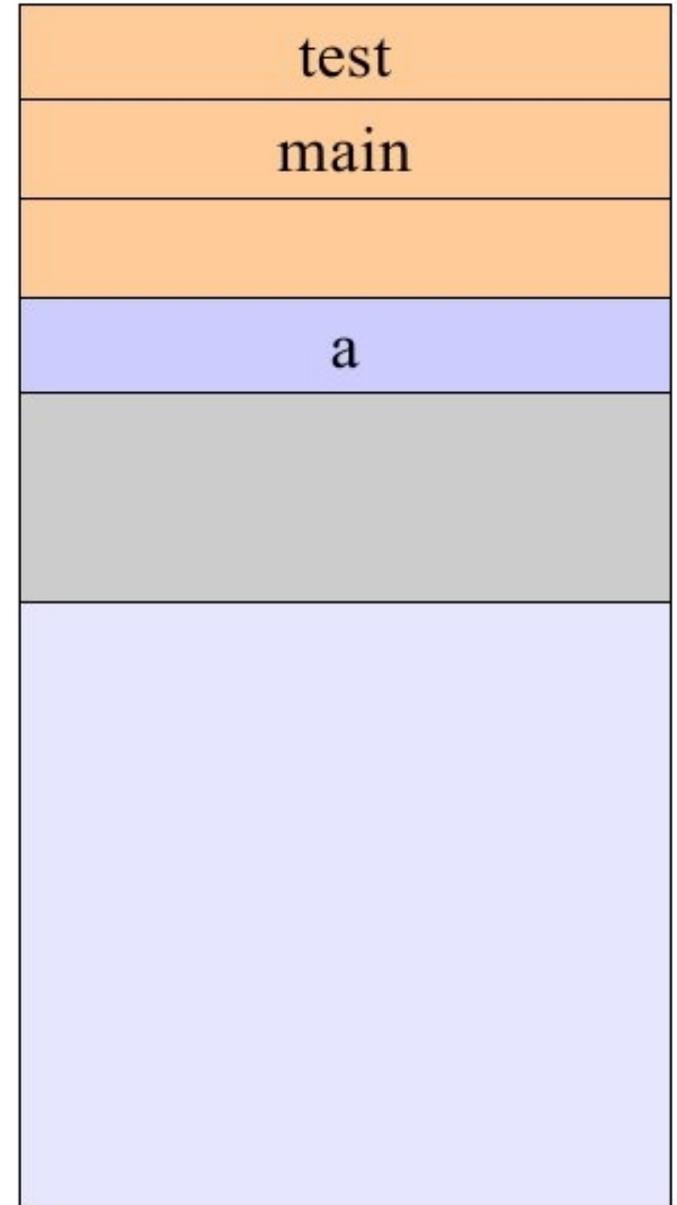
```
int a=3;
void test( )
{
    int b=2;
}
int main( )
{
    int c=1;
    test();
}
```



