

Titolo unità didattica: Allocazione Dinamica della memoria

[17]

Titolo modulo : Allocazione Dinamica della memoria in C
[01-C]

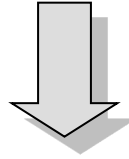
Le function C per l'allocazione dinamica della memoria

Argomenti trattati:

- ✓ le function C `malloc()` e `calloc()`
- ✓ `free()`
- ✓ `realloc()`

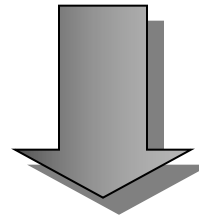
Prerequisiti richiesti: AP-05-*-C, AP-16-01-T

Organizzazione della memoria



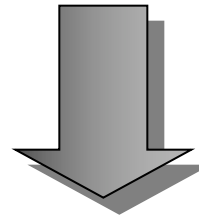
Tipo di memoria	Descrizione
Stack	È una parte <i>statica</i> , che contiene tutto ciò che sappiamo sarà allocato di certo (tutto ciò che è dichiarato nel codice quindi, come una variabile int)
Heap	È una parte <i>dinamica</i> , in cui la dimensione degli elementi può cambiare a “runtime” ovvero durante l’esecuzione del programma

Funzioni per gestire la memoria dinamica



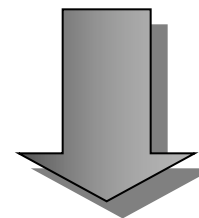
Allocazione

malloc()
calloc()



Liberare memoria

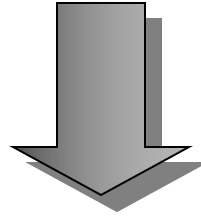
free()



Modificare spazio
allocato

realloc()

Usare una libreria



malloc.h

prototipi

```
void * malloc ( dim_totale );
```

```
void * calloc ( int num_elementi , int  
dim_elemento );
```

esempi

```
int *v;  
v = (int *) malloc(n * sizeof(int));
```

```
v = (int *) calloc(n, sizeof(int));
```

prototipi

```
void free (void ptr) ;
```

```
void realloc (void ptr , int dim_totale ) ;
```

```
#include <stdio.h>
#include <stdlib.h>
#include <malloc.h>

int main()
{
    int numero, *array, i; int allocati;
    numero = 100;
    printf("Numero di elementi dell'array:
    %d\n", numero);

    array = (int *) malloc(sizeof(int) *
    numero);

    return 0;
}
```