S2 L5

Svolgimento esercizi del 1/12/23

Giulia Salani

Progetto

Consegna

Traccia:

Per agire come un Hacker bisogna capire come pensare fuori dagli schemi.

L’esercizio di oggi ha lo scopo di allenare l’osservazione critica.

Dato il codice in allegato, si richiede allo studente di:

-Capire cosa fa il programma senza eseguirlo

-Individuare dal codice sorgente le casistiche non standard che il programma non gestisce (esempio, comportamenti potenziali che non sono stati contemplati)

-Individuare eventuali errori di sintassi / logici

-Proporre una soluzione per ognuno di essi

Svolgimento

Nelle prossime slide ho diviso il codice in blocchi, per questioni di leggibilità.

Per ciascun blocco presento prima gli errori riscontrati sul vecchio codice (slide con le caselle di testo rosse) e subito a seguire le soluzioni individuate e il codice da me modificato (slide con le caselle di testo verdi, dove a sinistra è riportato il codice e a destra la spiegazione di ciò che ho fatto).

Alla fine della sezione ci sono due screen con il nuovo codice.

```c
#include <stdio.h>

void menu ();
void moltiplica ();
void dividi ();
void ins_string();


int main ()

{
        char scelta = {'\0'};
        menu ();
        scanf ("%d", &scelta);

        switch (scelta)
        {
                case 'A':
                moltiplica();
                break;
                case 'B':
                dividi();
                break;
                case 'C':
                ins_string();
                break;
        }

return 0;

}
```

ERRORI DI SINTASSI:

1) Presenza delle parentesi graffe in char scelta.
2) La variabile «scelta» non è un INT bensì una stringa quindi viene puntata con la lettera %s.

CASISTICHE NON CONTEMPLATE:

1) In SWITCH, il programma non contempla che l'utente possa scrivere «a», «b», «c» (ovvero, le lettere minuscole) e non contempla nemmeno che l'utente possa scrivere altro.
2) Al termine della sua esecuzione, il programma non dà all'utente l'opzione di ricominciare.

```c
#include <stdio.h>
#include <string.h>

void menu ();
void moltiplica ();
void dividi ();
void ins_string();


int main ()

{
        char scelta = 0;
        printf("Benvenuto o benvenuta!\n");
        printf("Sono un assistente digitale. Posso aiutarti a svolgere alcuni compiti!\n");

        do
        {
                menu ();
                scanf ("%s", &scelta);

                switch (scelta)
                {
                        case 'A':
                        moltiplica();
                        break;
                        case 'a':
                        moltiplica();
                        break;
                        case 'B':
                        dividi();
                        break;
                        case 'b':
                        dividi();
                        break;
                        case 'C':
                        ins_string();
                        break;
                        case 'c':
                        ins_string();
                        break;
                        default: printf("Scelta non valida. Puoi scegliere fra 'A', 'B' e 'C'. \n");
                }

                printf("\nPosso esserti ancora d'aiuto? (S/N): \n");
                scanf(" %c", &scelta);
                while(scelta != 'S' && scelta != 's' && scelta != 'N' && scelta != 'n')
                {
                        printf("Scelta non valida.\n");
                        scanf(" %c", &scelta);
                }

        } while (scelta == 'S' || scelta == 's');

        printf("\nAlla prossima!");

return 0;

}
```

SOLUZIONI DI SINTASSI:

1) Ho eliminato le graffe da char scelta.
2) In «scanf», ora la variabile puntata è indicata con la lettera corretta («%s»).

SOLUZIONI CASISTICHE NON CONTEMPLATE:

1) In SWITCH, ho aggiunto due tipologie di casi. Ora se l'utente si sbaglia e digita le lettere «a», «b», «c» minuscole il programma procede comunque al void selezionato. Ho inoltre aggiunto un messaggio in corrispondenza di «default»: se l'utente non scrive né A, né B, né C (maiuscole o minuscole) riceve un messaggio che gli comunica che la scelta non è valida e gli ricorda le uniche scelte che può eseguire.