Codice

Dati

Stack
↑



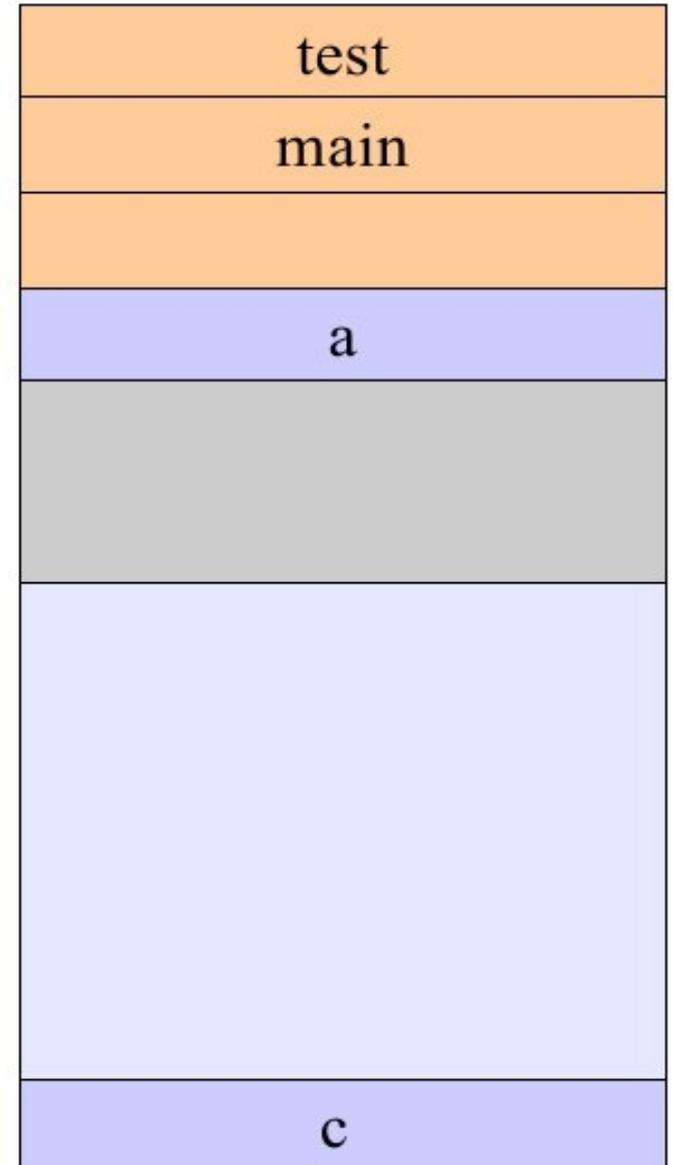
Variabili

```
int a=3;
void test( )
{
    int b=2;
}
int main( )
{
    int c=1;
    test();
}
```

Codice

Dati

Stack



Variabili

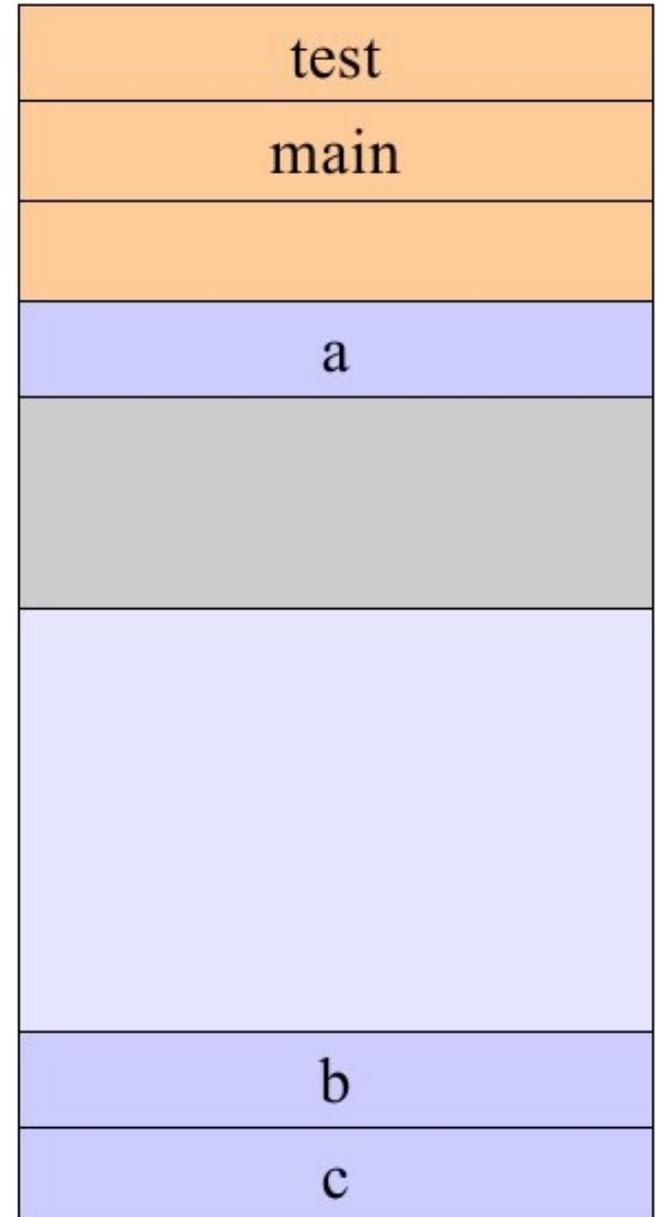
```
int a=3;
void test( )
{
    int b=2;
}
int main( )
{
    int c=1;
    test();
}
```



