

# BOF

```
#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;

}
```

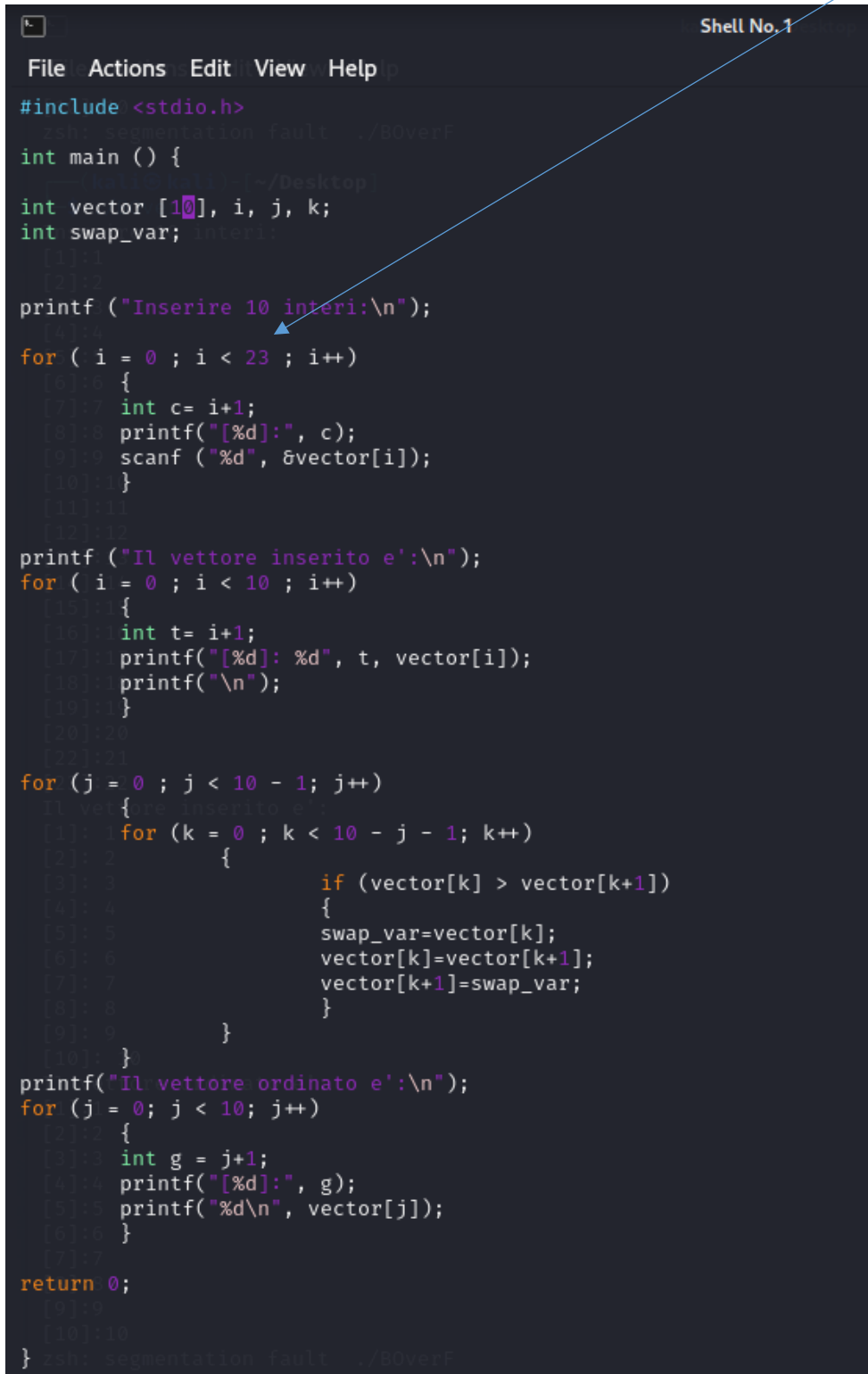
---

A prima vista, il codice proposto consiste in 3 array di vettori, ciascuno composto da 10 valori. L'utente inserisce questi valori numerici interi, andando a riempire il primo vettore. Successivamente vengono

dati in output gli altri 2 vettori, uno speculare al vettore coi valori inseriti dall'utente, e l'ultimo ordina i numeri in maniera crescente.