2) Perché il programma ricominci, ho inserito un DO/WHILE. In questo modo, dopo l'esecuzione il programma chiede all'utente se ha ancora bisogno di lui. Finché la scelta cade su «S» (maiuscola o minuscola), il programma riparte. Appena la scelta diventa «N», il programma si ferma e saluta l'utente con un «Alla prossima!». Se la scelta non è né S, né N, l'utente viene informato che la scelta non è valida.

1) Data la modifica eseguita sopra, ho spostato il messaggio di benvenuto fuori dal loop. In questo modo compare all'utente solamente una volta, all'apertura del programma.

```
void moltiplica ()
{
        short int  a,b = 0;
        printf ("Inserisci i due numeri da moltiplicare:");
        scanf ("%f", &a);
        scanf ("%d", &b);

        short int prodotto = a * b;

        printf ("Il prodotto tra %d e %d e': %d", a,b,prodotto);
}
```

ERRORI DI SINTASSI:

1) È sempre buona norma assegnare alle variabili valore 0 in partenza; ma, così come è scritto ora, il valore 0 viene assegnato solamente a «b».
2) Nessuna delle righe di output presenta «\n» per migliorare la leggibilità.
3) Se «a» e «b» sono short int, non è corretto nel primo scanf «%f».

CASISTICHE NON CONTEMPLATE:

Che l'utente voglia moltiplicare numeri reali e non solamente interi.

```c
void menu ()
{
        printf ("\nCome posso aiutarti?\n");
        printf ("A >> Moltiplicare due numeri\nB >> Dividere due numeri\nC >> Inserire una stringa\n");
}


void moltiplica ()
{
        float a = 0;
        float b = 0;
        printf ("Inserisci il primo numero da moltiplicare:\n");
        scanf ("%f", &a);
        printf ("Inserisci il secondo numero da moltiplicare:\n");
        scanf ("%f", &b);

        float prodotto = a * b;

        printf ("Il prodotto tra %f e %f è: %f",a,b,prodotto);
}
```

SOLUZIONI DI SINTASSI:

1) Ora sia ad «a» che a «b» è assegnato valore 0.
2) Per migliorare la user experience e guidare meglio l'utente, ho separato in due scanf i numeri da moltiplicare.
3) Per migliorare la leggibilità, ho inserito alcuni newline «\n».

SOLUZIONI CASISTICHE NON CONTEMPLATE:

1) Le variabili ora sono FLOAT, quindi se l'utente volesse moltiplicare due numeri reali, può farlo.

```c
void dividi ()
{
        int  a,b = 0;
        printf ("Inserisci il numeratore:");
        scanf ("%d", &a);
        printf ("Inserisci il denumeratore:");
        scanf ("%d", &b);

        int divisione = a % b;

        printf ("La divisione tra %d e %d e': %d", a,b,divisione);
}
```

ERRORI DI SINTASSI:

1) È sempre buona norma assegnare alle variabili valore 0 ma, così come è scritto ora, il valore 0 viene assegnato solamente a «b».
2) Non è propriamente un errore ma inserire degli «\n» aiuta la leggibilità.

CASISTICHE NON CONTEMPLATE:

1) Che l'utente voglia dividere numeri reali e non solamente numeri interi.
2) Il risultato di una divisione potrebbe non essere un numero intero.
3) Che l'utente possa tentare di dividere per 0, operazione matematicamente impossibile.

```c
void dividi ()
{
        float a = 0;
        float b = 0;
        printf ("Inserisci il numeratore:\n");
        scanf ("%f", &a);
        printf ("Inserisci il denominatore:\n");
        scanf ("%f", &b);
        if (b != 0)
        {
        float divisione = a / b;
        printf ("La divisione tra %f e %f e': %f", a,b,divisione);
        }
        else
        {
        printf("In aritmetica non è possibile dividere per 0\n");
        }
}
```

SOLUZIONI DI SINTASSI:

1) Ho definito «a» e «b» uguali a zero in due righe separate.
2) Ho inserito «\n» per migliorare la leggibilità.
3) Non esiste la parola denumeratore, ho corretto con denominatore.