Codice

Dati

Stack

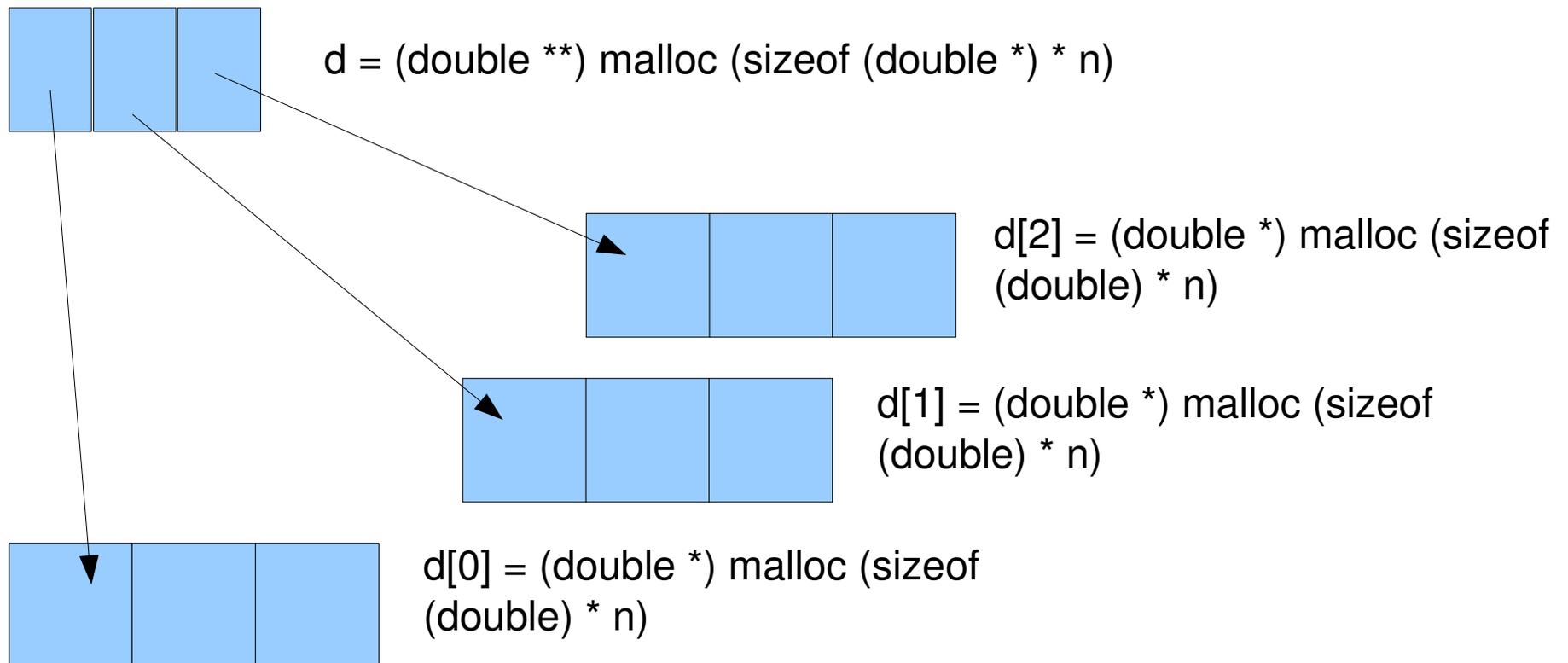


Esercizio

Scrivere un programma che allochi spazio necessario per contenere una matrice $N \times N$ di double. Tale programma deve inizializzare anche la matrice con numeri random. (N opzione da linea di comando) ([soluzione2.c](#))

