

LEGGETE LA GUIDA PER LA CREAZIONE DEI PROGETTI E PER IL DEBUGGING!

*Gli esercizi seguenti devono essere risolti, compilati e testati utilizzando il debugger. Per ognuno si deve realizzare una funzione `main()` che ne testi il funzionamento. **Fate progetti diversi per ogni esercizio.***

Esercizio 1

Creare un file “main.c”. Nel file, si realizzi in linguaggio C la funzione corrispondente alla seguente dichiarazione:

```
extern uint32_t *crea_int_vec (size_t n);
```

La funzione deve restituire un puntatore al primo di **n** interi senza segno a 32 bit allocati dinamicamente attraverso la funzione malloc.

I valori appena allocati devono essere inizializzati con il relativo indice. Ad esempio: il primo dovrà essere inizializzato a 0, il secondo a 1, il terzo a 2 e così via.

Esercizio 2

Creare un file “main.c”. Nel file, si realizzi in linguaggio C la funzione corrispondente alla seguente dichiarazione:

```
extern void distruggi_vec (uint32_t *p);
```

La funzione riceve come parametro un puntatore al primo di una serie di interi senza segno a 32 bit. Il suo scopo è quello di liberare la memoria che era stata precedentemente allocata dinamicamente.

Esercizio 3

Creare un file “main.c”. Nel file, si realizzi in linguaggio C la funzione corrispondente alla seguente dichiarazione:

```
extern double media (const double *x, unsigned int n);
```

La funzione deve calcolare il valore medio (cercate la formula su Wikipedia se non la ricordate) del vettore **x** contenente **n** elementi. Se **n** è 0 la media è zero.

Nel testare il programma si scriva una funzione main in cui il vettore deve essere allocato dinamicamente utilizzando le funzioni malloc e free.

Esercizio 4

Sia data la seguente funzione main:

```
#include <stdlib.h>
#include <stdint.h>

void doppio (uint32_t *vett, size_t size)
{
    /* completare la funzione doppio */
}

int main(void)
{
    size_t n = 20;

    uint32_t *vettore = crea_int_vec(n);

    doppio(vettore, n);

    return 0;
}
```

Riempire la definizione della funzione “doppio” sapendo che deve prendere un vettore di interi senza segno a 32 bit e per ognuno dei suoi elementi deve calcolarne il doppio e memorizzarlo all’interno del vettore nella stessa posizione.

Per la funzione crea_int_vec si può usare quella già creata nell’esercizio 1.

Esercizio 5

Sia data la seguente funzione main:

```
#include <stdlib.h>
#include <stdint.h>

uint32_t *somme_2a2 (uint32_t *vett, size_t size)
{
    /* completare la definizione di somme_2a2 */
}

int main(void)
{
    size_t n = 20;

    uint32_t *vettore = crea_int_vec(n);

    uint32_t *somme = somme_2a2(vettore, n);

    return 0;
}
```

La funzione somme_2a2 restituire un vettore allocato dinamicamente di (size / 2) interi senza segno a 32 bit che conterranno le somme due a due dei valori contenuti nel vettore passato come parametro.

Ad esempio, nel primo valore del vettore somme dovrà esserci la somma del primo e del secondo valore di vett, nel secondo valore di somme ci sarà la somma del terzo e del quarto valore di vett, e così via.

Se size è dispari la funzione somme_2a2 salta l’ultimo valore del vettore passato.