SOLUZIONI CASISTICHE NON CONTEMPLATE:

1) Le variabili «a» e «b» ora sono FLOAT, quindi se l'utente volesse dividere due fattori reali, può farlo.
2) Ho inserito un IF/ELSE così che, se l'utente tenta di dividere per 0, il programma comunica che non è un'operazione fattibile e non la fa partire.

```c
void ins_string ()
{
        char stringa[10];
        printf ("Inserisci la stringa:");
        scanf ("%s", &stringa);
}
```

CASISTICHE NON CONTEMPLATE:

L'utente potrebbe inserire una stringa troppo lunga.

```c
void ins_string ()
{
        char stringa[10];
        int c;
        while ( (c = getchar()) != '\n' && c != EOF) {}

        printf ("Inserisci la stringa (max 9 caratteri):\n");
        fgets (stringa, 10, stdin);
        printf ("La stringa: %s", stringa);
        printf (" è stata inserita correttamente.\n");

        while ( (c = getchar()) != '\n' && c != EOF) {}

}
```

SOLUZIONI CASISTICHE NON CONTEMPLATE:

Ho stabilito un limite per la stringa e, soprattutto, l'ho comunicato all'utente che altrimenti non potrebbe sapere perché la sua stringa viene troncata.
Nella restituzione della stringa all'utente, la stringa viene troncata dal programma al limite stabilito. A lui viene restituita solo la parte di stringa che sta dentro il limite.
Ho poi ripulito il buffer input per svuotarlo nel caso in cui l'utente effettivamente inserisca una stringa più lunga di quella che restituirà il programma.

Nuovo codice

```c
#include <stdio.h>
#include <string.h>

void menu ();
void moltiplica ();
void dividi ();
void ins_string();


int main ()

{
        char scelta = 0;
        printf("Benvenuto o benvenuta!\n");
        printf("Sono un assistente digitale. Posso aiutarti a svolgere alcuni compiti!\n");

        do
        {
                menu ();
                scanf ("%s", &scelta);

                switch (scelta)
                {
                        case 'A':
                        moltiplica();
                        break;
                        case 'a':
                        moltiplica();
                        break;
                        case 'B':
                        dividi();
                        break;
                        case 'b':
                        dividi();
                        break;
                        case 'C':
                        ins_string();
                        break;
                        case 'c':
                        ins_string();
                        break;
                        default: printf("Scelta non valida. Puoi scegliere fra 'A', 'B' e 'C'. \n");
                }

                printf("\nPosso esserti ancora d'aiuto? (S/N): \n");
                scanf(" %c", &scelta);
                while(scelta != 'S' && scelta != 's' && scelta != 'N' && scelta != 'n')
                {
                        printf("Scelta non valida.\n");
                        scanf(" %c", &scelta);
                }

        } while (scelta == 'S' || scelta == 's');

        printf("\nAlla prossima!");

return 0;

}
```

```c
void menu ()
{
        printf ("\nCome posso aiutarti?\n");
        printf ("A >> Moltiplicare due numeri\nB >> Dividere due numeri\nC >> Inserire una stringa\n");
}


void moltiplica ()
{
        float a = 0;
        float b = 0;
        printf ("Inserisci il primo numero da moltiplicare:\n");
        scanf ("%f", &a);
        printf ("Inserisci il secondo numero da moltiplicare:\n");
        scanf ("%f", &b);

        float prodotto = a * b;

        printf ("Il prodotto tra %f e %f è: %f",a,b,prodotto);
}


void dividi ()
{
        float a = 0;
        float b = 0;
        printf ("Inserisci il numeratore:\n");
        scanf ("%f", &a);
        printf ("Inserisci il denominatore:\n");
        scanf ("%f", &b);
        if (b != 0)
        {
        float divisione = a / b;
        printf ("La divisione tra %f e %f e': %f", a,b,divisione);
        }
        else
        {
        printf("In aritmetica non è possibile dividere per 0\n");
        }
}


void ins_string ()
{
        char stringa[10];
        int c;
        while ( (c = getchar()) != '\n' && c != EOF) {}

        printf ("Inserisci la stringa (max 9 caratteri):\n");
        fgets (stringa, 10, stdin);
        printf ("La stringa: %s", stringa);
        printf (" è stata inserita correttamente.\n");

        while ( (c = getchar()) != '\n' && c != EOF) {}

}
```

