

Esame 20240904

Esercizio 1

(1) Esercizio 1 v1

ESSAY marked out of 10 penalty 0 File picker

Il formato DIMACS CNF è una rappresentazione testuale di una formula in forma normale congiuntiva. Una formula in forma normale congiuntiva è una congiunzione (logica "and") di un insieme di clausole. Ogni clausola è una disgiunzione (logica "or") di un insieme di letterali. Un letterale è una variabile o una negazione di una variabile. DIMACS CNF utilizza gli **interi positivi** per rappresentare le variabili e la loro negazione (valore negativo) per rappresentare la corrispondente variabile negata. Un file DIMACS cnf contiene una intestazione (prima riga) così formata: `p cnf <numero clausole> <numero variabili>`, seguita da una serie di clausole. Ogni clausola inizia con un numero positivo che specifica quanti letterali ci sono nella clausola. Ad esempio, la formula $(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3)$ può essere rappresentata nel formato DIMACS CNF come segue:

```
p cnf 2 3
3 1 2 -3
3 -1 -2 3
```

Si scriva un programma C++ che prende come argomento un nome di un file che deve essere interpretato come file DIMACS cnf. Il programma opera come segue:

- Apre in lettura il file specificato a linea di comando, e segnala errore in caso di errore di apertura.
- Legge l'intestazione e in caso di errore (i.e., manca o scritto diversamente `p`, manca o scritto diversamente `cnf`, numero di variabili o clausole non positivo) segnala errore e termina.
- Alloca un array di puntatori ad interi di dimensione pari al numero di clausole + 1 per memorizzare `nullptr` nella posizione `<numero clausole>` per segnalare la fine delle clausole.
- Legge le clausole e le memorizza nell'array di puntatori. Ogni clausola è memorizzata come un array di interi (da allocare secondo la dimensione della clausola specificata nel file) terminato da 0. Ad esempio, la clausola `3 -1 -2` viene memorizzata nell'array di interi di 4 elementi come segue: `[3, -1, -2, 0]`.
La lettura delle clausole deve terminare non appena si incontra un letterale che corrisponde ad una variabile con valore maggiore di `<numero variabili>`, o un letterale uguale a 0, o se il numero di letterali della clausola non corrisponde a quanto specificato nella clausola (primo valore), o se si incontra la fine del file prima di aver terminato la lettura delle `<numero clausole>` o del numero di letterali per clausola.
- Chiude il file aperto in lettura precedentemente.
- Se il file letto contiene più clausole di quelle specificate nell'intestazione (da verificare prima della chiusura), o se il file contiene clausole con più letterali di quelli specificati, o se il file contiene clausole con letterali che non corrispondono a variabili, il programma segnala errore (si veda esempio) e termina (deallocando tutta la memoria allocata).
- Chiama una funzione `print_clauses` che riceve come **unico** parametro un array di puntatori a intero terminato da `nullptr`, ovvero l'array precedentemente allocato e popolato (si assuma sia corretto), e stampa a video le clausole una per riga nell'ordine letto. La funzione prima di ritornare il controllo al chiamante stampa un suffisso che contiene `p cnf`

<numero clausole> <numero variabili>.

(Il numero di variabili si può ricavare come il max tra i letterali delle clausole, pertanto potrebbe essere inferiore a quanto specificato nel file letto precedentemente!).

- Dealloca tutta la memoria allocata precedentemente.

Il file `esercizio1.cpp` contiene gli include che definiscono le funzioni consentite ed è da completare con quanto richiesto sopra. Di seguito è riportato un esempio di esecuzione (**la formattazione dell'output e i messaggi di errore devono essere rispettati**).

```
computer > ./a.out buggy1.cnf
Error reading file "buggy1.cnf": eof or clause out of bound, or 0 encountered, or more clauses encount
computer > ./a.out buggy2.cnf
Error reading file "buggy2.cnf": eof or clause out of bound, or 0 encountered, or more clauses encount
computer > ./a.out buggy3.cnf
Error reading file "buggy3.cnf": eof or clause out of bound, or 0 encountered, or more clauses encount
computer > ./a.out buggy5.cnf
Error reading header of file "buggy5.cnf"
computer > ./a.out input.cnf
20 -10 5 0
-2 -3 0
-4 -8 9 30 0
22 -11 15 0
-21 -9 0
-14 -18 19 -30 0
-24 -28 29 30 0
12 -21 25 0
-1 -29 0
-24 -28 29 -10 0
p cnf 10 30
```

Note:

- Scaricare il file `esercizio1.cpp`, modificarlo per inserire il codice necessario per rispondere a questo esercizio. **Caricare il file sorgente risultato delle vostre modifiche a soluzione di questo esercizio** nello spazio apposito.
- All'interno di questo programma **non è ammesso** l'utilizzo di variabili globali o di tipo `static` e di funzioni di libreria al di fuori di quelle definite in `iostream`, `fstream`, `cstdlib`, `cstring` (per e.g. `strcmp`).
- Si sconsiglia di usare `atoi` o `atol` per convertire una stringa in un intero, invece si consiglia di leggere direttamente interi da file. È possibile usare `std::max` per calcolare il massimo tra due numeri.
- Si ricorda che, gli esempi di esecuzione sono puramente indicativi, e la soluzione proposta NON deve funzionare solo per l'input fornito, ma deve essere robusta a variazioni compatibili con la specifica riportata in questo testo.
- Si ricorda di inserire solo nuovo codice e di **NON MODIFICARE** il resto del programma (pena annullamento dell'esercizio).

`esercizio1.cpp`

`input.cnf`

`buggy1.cnf`

`buggy2.cnf`

`buggy3.cnf`

`buggy4.cnf`

Information for graders:

Total of marks: 10