Manuale per il docente:

Linee guida per l’attività “L’istruzione break”:

1. Introduzione:

L’istruzione "break" è un tipo di istruzione che comporta l'uscita forzata da un ciclo. È utile sotto determinati aspetti, ma un abuso di questa istruzione comporta, la maggior parte delle volte, la scrittura di codice ridondante e di difficile lettura. In particolare, è stato notato come essa sia utilizzata più volte all'interno di istruzioni di controllo "if", al verificarsi di una o più condizioni. La verifica di queste condizioni è spesso omessa all'interno della guardia dei cicli for/while, producendo di conseguenza una serie di if a cascata che, come già detto, rendono il codice ridondante. Questo modo di procedere potrebbe far pensare che gli studenti si trovino più a loro agio seguendo uno stile di programmazione di più basso livello, ispirandosi alle istruzioni di salto del linguaggio Assembly (istruzioni di salto condizionato in questo caso), per permettere l’uscita dal ciclo. Per rendere il codice più facile da comprendere e per sfruttare al meglio la maggiore espressività del linguaggio C, è buona norma sostituire le istruzioni di verifica all'interno degli if, aggiungendo le stesse istruzioni all'interno della guardia delle istruzioni for/while, collegandole con proposizioni logiche come “AND”(&&) e “OR”(||). Questa attività ha, quindi, l'obiettivo di porre particolare attenzione sull'istruzione break e sui suoi utilizzi.

1. Prerequisiti e Modelling:

A tal proposito è opportuno, oltre ad introdurre il concetto dell’istruzione break, un esercizio preliminare da svolgere in classe e/o in via telematica. Un buon punto di partenza può essere quello di svolgere un esercizio che può essere risolto in 2 modi distinti. Ad esempio, una buona esercitazione preliminare è la seguente:

Dato un array di interi, si vogliono trovare tutti gli elementi che sono divisibili per 2 e per 3. Si chiede, inoltre, di svolgere questo esercizio con e senza l’uso dell’istruzione break, facendo notare la differenza tra i 2 esempi di codice che verranno proposti. Una volta proposto questo tipo di esercitazione, è necessario sottolineare, come detto prima, come sia consigliabile utilizzare il meno possibile l’istruzione break all’interno dei cicli for/while, ed utilizzare, invece, le condizioni di uscita dal ciclo all’interno della guardia dei cicli for/while.

1. Concetti richiamati:

Nei vari esercizi proposti, seguendo quello che è l’approccio a spirale di Bruner, quindi, verrà riproposto l’uso delle proposizioni logiche e il loro uso all’interno della guardia dei cicli, oltre all’istruzione break. Si presuppone, infatti, che alcuni studenti non abbiano bene in mente la congiunzione logica nell’uso contemporaneo di due istruzioni come AND e OR.

1. Attività e Coaching:

Di seguito vengono proposte delle attività da proporre agli studenti:

Primo esempio: Dato un array di interi, verificare se questo contiene numeri primi e, nel caso ci siano, inserirli all'interno di un altro array e stamparne a video il contenuto. In questo caso viene utilizzato un ciclo annidato: Il ciclo più esterno serve per scorrere l'array, mentre quello più interno incrementa una variabile j, la quale funge da divisore per gli interi all'interno dell'array. L'iterazione più interna può interrompersi per due motivi:

1- Il divisore j è più grande dell'intero in posizione i-esima che si sta esaminando. In questo caso non c'è motivo di continuare ad utilizzare l'operatore modulo %: l'elemento in posizione i-esima è certamente un numero primo. Inserisco l'intero in posizione i-esima all'interno del mio array di numeri primi.

2- In posizione i-esima è presente un numero NON primo: il risultato della divisione modulo dà come risultato 0, quindi quell'elemento è divisibile per un opportuno valore di j. A questo punto ho una condizione per cui è inutile continuare a verificare il resto della divisione. Passo quindi all'elemento successivo.

Coaching: Questo esempio non dovrebbe presentare grosse difficoltà. Si raccomanda soltanto di prestare attenzione all’uso dell’istruzione break. Un errore può essere infatti quello di inserire questa istruzione all’interno di uno dei due cicli, ed uscire prematuramente da uno di questi prima che la scansione dell’array sia stata completata.

Secondo esempio: Dato un array di interi, verificare se è presente un elemento la cui somma delle singole cifre sia uguale a 10. Se tale elemento esiste, allora il programma termina. Viene quindi utilizzata una variabile "trovato" ed una variabile "somma". In particolare, se "somma" vale 10, allora "trovato" vale 1, e quindi il programma termina, in quanto si è trovato l'elemento desiderato. Nell'esempio proposto viene posta particolare attenzione sull'utilizzo dell'istruzione break e sul suo abuso: Come già detto, omettere istruzioni di guardia nei cicli for/while e verificarle in seguito con degli if a cascata, rende il codice ripetitivo e di difficile lettura. Si può fare di meglio...

Coaching: Anche in questo caso si consiglia di prestare massima attenzione all’utilizzo dell’istruzione break. Dal testo dell’esercizio, non dovrebbero esserci particolari difficoltà nel capire dove inserire questa istruzione.

Terzo esempio: Una rivisitazione del precedente, in questo caso il codice non prevede l'utilizzo dell'istruzione break. Infatti, le istruzioni che prima venivano verificate con degli if a cascata e che utilizzavano l'istruzione break, ora sono messe all'interno della guardia delle istruzioni for e while. Il codice risulta più compatto e semplice da leggere.

Coaching: Da quanto scritto, a differenza del secondo esempio, in questo caso non si può usare l’istruzione break, e quindi le condizioni di uscita dal ciclo sono messe all’interno della guardia dei cicli for e while. Si raccomanda, quindi, di scegliere con accuratezza l’utilizzo delle istruzioni AND (&&) ed OR (||). Un tipico errore può essere, infatti, quello di confondere i risultati di queste due operazioni dell’algebra booleana quando sono presenti all’interno della stessa guardia.