Compito

Consegna

Traccia:

Riprendete il codice del programma che avete scritto e pensiamo all'ottimizzazione del codice alla gestione delle situazioni non previste e facciamo le seguenti considerazioni:

-Cosa succede se l'utente inserisce una lettera diversa da A o B in fase di scelta iniziale? Il programma termina, ma non è una casistica che abbiamo gestito.

-Cosa succede se l'utente inserisce un nome che ha più caratteri della dimensione dell'array «nome» che abbiamo dichiarato inizialmente nella fase di avvio nuova partita? Riceveremo un errore (provate ad inserire una sequenza molto lunga di caratteri).

-Cosa succede se l'utente inserisce la lettera D per la risposta alle domande durante una partita? O un carattere numerico? Tutte queste situazioni vanno considerate in fase di programmazione in quanto errori logici o errori di mancata gestione di situazioni non standard potrebbero portare a bug nel codice che potrebbero essere sfruttati da un attaccante per prendere controllo dell'esecuzione del programma ed eseguire codice malevolo.

Traccia:

Riprendete il programma scritto in precedenza ed identificate tutte le casistiche non contemplate.

Provate a proporre un modello per gestirle modificando il codice sorgente del vostro programma.

Aiutatevi pure con le risorse online, piccolo aiuto: cercate come gestire in maniera sicura l'input dell'utente (soprattutto quando parliamo di stringhe).

Svolgimento

Kali Linux [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

kali@kali: ~/Desktop/Epicode_Lab

File Actions Edit View Help

GNU nano 7.2 MikeBongiorno.c

```c
#include <stdio.h>

char Iniziale()
{
    printf("Cosa vuoi fare?\n");
    printf("A) Nuova partita\n");
    printf("B) Esci dal gioco\n");

    char scelta;
    printf("Quindi?: ");
    scanf(" %c", &scelta);

    while (scelta != 'A' && scelta != 'a' && scelta != 'B' && scelta != 'b')
    {
        printf("Scelta non valida, riprova.");
        scanf("%c", &scelta);
    }

    if (scelta == 'A' || scelta == 'a')
    {
        printf("Hai scelto di iniziare una nuova partita. Partiamo!!!\n");
    }
    else
    {
        printf("Peccato! Avremmo potuto divertirci.\n");
    }

    return scelta;
}

void NuovaPartita(char *NomeGiocatore)
{
    printf("Come ti chiami?: ");
    scanf("%s", NomeGiocatore);
    printf("Quindi ti chiami: %s\n", NomeGiocatore);
}

void setDiDomande()
{
    char Domanda1[] = "In quale anno è stato presentato il primo cellulare della storia?\n";
    char Domanda2[] = "In quale anno fu iniziata la costruzione della Grande Muraglia Cinese?\n";
    char Domanda3[] = "In quale secolo fu inventato il denaro, sottoforma di moneta?\n";
    char Domanda4[] = "Quanto durò la guerra più breve del mondo?\n ";
    char Domanda5[] = "A quanti anni fa risale il letto più vecchio della storia? \n ";

    int Punteggio = 0;
    char Scelta;

    printf("Ottimo, possiamo partire\n");

    printf("———————————————————————— \n");
    printf("———————————————————————— \n");
```

In char Iniziale (), con un **WHILE**, si impedisce all'utente di procedere finché non dà una risposta valida.

