

[illegible]

```

        else {
            manager.setDatiMan();
            man = true;
        }
        break;
    case '2' : if (!man)
        {
            cout << "\n\n\tAggiungere prima il manager!!\n";
            getch();
        }
        else aggiungi(manager);
        break;
    case '3' : if (!man)
        cout << "\n\n\tAggiungere prima il manager!!\n";
        else
        {
            system("cls");
            manager.outMan();
        }
        getch();
        break;
    }
}while(scelta != 'e');
return 0;
} //fine main

```

```

char menu()
{
    char scelta;
    system("cls");
    cout << "\n\tmenu' \n\n";
    cout << "\n\t1 - aggiungi manager";
    cout << "\n\t2 - aggiungi impiegato al gruppo";
    cout << "\n\t3 - visualizza manager";
    cout << "\n\t4 - esci\n";
    scelta = getch();
    return scelta;
} //fine menu

```

```

void aggiungi(Manager & manager)
{
    Impiegato nuovo;
    nuovo.setDatiImp();
    manager.addToGrp(nuovo);
}

```

```

///////////////////////////////////////////////////7
//impiegato
Impiegato::Impiegato()
{ //inizializzazione delle proprietà dell'impiegato
    this->nome = "";
    this->cognome = "";
    this->dipartimento = "";
    this->eta = 0;
    this->stipendio = 0;
}

```

```

void Impiegato::setDatiImp()
{ //inserimento dei dati
    cout << "\n\tInserimento dati personali : \n\n";
    this->nome = inserisciStringa("\tnome : ");
    this->cognome = inserisciStringa("\tcognome : ");
    this->eta = inserisciIntero("\teta' : ");
}

```

```

    this->dipartimento = inserisciStringa("\tdipartimento : ");
    this->stipendio = inserisciDouble("\tstipendio : ");
}

void Impiegato::outImp()
{ //visualizzazione dei dati
    cout << "\n\tnome : " << this->nome;
    cout << "\tcognome : " << this->cognome;
    cout << "\n\teta' : " << this->eta;
    cout << "\tdipartimento : " << this->dipartimento;
    cout << "\n\tstipendio : " << this->stipendio;
}

string Impiegato::inserisciStringa(string messaggio)
{ //controlla i caratteri inseriti e ritorna una stringa
    string temp;
    do{
        cout << messaggio;
        cin >> temp;
        if (temp.size() > MaxString) {cout << "\tsono consentiti massimo " << MaxString << "
caratteri. Riprova!!\n";}
        }while(temp.size() > MaxString);
    return temp;
}

int Impiegato::inserisciIntero(string messaggio)
{ //controlla i caratteri inseriti e ritorna un intero
    string temp;
    bool corretto = false;
    do{
        corretto = true;
        cout << messaggio;
        cin >> temp;
        for (int i = 0 ; i < temp.size() ; i++)
            if (!isdigit(temp.at(i)))
                corretto = false;
        if (!corretto) cout << "\tci sono caratteri non consentiti. Ripeti!!\n";
    }while(!corretto);
    return atoi(temp.c_str()); //conversione in intero
}

double Impiegato::inserisciDouble(string messaggio)
{ //controlla i caratteri inseriti e ritorna un double
    string temp;
    bool corretto = false;
    do{
        corretto = true;
        cout << messaggio;
        cin >> temp;
        for (int i = 0 ; i < temp.size() ; i++)
            if (!isdigit(temp.at(i)) && temp.at(i) != '.')
                corretto = false;
        if (!corretto) cout << "\tci sono caratteri non consentiti. Ripeti!!\n";
    }while(!corretto);
    return atof(temp.c_str()); //atoi(temp.c_str())
}

//////////manager

Manager::Manager()
{ //inizializzazione del livello e del numero degli impiegati già inseriti
    this->livello = "";
    this->numImp = 0;
}

void Manager::setDatiMan()
{
    setDatiImp(); //inserimento dei dati del manager
    this->livello = inserisciStringa("\tlivello : ");
}

```

```

void Manager::outMan()
{ //visualizzazione dei dati del manager e dei dipendenti del gruppo
  cout << "\n\tDati del manager : \n";
  outImp();
  cout << "\tlivello : " << this->livello << endl;
  cout << "\tdipendenti aggiunti al gruppo : " << this->numImp << endl;
  if (numImp > 0)
  {
    cout << "\n\tDipendenti del gruppo : ";
    for (int i = 0 ; i < numImp ; i++)
    {
      cout << "\n\n\tdipendente " << (i+1) << endl ;
      gruppo[i]->outImp();
    }
  }
}

void Manager::addToGrp(Impiegato & temp)
{ //aggiunta di un impiegato al gruppo
  if (numImp < MaxGruppo)
  {
    gruppo[numImp] = new Impiegato;
    (*gruppo[numImp]) = temp;
    this->numImp++;
  }
  else cout << "\n\tnumero di dipendenti massimi raggiunto!!\n";
}

```