

```

#include <iostream>

using namespace std;
int const NMAX=30;
float const acc=9.81;

int leggiInt(int vmin,int vmax, string msg);
string leggiStr(string msg);
char leggiChar(string msg);
int leggiFloat(int vmin,int vmax, string msg);
void insOgg(int n, string nome[],float massa[],string unity[],int i);
void visOgg(int n,string nome[], float massa[], string unity[]);
string misura( string msg);
void pesoMinore(float massa[],string unity[],int n,int i);
void mediaPeso(float massa[],int n);
void ricerca(string name,string nome[],int n);
float calcoloPeso(float massa[],string unity[]);
/*
n          numero degli oggetti          int
nome       nome oggetto                  vettore di NMAX stringhe
massa      massa oggetto                 vettore di NMAX reali
unity      unita della massa              vettore di NMAX stringhe
i          numero indice                  int
m          lettera menu                   carattere

INIZIO
a
insOgg(n,nome[],massa[],unity[],i);
i=i+1;
b
visOgg(n,nome[],massa[],unity[]);
c
pesominore(n, massa[],nome[],i);
d
mediaPeso(n, massa[],nome[],i);
e
ricerca(n, massa[],nome[],i);
f

FINE

*/
int main(){
    int n;
    string nome[NMAX];
    float massa[NMAX];
    string unity[NMAX];
    string name;
    int i=0;
    char m;
    do{
        cout<<"a. inserimento di un oggetto con la relativa massa"<<endl<<"b. la visualizzazione degli oggetti
inseriti con la relativa massa espressa in kg ;"<<endl<<"c. elenco dell'i%oggetto o degli oggetti con peso minore,
sapendo che il peso P di un oggetto i% dato dal prodotto della massa m* accelerazione di gravita'i% g (9,8 m/s2)
P=m*g dove m deve essere espressa in kg; (il calcolo del peso DEVE essere realizzato tramite una funzione)"<<
endl<<"d. stampa della media del peso di tutti gli oggetti; (la somma dei Pesi deve essere realizzata tramite
una funzione che usa la stessa funzione del punto"<<endl<<"e. dato un intervallo fornito in input di peso
(P1,P2) stabilire quanti oggetti rientrano in questo intervallo"<<endl<<"f. elenco degli oggetti in ordine
crescente di peso (usare la stessa funzione del punto "<<endl<<"g. termina";
        cin>>m;
        switch(m){
            case 'a':
                insOgg(n,nome,massa,unity,i);
                i=i+1;
                break;
            case 'b':

```

```

        visOgg(n,nome, massa, unity);
        break;
    case 'c':
        pesoMinore(massa, unity, n, i);
        break;
    case 'd':
        mediaPeso(massa, n);
        break;
    case 'e':
        ricerca(name, nome, n);
        break;
    case 'f':
        break;
    }
}while(m!='g');
}
/*
n        numero        intero

funzione leggiInt
    scrivi msg
    leggi n e controlla che sia >vmin && <vmax
FINE
*/
int leggiInt(int vmin, int vmax, string msg){
    int n;
    do{
        cout<<msg<<endl;
        cin>>n;
    }while(n<vmin || n>vmax);
    return n;
}

/*
s        parola        stringa

funzione leggiStr
    scrivi msg
    leggi s e controlla che sia !=""
FINE
*/
string leggiStr(string msg){
    string s;
    do{
        cout<<msg<<endl;
        cin>>s;
    }while(s=="");
    return s;
}

/*
c        parola        stringa

funzione leggiStr
    scrivi msg
    leggi s e controlla che sia !='\n'
FINE
*/

char leggiChar(string msg){
    char c;
    do{
        cout<<msg<<endl;
        cin>>c;
    }

```

```

    }while(c=='\n');
    return c;
}

/*
n        numero        reale

funzione leggiFloat
    scrivi msg
    leggi n e controlla che sia >vmin && <vmax
FINE
*/

int leggiFloat(int vmin,int vmax, string msg){
    float n;
    do{
        cout<<msg<<endl;
        cin>>n;
    }while(n<vmin || n>vmax);
    return n;
}

/*
procedura insOgg
    n=leggiInt(1,NMAX,"quanti oggetti devi registrare")
    i=0
    mentre(i<n)
        nome[i]=leggiStr("nome oggetto")
        unity[i]=misura("unita di misura(kg,mg,g)")
        massa[i]=leggiFloat(0,10000,"massa oggetto")
        i=0
    fciclo
FINE
*/

void insOgg(int n, string nome[],float massa[],string unity[],int i){
    n=leggiInt(1,NMAX,"quanti oggetti devi registrare");
    for(i=0;i<n;i++){
        nome[i]=leggiStr("nome oggetto");
        unity[i]=misura("unita di misura(kg,mg,g)");
        massa[i]=leggiFloat(0,10000,"massa oggetto");
    }
}

/*
i        indice vettore        intero

procedura visOgg
    i=0
    mentre(i<n)
        se(unity[i]=="g")
            massa[i]=massa[i]/1000
            unity[i]="kg"
        fse
        se(unity[i]=="mg")
            massa[i]=massa[i]/1000000
            unity[i]="kg"
        fse
        scrivi nome[i]
        scrivi massa[i] & unity[i]
    fciclo
FINE
*/

void visOgg(int n,string nome[], float massa[],string unity[]){
    for(int i=0;i<n;i++){

```

```

        if(unity[i]=="g"){
            massa[i]=massa[i]/1000;
            unity[i]="kg";
        }
        if(unity[i]=="mg"){
            massa[i]=massa[i]/1000000;
            unity[i]="kg";
        }
        cout<<"nome="<<nome[i]<<endl;
        cout<<"massa="<<massa[i]<<unity[i]<<endl;
        cout<<endl;
    }
}

/*
s        unita di misura della massa        stringa

funzione misura
    ripeti
        scrivi msg
        leggi s
    finche(s=="kg" || s=="mg" || s=="g" )
FINE
*/

string misura(string msg){
    string s;
    do{
        cout<<msg<<endl;
        cin>>s;
    }while(s!="kg" && s!="mg" && s!="g");
    return s;
}

/*

i=0

*/

void pesominore(int n,float massa[],string nome[],string unity[],int i){
    i=0;
    float peso;
    float pesomin=calcoloPeso(massa,unity);
    for(i=1;i<n;i++){
        pesomin=calcoloPeso(massa,unity);
    }
    bool trovato=false;
    for(i=0;i<n;i++){
        pesomin=ricerca
    }
    if(!trovato)
        cout<<"non trovato"<<endl;
}

float calcoloPeso(int massa,string unity){
    float peso;
    peso[i]=massa[i]*acc;
    return peso;
}

void mediaPeso(int n,float massa[],string nome[],int i){
    float somma,media;
    for(i=0;i<n;i++){
        somma=int calcoloPeso(int n,float massa[],string nome[],int i);

```

```

    }
    media=somma/n;
    cout<<"la media dei pesi e "<<media<<endl;
}
void ricerca(int n,float massa[],string nome[],int i){
    int a,b;
    float peso;
    bool trovato=false;
    a=leggiInt(0,1000000,"dammi margine a dell'intervallo");
    b=leggiInt(0,1000000,"dammi margine b dell'intervallo");
    for(i=0;i<n;i++){
        if(calcoloPeso()>=a && calcoloPeso<=b)
            cout<<nome[i]<<" appartiene all'intervallo"<<endl;
            trovato=true;
    }
    if(!trovato)
        cout<<"nessun peso appartiene all'intervallo"<<endl;
}

```