Per creare un errore di segmentazione, si può procedere in due modi, o **aumentare il numero di valori che l'utente può inserire**, andando a sovraccaricare il tutto, o **diminuire la grandezza degli array**.

Per la dimostrazione scelgo il primo metodo. Andiamo ad esempio a far inserire 23 valori all'utente, e vediamo il risultato.



```
Shell No. 1
File Actions Edit View Help
#include <stdio.h>
zsh: segmentation fault ./80verf
int main () {
    printf("Inserire 10 interi:\n");
    for ( i = 0 ; i < 23 ; 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;
} zsh: segmentation fault ./80verf
```

```
kali@kali: ~/Desktop
File Actions Edit View Help
[10]:10
zsh: segmentation fault ./BOverF

(kali@kali)-[~/Desktop]
$ ./BOverF
Inserire 10 interi:
[1]:1
[2]:2
[3]:3
[4]:4
[5]:5
[6]:6
[7]:7
[8]:8
[9]:9
[10]:10
[11]:11
[12]:12
[13]:13
[14]:14
[15]:15
[16]:16
[17]:17
[18]:18
[19]:19
[20]:20
[22]:21
[23]:22
Il vettore inserito e':
[1]: 1
[2]: 2
[3]: 3
[4]: 4
[5]: 5
[6]: 6
[7]: 7
[8]: 8
[9]: 9
[10]: 10
Il vettore ordinato e':
[1]:1
[2]:2
[3]:3
[4]:4
[5]:5
[6]:6
[7]:7
[8]:8
[9]:9
[10]:10
zsh: segmentation fault ./BOverF

(kali@kali)-[~/Desktop]
$
```

Come previsto, il programma va in segmentation fault.

Ora andiamo a modificare il programma, inserendo controlli dell'input dell'utente, e dando la possibilità a quest'ultimo di scegliere tra il programma con segmentation fault, e quello corretto.

In questa maniera andiamo ad evitare che l'utente sbagli ad inserire un simbolo o una lettera invece che un numero, e gli diamo la possibilità di usare il programma corretto o quello con errore di segmentazione.

```

#include <stdio.h>

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

    //menù con scelta programma
    printf("Benvenuto, i NetRaiders sono qui per aiutarti, o forse no!\n");
    printf("Scegli il programma da eseguire:\n");
    printf("1. Programma corretto\n");
    printf("2. Programma con errore di segmentazione\n");
    printf("Scelta: ");
    scanf("%d", &scelta);

    if (scelta == 1) {
        printf("Esecuzione del programma corretto...\n");

        printf("Inserire 10 interi:\n");
        for (i = 0; i < 10; i++) { //// inserimento esclusivo numero, in caso di inserimento lettera o simbolo, da messaggio errore e non va avanti con l'array
            printf("[%d]:", i + 1);
            while (scanf("%d", &vector[i]) != 1) {
                printf("Input non valido. Devi inserire un numero intero.\n");
                while (getchar() != '\n'){}
                printf("[%d]:", i + 1);
            }
        }

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

        // Ordinamento del vettore
        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]: %d\n", g, vector[j]);
        }
    }
}

```

```

    }
    else if (scelta == 2) {
        printf("Esecuzione del programma con errore di segmentazione...\n");

        printf("Inserire 23 numeri interi:\n");
        for (i = 0; i < 23; i++) { // Provocare un errore di segmentazione inserendo più interi del consentito
            printf("[%d]:", i + 1);

            // inserimento esclusivo numero
            while (scanf("%d", &vector[i]) != 1) {
                printf("Input non valido. Devi inserire un numero intero.\n");
                while (getchar() != '\n'){}
                printf("[%d]:", i + 1);
            }
        }

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

        // Ordinamento del vettore
        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]: %d\n", g, vector[j]);
        }
    }
    else {
        printf("Scelta non valida.\n");
        return 1; // Uscita con codice di errore
    }

    return 0;
}

```