

BUILD WEEK 2 GIORNO 3

Andiamo ad analizzare il codice in C fornito

```
#include <stdio.h>

int main () {
    int vector [10], i, j, k;
    int swap_var;

    printf ("Inserire 10 interi:\n");
    for ( i = 0 ; i < 10 ; i++)
    {
        int c= i+1;
        printf("[%d]:", c);
        scanf ("%d", &vector[i]);
    }

    printf ("Il vettore inserito e':\n");
    for ( i = 0 ; i < 10 ; i++)
    {
        int t= i+1;
        printf("[%d]: %d", t, vector[i]);
        printf("\n");
    }

    for (j = 0 ; j < 10 - 1; j++)
```

```

    {
        for (k = 0 ; k < 10 - j - 1; k++)
        {
            if (vector[k] > vector[k+1])
            {
                swap_var=vector[k];
                vector[k]=vector[k+1];
                vector[k+1]=swap_var;
            }
        }
    }
    printf("Il vettore ordinato e':\n");
    for (j = 0; j < 10; j++)
    {
        int g = j+1;
        printf("[%d]:", g);
        printf("%d\n", vector[j]);
    }

    return 0;
}
```

A primo impatto il codice richiede di inserire 10 numeri interi ed inserirli uno ad uno, una volta inseriti tutti e 10 i valori il programma ristampa i valori inseriti ordinandoli

Dopo l'esecuzione ci accorgiamo che effettivamente il programma esegue ciò che pensavamo richiedendo appunto 10 numeri interi

La richiesta è quella di andare a modificare il codice per far sì che una volta inserito un valore errato venga prodotto un **Errore di segmentazione** che ha luogo quando un programma tenta di accedere ad una posizione di memoria alla quale non gli è permesso accedere, oppure quando tenta di accedervi in una maniera che non gli è concessa.

Procediamo con le modifiche:

```
#include <stdio.h>

int main () {

int vector [10], i, j, k;
int swap_var;

printf ("Inserire 10 interi:\n");

for ( i = 0 ; i < 10 ; i++)
{
    int c= i+1;
    printf("[%d] ", c);
    scanf ("%d", &vector[i*900]);
}

printf ("Il vettore inserito e':\n");
for ( i = 0 ; i < 10 ; i++)
{
    int t= i+1;
    printf("[%d]: %d", t, vector[i]);
    printf("\n");
}

for (j = 0 ; j < 10 - 1; j++)
```

Andando ad inserire un valore moltiplicato per il vettore questo genererà l'errore di segmentazione

```
[parrot@parrot]~$ cd Desktop
[parrot@parrot]~/Desktop$ gcc BW_D3_BOF.c -o BW_D3_BOF
[parrot@parrot]~/Desktop$ ./BW_D3_BOF
Inserire 10 interi:
[1]:123456789012
[2]:1234567890987
[3]:1234567654321
[4]:9876543217654
[5]:9876543298765
[6]:987654387654
[7]:8765432345678
[8]:1234567890987654
[9]:123456789876543
[10]:12345678987654
Il vettore inserito e':
[1]: -1097262572
[2]: 0
[3]: 1263814400
Segmentation fault
```