

Una Coppia di numeri **interi** è formata da una parte positiva e da una parte negativa. Entrambe le parti possono assumere valori *potenzialmente molto grandi in valore assoluto*.

Un Insieme è invece una sequenza di d elementi di tipo Coppia.

Implementare le seguenti operazioni che possono essere compiute su una coppia o su un insieme di coppie:

PRIMA PARTE (*qualora siano presenti errori di compilazione, collegamento o esecuzione in questa parte, l'intera prova verrà considerata insufficiente e pertanto non verrà corretta*)

✓ Coppia $c(p, q)$;

Costruttore della classe Coppia che crea una coppia positivo-negativo, in cui la parte positiva è inizializzata a p e la parte negativa a $-q$ (p e q si suppongano *entrambi positivi*).

✓ Coppia c ;

Costruttore di default per la classe Coppia, che inizializza una coppia con parte positiva pari a 5 e parte negativa pari a -9.

✓ `cout<<c;`

Operatore di uscita per la coppia di numeri. L'uscita deve avere il seguente formato:

`<parte_positiva, parte_negativa>`

Esempio: nel caso di una coppia con parte positiva 6 e parte negativa -4 deve stampare:

`<6, -4>`

✓ Insieme $i(d)$;

Costruttore della classe Insieme, che inizializza un insieme composto da d coppie (ciascuna coppia dell'insieme è inizializzata al suo valore di default: `<5,-9>`).

✓ `cout<<i;`

Operatore di uscita per l'insieme di coppie. L'uscita deve avere il seguente formato:

`<partePos1, parteNeg1><partePos2, parteNeg2> (ecc...)`

Esempio: nel caso di un insieme di tre coppie con valori default, l'uscita deve essere:

`<5, -9><5, -9><5, -9>`

SECONDA PARTE

✓ `i = i2;`

Operatore di assegnamento, che sostituisce il valore dell'insieme risultato i col valore dell'insieme $i2$.

✓ `-i;`

Operatore di sottrazione unario, che scambia la parte positiva con la parte negativa di ogni coppia presente nell'insieme i e restituisce l'insieme così modificato.

Esempio: se i è composto dalle coppie `<5, -9><5, -9>`, allora `cout<<-i;` deve stampare:

`<9, -5><9, -5>`

✓ `~Insieme();`

Distruzione della classe Insieme.

Mediante il linguaggio C++, realizzare i tipi di dato astratti definiti dalle precedenti specifiche. Individuare eventuali situazioni di errore e metterne in opera un corretto trattamento.

NOTE SULLO SVOLGIMENTO DELLA PROVA PRATICA

AVVIO E IDENTIFICAZIONE

- Avviare la macchina in modalità diskless, scegliere “Fondamenti di Informatica I” ed effettuare il login: **nome:** studenti **password:** studenti
- Aprire un terminale e al prompt spostarsi sulla cartella ‘elaborato’ (`$ cd ~/elaborato`). Si utilizzi il comando `pwd` per verificare che ci si trovi nella cartella corretta `/home/studenti/elaborato`.
- Sempre al prompt dare il comando `ident`, sempre da dentro la cartella. Lo script richiede i propri dati (cognome, nome, numero di matricola e password (la password **non va dimenticata** in quanto è indispensabile per scaricare da internet il proprio elaborato a consegna avvenuta). Il comando `ident` crea il file `matricola.txt` nella cartella corrente. Lo script può essere lanciato più volte, in tal caso il file `matricola.txt` viene sovrascritto. Per verificare che il file sia stato creato e che il contenuto sia quello giusto dare il comando (la password è codificata):
`$ cat /home/studenti/elaborato/matricola.txt`
- A questo punto il docente verifica che tutti gli studenti abbiamo effettuato l’identificazione, dopodiché provvede a inviare i seguenti file nella cartella `elaborato` del proprio PC: `compito.h`, `compito.cpp`, `main.cpp`. Controllare pertanto che questi file, insieme al file `matricola.txt`, siano presenti sul proprio elaboratore.

SVOLGIMENTO DELLA PROVA

- Definire ed implementare il tipo di dato astratto richiesto e le relative funzioni nei file `compito.h` e `compito.cpp`. Il file `main.cpp` contiene la funzione principale `main()` ed è utilizzato dallo studente per testare la sua implementazione della classe. Il file `main.cpp` può essere modificato a piacere. In sede di valutazione dell’elaborato verrà considerato **esclusivamente il contenuto dei file `compito.h` e `compito.cpp`** ed è pertanto **vietato cambiare nome a tali file**.

Per compilare e linkare dare il comando:

```
$ g++ main.cpp compito.cpp (eseguibile invocabile tramite $ ./a.out)
(utilizzare g++ -g per includere le informazioni di debug qualora si intenda debuggare con ddd).
```

PER CONSEGNARE O RITIRARSI

Recarsi dal docente **dopo aver preso nota dell’identificativo della macchina** (esempi: g34, s23, c22, ...).

USCITA CHE DEVE PRODURRE IL PROGRAMMA

Test 1: costruttore e op. di uscita - classe Coppia (deve stampare <6,-4>
<6,-4>

Test 2: costruttore e op. di uscita - classe Insieme (deve stampare <5,-9><5,-9><5,-9>
<5,-9><5,-9><5,-9>

Test 3: operatore di assegnamento (deve stampare <5,-9><5,-9>
<5,-9><5,-9>

Test 4: del distruttore (non deve stampare nulla)

Test 5: operator - unario (deve stampare <9,-5><9,-5>
<9,-5><9,-5>