`ParteFraz` è una classe che deve essere in grado di memorizzare la parte frazionaria di un numero reale, in virgola fissa, su 16 cifre dopo la virgola.

Implementare le seguenti operazioni che possono essere effettuate su una `ParteFraz`.

(E' VIETATO USARE UN VETTORE DI BOOLEANI O DI INTERI)

--- PRIMA PARTE ---

✓ `ParteFraz p(pf);`
Costruttore che inizializza l'oggetto `p` di tipo `ParteFraz`, utilizzando l'argomento `pf` di tipo reale in doppia precisione. `pf` deve essere non negativo e minore stretto di uno. `pf` può valere anche zero.

✓ `cout<<p;`
Operatore di uscita per la classe `ParteFraz`, che deve mostrare a video l'oggetto `p` nel seguente formato:
`1010000000000000b`
qualora la parte frazionaria fosse stata pari a 0.625.

✓ `p.visBase16();`
Operatore che mostra il contenuto della `ParteFraz p` in base 16. Per esempio, nel caso in cui `p` contenga la parte frazionaria 0.625, la funzione dovrà mostrare a video la seguente stringa:
`0x6000`

✓ `double(p);`
Operazione che converte una parte frazionaria in un valore double. Per esempio, nel caso in cui la parte frazionaria `p` sia stata generata con il valore double 0.25, dovrà restituire 0.25, ma come tipo double, non `ParteFraz`, ovviamente. In altre parole si tratta di implementare l'operazione di conversione inversa.

--- SECONDA PARTE ---

Una `VirgolaFissa` è una classe che deve memorizzare sia la parte intera che la parte frazionaria di un numero reale **non negativo**, assumendo di rappresentarlo in virgola fissa e di riservare 16 bit per la parte intera e 16 per quella frazionaria. La classe `VirgolaFissa` **deve** usare la classe `ParteFraz`.

Implementare le seguenti operazioni che possono essere effettuate su una `VirgolaFissa`.

✓ `VirgolaFissa v(pi,pf);`
Costruttore per la classe `VirgolaFissa`, che prende in ingresso il numero naturale `pi` (assunto non negativo) per la parte intera ed un numero reale non negativo e minore di uno `pf` per la parte frazionaria.

✓ `cout<<v;`
Operatore di uscita per la classe `VirgolaFissa`, che deve mostrare a video l'oggetto `v` nel seguente formato:
`(3,0.25)`
qualora la parte intera sia pari a 3 e la parte frazionaria pari a 0.25.

✓ `v.raddoppia();`

Funzione che raddoppia il valore di `v`. Se `v` prima valeva (3,0.25), dopo dovrà valere (6,0.5).

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.

E' invece consentito l'uso della libreria cmath.

Richiami teorici	*Esempio di applicazione dell'algoritmo al caso di parte frazionaria pari a 0.3*
f0 = F(r) *Se f0 ≠ 0 eseguire la seguente procedura iterativa:* *f-1 = F(f0 * 2) a-1 = I(f0 * 2)* *f-2 = F(f-1 * 2) a-2 = I(f-1 * 2)* *…* *fino a che f-j è uguale a zero oppure si è raggiunta la precisione desiderata.*	**Esempio per pf = 0.3** **F** | **I** f_{-1}=F(0.3*2=**0.6**)=0.6 | a_{-1}=I(**0.6**)=**0** f_{-2}=F(0.6*2=**1.2**)=0.2 | a_{-2}=I(1.2)=**1** f_{-3}=F(0.2*2=**0.4**)=0.4 | a_{-3}=I(0.4)=**0** f_{-4}=F(0.4*2=**0.8**)=0.8 | a_{-4}=I(0.8)=**0** f_{-5}=F(0.8*2=**1.6**)=0.6 | a_{-5}=I(1.6)=**1** f_{-6}=F(0.6*2=**1.2**)=0.2 | a_{-6}=I(1.2)=**1**

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

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

    cout<<endl<<"Test del costruttore di PartFraz (deve stampare '0100000000000000000b')"<<endl;
    ParteFraz p(0.25);
    cout<<p<<endl;

    cout<<endl<<"Test del costruttore di PartFraz (deve stampare '1010000000000000000b')"<<endl;
    ParteFraz p2(0.625);
    cout<<p2<<endl;

    cout<<endl<<"Test della visBase16() della classe ParteFraz (Deve stampare '6000')"<<endl;
    ParteFraz p3(0.375);
    p3.visBase16();

    cout<<endl<<"Test dell'operatore di conversione a double di una ParteFraz (Deve stampare '0.375')"<<endl;
    cout<<double(p3)<<endl;


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

    cout<<endl<<"Test del costruttore di VirgolaFissa (deve stampare '(3,0.375)')"<<endl;
    VirgolaFissa v(3u, 0.375);
    cout<<v<<endl;

    cout<<endl<<"Test della funzione raddoppia di VirgolaFissa (deve stampare '(6,0.75)')"<<endl;
    v.raddoppia();
    cout<<v<<endl;

    cout<<endl<<"Altro test della funzione raddoppia di VirgolaFissa (deve stampare '(13,0.5)')"<<endl;
    v.raddoppia();
    cout<<v<<endl;
}
```

USCITA ATTESA

```
--- PRIMA PARTE ---

Test del costruttore di PartFraz (deve stampare
'0100000000000000000b')
0100000000000000b


Test del costruttore di PartFraz (deve stampare
'1010000000000000000b')
1010000000000000b


Test della visBase16() della classe ParteFraz
(Deve stampare '0x6000')
0x6000

Test dell'operatore di conversione a double di
una ParteFraz (Deve stampare '0.375')
0.375
```

```
--- SECONDA PARTE ---

Test del costruttore di VirgolaFissa (deve
stampare '(3,0.375)')
(3,0.375)


Test della funzione raddoppia di VirgolaFissa
(deve stampare '(6,0.75)')
(6,0.75)


Altro test della funzione raddoppia di
VirgolaFissa (deve stampare '(13,0.5)')
(13,0.5)
```