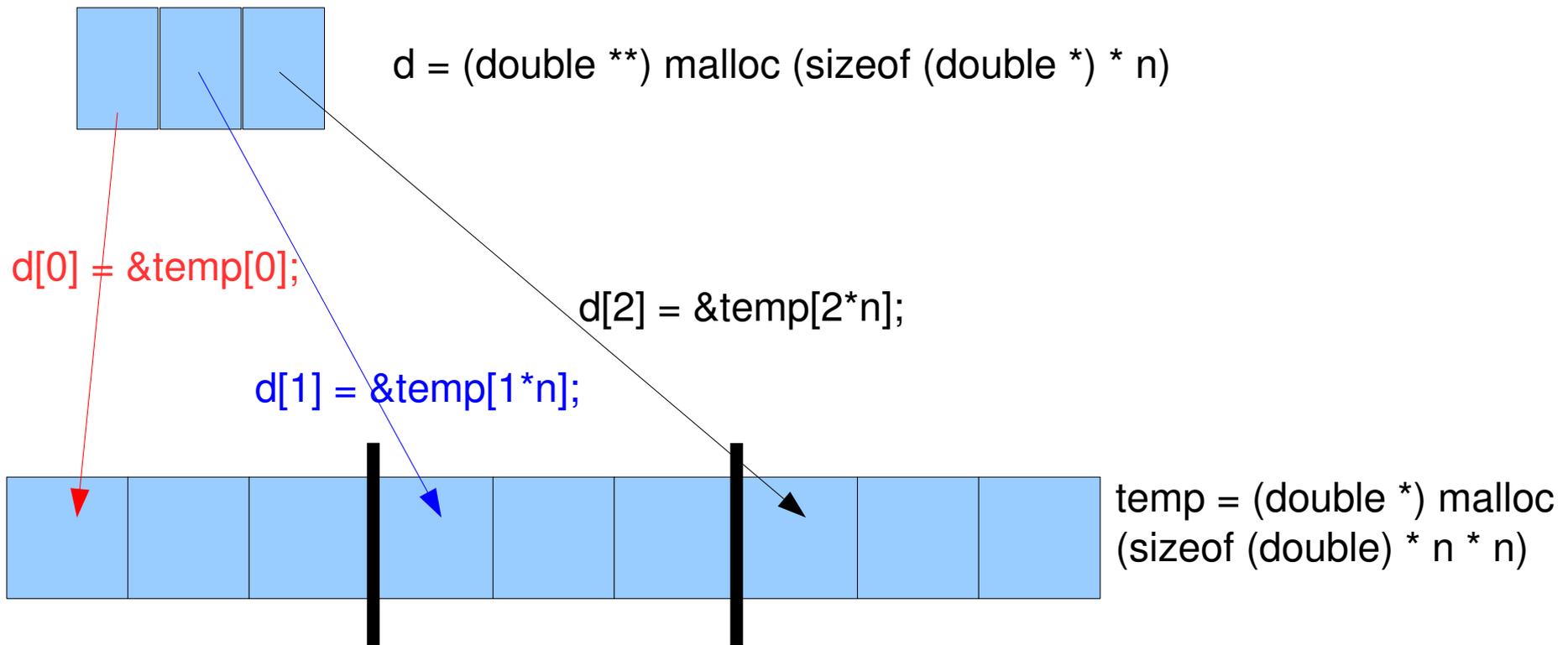
Allocazione dinamica della memoria

Vedi `soluzione2.c` (primo)



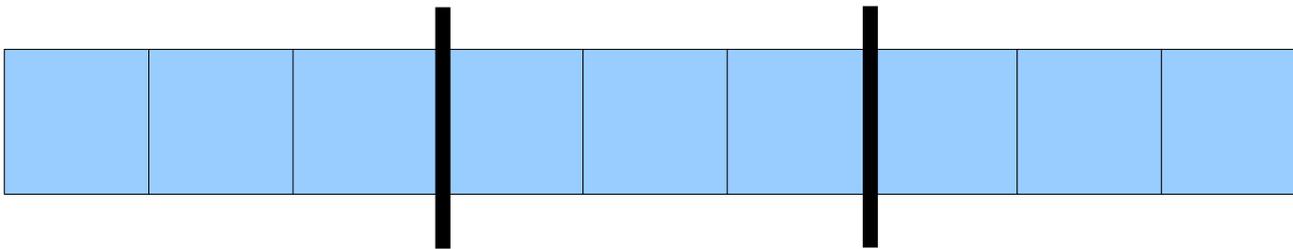
Allocazione dinamica della memoria

Vedi soluzione2.c (secondo)



Allocazione dinamica della memoria

Vedi soluzione2.c (terzo)



```
d = (double *) malloc (sizeof (double) * n * n)
```