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark M-] To Bracket M-Q Previous ^B Back ^← Prev Word ^A Home
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo M-6 Copy ^Q Where Was M-W Next ^F Forward ^→ Next Word ^E End

CTRL (DESTRA)

Search ITA 11:02 03/12/2023

Kali Linux [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

kali@kali: ~/Desktop/Epicode_Lab

File Actions Edit View Help

GNU nano 7.2 MikeBongiorno.c

```
    printf("———————————————————— \n");

    // PRIMA DOMANDA
    printf("Prima domanda: \n");
    printf(Domanda1);
    printf("A) 1990 ——  B) 1971 —— C) 1983  \n");
    printf("SCELTA: ");
    scanf(" %c", &Scelta);

    while (Scelta ≠ 'A' && Scelta ≠ 'B' && Scelta ≠ 'C')
    {
        printf("Scelta non valida. Riprova: ");
        scanf(" %c", &Scelta);
    }

    if (Scelta == 'A' || Scelta == 'B')
    {
        printf("Risposta errata D:\n");
        Punteggio--;
    }
    else
    {
        printf("Risposta corretta :D\n");
        Punteggio++;
    }
    // FINE PRIMA DOMANDA

    // SECONDA DOMANDA
    printf("Seconda domanda: \n");
    printf(Domanda2);
    printf("A) 1026dC ——  B) 215aC —— C) 215dC  \n");
    printf("SCELTA: ");
    scanf(" %c", &Scelta);

    while (Scelta ≠ 'A' && Scelta ≠ 'B' && Scelta ≠ 'C')
    {
        printf("Scelta non valida. Riprova: ");
        scanf(" %c", &Scelta);
    }

    if (Scelta == 'A' || Scelta == 'C')
    {
        printf("Risposta errata D:\n");
        Punteggio--;
    }
    else
    {
        printf("Risposta corretta :D\n");
        Punteggio++;
    }
    // FINE SECONDA DOMANDA
```

Anche in ciascuna delle domande, con un **WHILE**, si impedisce all'utente di procedere finché non dà una risposta valida.

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark M-] To Bracket M-Q Previous ^B Back ^◂ Prev Word ^A Home
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo M-6 Copy ^Q Where Was M-W Next ^F Forward ^▸ Next Word ^E End

CTRL (DESTRA)

Search

ITA 11:03 03/12/2023

Kali Linux [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

1 2 3 4 11:03

kali@kali: ~/Desktop/Epicode_Lab

File Actions Edit View Help

GNU nano 7.2 MikeBongiorno.c

```
    // TERZA DOMANDA
    printf("Terza domanda: \n");
    printf(Domanda3);
    printf("A) XXdC ——  B) IIdC —— C) VIII aC  \n");
    printf("SCELTA: ");
    scanf(" %c", &Scelta);

    while (Scelta ≠ 'A' && Scelta ≠ 'B' && Scelta ≠ 'C')
    {
        printf("Scelta non valida. Riprova: ");
        scanf(" %c", &Scelta);
    }

    if (Scelta == 'A' || Scelta == 'B')
    {
        printf("Risposta errata D:\n");
        Punteggio--;
    }
    else
    {
        printf("Risposta corretta :D\n");
        Punteggio++;
    }
    // FINE TERZA DOMANDA

    // QUARTA DOMANDA
    printf("Quarta domanda: \n");
    printf(Domanda4);
    printf("A) 9 giorni  ——  B) 47 ore —— C) 38 minuti  \n");
    printf("SCELTA: ");
    scanf(" %c", &Scelta);

    while (Scelta ≠ 'A' && Scelta ≠ 'B' && Scelta ≠ 'C')
    {
        printf("Scelta non valida. Riprova: ");
        scanf(" %c", &Scelta);
    }

    if (Scelta == 'A' || Scelta == 'B')
    {
        printf("Risposta errata D:\n");
        Punteggio--;
    }
    else
    {
        printf("Risposta corretta :D\n");
        Punteggio++;
    }
    // FINE QUARTA DOMANDA

    // QUINTA DOMANDA
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark M-] To Bracket M-Q Previous ^B Back ^◂ Prev Word ^A Home
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo M-6 Copy ^Q Where Was M-W Next ^F Forward ^▸ Next Word ^E End

CTRL (DESTRA)

Search ITA 11:03 03/12/2023

Kali Linux [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

1 2 3 4 11:04

kali@kali: ~/Desktop/Epicode_Lab

File Actions Edit View Help

GNU nano 7.2 MikeBongiorno.c

