

```
1: class ProntoSoccorso {
2: public:
3:     ProntoSoccorso(){pazienti=new Paziente*[30]; pazientiInseriti=0;}
4:     ~ProntoSoccorso();
5:     void aggiungiPaziente(Paziente* p);
6:     void rimuoviProssimoPaziente();
7:     Paziente* prossimoPaziente() const;
8:     unsigned int numPazientiInAttesa() const;
9:     friend ostream& operator<<(ostream& out, const ProntoSoccorso& p);
10: private:
11:     Paziente** pazienti;
12:     unsigned int pazientiInseriti; //I pazienti inseriti
13: };

//Completare la definizione della classe Paziente (max 2 punti)
class Paziente {
    public:

    private:
        unsigned int prioritata;
};

//Implementare i seguenti metodi (max 5 punti)
void ProntoSoccorso::aggiungiPaziente(Paziente* p)
{ //Aggiunge un paziente (se non sfora la capienza) in ordine di priorità

}

Paziente* ProntoSoccorso::prossimoPaziente() const {

}

//Completare opportunamente il main (max 2 punti)
int main() {
    ProntoSoccorso* ps = new ProntoSoccorso();
    ProntoSoccorso* ps2 = new ProntoSoccorso();
    Paziente* p1 = new Paziente("nome1", "cognome1", "indirizzo1", 0); //Priorità 0
    Paziente* p2 = new Paziente("nome2", "cognome2", "indirizzo2", 1); //Priorità 1
    ps->aggiungiPaziente(p1); ps->aggiungiPaziente(p2);
    ps2->aggiungiPaziente(p1);

    return 0;
}
```

Programmazione Ad Oggetti. 22 Settembre 2016

Cognome: _____ Nome: _____

Matricola: _____ Esercizi lab: _____

Rispondere alle seguenti domande a risposta multipla (3 punti per risposta esatta, -2 punti per risposta sbagliata):

1. Quale tra le seguenti implementazioni è corretta?

- a) `~ProntoSoccorso() {}`
- b)

```
~ProntoSoccorso() {
    for(int i=0; i < pazientiInseriti; i++)
        delete pazienti[i];
}
```
- c)

```
~ProntoSoccorso() {
    for(int i=0; i < pazientiInseriti; i++)
        delete pazienti[i];
    delete pazienti;
}
```
- d) Nessuna delle precedenti

2. Quale tra le seguenti implementazioni è corretta?

- a) `void rimuoviProssimoPaziente(){ pazienti.remove(prossimoPaziente()); }`
- b) `void rimuoviProssimoPaziente(){ pazienti.pop_back(); }`
- c)

```
void rimuoviProssimoPaziente(){
    int j = 0; int i = 0;
    Paziente* p = prossimoPaziente();
    for(; i < pazientiInseriti; i++) {
        (*this)[j]=(*this)[i];
        if(*at(i)!=*p)
            j++;
        else
            delete (*this)[j];
    }
    resize(j);
    return (i!=j);
}
```
- d) Nessuna delle precedenti

3. Quale metodo tra il costruttore di copia e l'operatore di assegnamento dovrebbe essere inserito nella classe ProntoSoccorso?

- a) Il costruttore di copia
- b) L'operatore di assegnamento
- c) Sono entrambi necessari
- d) Sono entrambi inutili

4. Quale tra le seguenti implementazioni è corretta?

- a) `unsigned int numPazientiInAttesa() const { return pazienti.pazientiInseriti; }`

- b) `unsigned int numPazientiInAttesa() const { return pazientiInseriti; }`
- c) `unsigned int numPazientiInAttesa() const { return pazienti.size(); }`
- d) `unsigned int numPazientiInAttesa() const { return pazienti.capacity(); }`
5. Quale tra le seguenti implementazioni è corretta?
- a) `friend ostream& operator<<(ostream& out, const ProntoSoccorso& p) {
for(int i = 0; i < p.pazientiInseriti; i++)
out << *(p.pazienti[i]) << endl;
return out;
}`
- b) `friend ostream& operator<<(ostream& out, const ProntoSoccorso& p) {
for(int i = 0; i < p.size(); i++)
out << p.pazienti[i] << endl;
return out;
}`
- c) `friend ostream& operator<<(ostream& out, const ProntoSoccorso& p) {
for(int i = 0; i < p.size(); i++)
out << *(p.pazienti[i]) << endl;
return out;
}`
- d) `friend ostream& operator<<(ostream& out, const ProntoSoccorso& p) {
for(int i = 0; i < p.pazientiInseriti; i++)
out << p.pazienti[i] << endl;
return out;
}`
6. Sia `class ProntoSoccorsoSpeciale : private ProntoSoccorso` la definizione di una classe `ProntoSoccorsoSpeciale`. Quale tra le seguenti istruzioni non è consentita nel costruttore di `ProntoSoccorsoSpeciale`?
- a) `cout << pazientiInseriti << endl;`
- b) `cout << numPazientiInAttesa() << endl;`
- c) Sono entrambe consentite
- d) Sono entrambe sbagliate
7. Quale tra le seguenti istruzioni sarebbe scorretta nella funzione `main`?
- a) `ProntoSoccorsoSpeciale* p = new ProntoSoccorso();
cout << p->numPazientiInAttesa() << endl;`
- b) `ProntoSoccorsoSpeciale* p = new ProntoSoccorso();
cout << p->pazientiInseriti << endl;`
- c) Sono entrambe consentite
- d) Sono entrambe sbagliate