Un matematico chiede ad uno studente di ingegneria informatica di implementargli le operazioni tipiche dell'algebra tropicale, una particolare algebra definita sull'insieme dei numeri interi, in cui la somma ("tropicale") viene calcolata come il massimo tra i due interi, e il prodotto ("tropicale") come la somma convenzionale tra due interi. Implementare pertanto `TropInt`, un tipo di dato astratto basato su numeri interi, in cui le operazioni di somma e prodotto sono ridefinite come sopra. Implementare inoltre le seguenti operazioni descritte sotto.

--- PRIMA PARTE ---

✓ `TropInt t(ti);`
Costruttore che inizializza l'intero tropicale `t` con `ti`. NB: Un intero tropicale è identico agli interi convenzionali, solo le operazioni di somma e prodotto sono definite differentemente.

✓ `cout<<t;`
Operatore di uscita per la classe `TropInt`, che banalmente mostra a video il valore intero.

✓ `t1+t2;`
Operazione che restituisce un nuovo `TropInt` avente valore intero pari **al massimo** tra `t1` e `t2`. Si faccia in modo che questa operazione funzioni anche tra un `TropInt` ed un intero convenzionale: `3+t1` deve calcolare il massimo tra 3 e `t1`.

✓ `t1*t2;`
Operazione che restituisce un nuovo `TropInt` avente valore intero pari **alla somma** tra `t1` e `t2`.

✓ `t1.scomponiInPrimi();`
Operazione che mostra a video i fattori primi di cui è composto l'intero tropicale `t1`, nel seguente formato (i vari fattori primi debbono apparire in ordine crescente, eventualmente ripetuti, e separati da un singolo spazio bianco):

```
2 2 2 3 3 3 3 5
```

In particolare, la stampa si riferisce al caso in cui `t1` valga 3240. Infatti, la scomposizione in primi di 3240 è $\mathbf{2^3 \cdot 3^4 \cdot 5}$.
La stessa identica stampa deve essere fornita per interi negativi, senza specificare il segno negativo. L'intero tropicale -3240 avrà pertanto la stessa identica scomposizione in primi. Nel caso in cui l'intero tropicale valga 0, 1 oppure -1, la funzione non deve stampare nulla.

Suggerimento: un possibile algoritmo per stampare la scomposizione in fattori primi consiste innanzitutto nel definire un nuovo intero `n`, pari al valore assoluto dell'intero tropicale da fattorizzare. Successivamente, come prima cosa, si individuano gli eventuali fattori 2 presenti nel numero `n`. `n` va aggiornato ogni volta, dividendolo `n` per 2. Dopodiché si procede con la ricerca della presenza dei vari fattori dispari presenti nel numero, finché `n` non diventa pari ad 1.

--- SECONDA PARTE ---

Un `VectTropInt` è vettore di interi tropicali di lunghezza fissa, pari a quattro (ossia non possono esistere vettori di dimensione diversa da quattro). NB: Questa classe deve utilizzare la classe `TropInt`.

Su un vettore di interi tropicali sono definite le seguenti operazioni:

✓ `VectTropInt v(c);`
Costruttore che crea ed inizializza il vettore `v` di 4 interi tropicali, inizializzato attingendo dal vettore di 4 interi `c` passato come parametro.

✓ `v1*v2;`
Operazione di prodotto, che restituisce come risultato un singolo intero tropicale, calcolato usando la seguente formula:

$$\sum_{i=0}^{3} v1[i] * v2[i]$$

dove però le operazioni di somma e prodotto sono quelle dell'aritmetica tropicale (massimo e somma, rispettivamente).

Ad esempio, nel caso in cui `v1` contenga gli interi `{1,-2,4,-8}` e `v2` gli interi `{5,20,17,10}`, il risultato deve essere un intero tropicale pari a 21, in quanto:

`max(1+5,-2+20,4+17,-8+10) = max(6,18,21,2) = 21`

Si noti come, a livello formale, questo prodotto assomigli al prodotto scalare tra due vettori di interi. La differenza è solo nel tipo del risultato (`TropInt` invece che `int`), e nel valore numerico, in quanto in questo caso somma e prodotto sono definiti diversamente.

Infine, aggiungere il seguente metodo statico alla classe TropInt:

✓ `VectTropInt::maxPos()`

Che restituisca il massimo intero **strettamente positivo** utilizzato fino all'istante in cui viene invocata quella funzione per inizializzare un nuovo intero tropicale (il massimo positivo in assoluto, contando dunque anche i risultati delle operazioni di somma e prodotto. Nel caso in cui siano stati creati solo interi tropicali con interi negativi o nulli, la funzione deve restituire zero. Per semplicità è sufficiente fare in modo che `maxPos()` restituisca il valore del massimo numero intero streattamente positivo passato al costruttore fino a quel momento.

Mediante il linguaggio C++, realizzare il tipo di dato astratto definito dalle precedenti specifiche. Non è permesso utilizzare funzionalità della libreria STL come il tipo `string`, il tipo `vector`, il tipo `list`, ecc.

```cpp
// file main.cpp
#include <iostream>
#include "compito.h"
using namespace std;

int main() {
    cout<<"---PRIMA PARTE---";

    TropInt t1(54), t2(28);
    cout<<endl<<"Test del costruttore (deve stampare 54 e 28)"<<endl;
    cout<<t1<<" e "<<t2<<endl;

    cout<<"Testing op. + (deve stampare 54)"<<endl;
    cout<<t1+t2<<endl; // deve stampare 54, ossia il massimo dei due

    cout<<"Testing op. * (deve stampare 82)"<<endl;
    cout<<t1*t2<<endl; // deve stampare 82, ossia la somma, e non il prodotto!

    cout<<"Primo test della scomponi in numeri primi (deve stampare 2 3 3 3):"<<endl;
    t1.scomponiInPrimi(); cout<<endl;

    cout<<"Secondo test della scomponi in numeri primi (deve stampare 2 2 7):"<<endl;
    t2.scomponiInPrimi(); cout<<endl;


    cout<<endl<<"---SECONDA PARTE---"<<endl;

    const int c1[DIM] = {1,-2, 4,-8};
    VectTropInt v1(c1);
    const int c2[DIM] = {5,20,17,10};
    VectTropInt v2(c2);
    cout<<"Test del prodotto scalare (deve stampare 21)"<<endl;  // max(6,18,21,2) = 21
    cout<<v1*v2<<endl;

    cout<<"Test del metodo statico maxPos (deve stampare 82)"<<endl;
    cout<<TropInt::maxPos()<<endl;

    return 0;
}
```

USCITA ATTESA

```
--- PRIMA PARTE ---
Test del costruttore (deve stampare 54 e 28)
54 e 28
Testing op. + (deve stampare 54)
54
Testing op. * (deve stampare 82)
82
Primo test della scomponi in numeri primi (deve
stampare 2 3 3 3):
2 3 3 3
Secondo test della scomponi in numeri primi
(deve stampare 2 2 7):
2 2 7
```

```
--- SECONDA PARTE ---
Test del prodotto scalare (deve stampare 21)
21
Test del metodo statico maxPos (deve stampare 82)
82
```