```
    // FINE QUARTA DOMANDA

    // QUINTA DOMANDA
    printf("Quinta domanda: \n");
    printf(Domanda5);
    printf("A) 3.000 anni fa  ——  B) 77.000 anni fa —— C) 13.000 anni fa  \n");
    printf("SCELTA: ");
    scanf(" %c", &Scelta);

    while (Scelta ≠ 'A' && Scelta ≠ 'B' && Scelta ≠ 'C')
    {
        printf("Scelta non valida. Riprova: ");
        scanf(" %c", &Scelta);
    }

    if (Scelta == 'A' || Scelta == 'C')
    {
        printf("Risposta errata D:\n");
        Punteggio--;
    }
    else
    {
        printf("Risposta corretta :D\n");
        Punteggio++;
    }
    // FINE QUINTA DOMANDA

    printf("Il tuo punteggio finale è: %d\n", Punteggio);
}

int main()
{
    char scelta;
    char NomeGiocatore[20];
    printf ("Benvenuto o benvenuta!\n");
    printf ("Questo quiz si chiama: Chi Vuol Essere Alessandro Barbero?\n");
    printf ("Ti aggiudicherai un punto per ogni risposta corretta.\n");
    printf ("Per ogni risposta sbagliata però, perderai un punto! Eh già, la storia è crudele.\n");
    do
    {
        scelta = Iniziale();

        if (scelta == 'A' || scelta == 'a')
        {
            NuovaPartita(NomeGiocatore);
            setDiDomande();
            // giocare di nuovo?
            printf("\n Vuoi giocare di nuovo? (S/N): ");
            scanf(" %c", &scelta);

            while (scelta ≠ 'S' && scelta ≠ 's' && scelta ≠ 'N' && scelta ≠ 'n')
            {
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark M-] To Bracket M-Q Previous ^B Back ^◂ Prev Word ^A Home
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo M-6 Copy ^Q Where Was M-W Next ^F Forward ^▸ Next Word ^E End

CTRL (DESTRA)

Search ITA 11:04 03/12/2023

Kali Linux [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

kali@kali: ~/Desktop/Epicode_Lab

File Actions Edit View Help

GNU nano 7.2 MikeBongiorno.c

```
    if (Scelta == 'A' || Scelta == 'C')
    {
        printf("Risposta errata D:\n");
        Punteggio--;
    }
    else
    {
        printf("Risposta corretta :D\n");
        Punteggio++;
    }
    // FINE QUINTA DOMANDA

    printf("Il tuo punteggio finale è: %d\n", Punteggio);
}

int main()
{
    char scelta;
    char NomeGiocatore[20];
    printf ("Benvenuto o benvenuta!\n");
    printf ("Questo quiz si chiama: Chi Vuol Essere Alessandro Barbero?\n");
    printf ("Ti aggiudicherai un punto per ogni risposta corretta.\n");
    printf ("Per ogni risposta sbagliata però, perderai un punto! Eh già, la storia è crudele.\n");
    do
    {
        scelta = Iniziale();

        if (scelta == 'A' || scelta == 'a')
        {
            NuovaPartita(NomeGiocatore);
            setDiDomande();
            // giocare di nuovo?
            printf("\n Vuoi giocare di nuovo? (S/N): ");
            scanf(" %c", &scelta);

            while (scelta != 'S' && scelta != 's' && scelta != 'N' && scelta != 'n')
            {
               printf("Scelta non valida. Riprova: ");
               scanf(" %c", &scelta);
            }
        }

        else
        {
           printf("Alla prossima!");
        }

    } while (scelta == 'S' || scelta == 's');

    return 0;
}
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark M-] To Bracket M-Q Previous ^B Back ^◂ Prev Word ^A Home
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo M-6 Copy ^Q Where Was M-W Next ^F Forward ^▸ Next Word ^E End

Al termine della partita, l'utente può scegliere se ricominciare tramite S o N. Finché la sua scelta non è una di quelle valide, non può andare avanti. Anche qui il metodo utilizzato è quello del **WHILE**.

CTRL (DESTRA)

Search

ITA 11:04 03/12/2023