**题目一**

**代码**

#pragma GCC optimize("Ofast", "inline", "-ffast-math")

#pragma GCC target("avx,sse2,sse3,sse4,mmx")

#include<bits/stdc++.h>

#define inf 0x3f3f3f3f

//#define int long long

using namespace std;

const int N=2e5+7;

const int mod=1e9+7;

//int read(){ int x=0,f=1;char ch=getchar();while(ch<'0'||ch>'9'){if(ch=='-') f=f\*-1;ch=getchar();}while(ch>='0'&&ch<='9'){x=x\*10+ch-'0';ch=getchar();}return x\*f;}

struct item{

int id;

string name,clas;

int num,price;

}it[N];

int cnt=0;

map<string,int>mp; //商品对应的编号

vector<string>ryp; //日用品

map<string,int>num\_of\_cls;

map<string,int>sum\_of\_cls;

map<string,int>cls;

bool cmp1(item X,item Y){

return X.id<Y.id;

}

bool cmp2(item X,item Y){

return X.num>Y.num;

}

signed main(){

// ios::sync\_with\_stdio(0);

// cin.tie(0);cout.tie(0);

// freopen("in.cpp","r",stdin);

// freopen("out.cpp","w",stdout);

int ID,NUM,PRI;

string NAME,CLS,opt;

cout<<"仓库管理系统 By:2020101603 rwl\n";

cout<<"输入指令：end-结束;modify-新增或修改商品信息;del-删除商品;pr\_ryp-打印日用品信息\n";

cout<<"pr\_sort-按库存从多到小输出商品;cal-计算每个类别商品总数和平均单价，以及总个数\n";

while(cin>>opt){

if(opt=="end") break;

if(opt=="modify"){

cout<<"\n输入新增或修改货品的编号，名称，类别，数量和单价\n";

cin>>ID>>NAME>>CLS>>NUM>>PRI;

if(mp[NAME]){

int to=mp[NAME];

sum\_of\_cls[CLS]-=it[to].num\*it[to].price;

num\_of\_cls[CLS]-=it[to].num;

it[to].id=ID;

it[to].num+=NUM;

it[to].clas=CLS;

it[to].price=PRI;

num\_of\_cls[CLS]+=it[to].num;

sum\_of\_cls[CLS]+=it[to].num\*it[to].price;

cout<<"成功修改货物信息!\n\n";

}else{

cls[CLS]++;//这个类别的商品+1

it[++cnt].id=ID;

it[cnt].name=NAME;

it[cnt].clas=CLS;

it[cnt].num=NUM;

it[cnt].price=PRI;

num\_of\_cls[CLS]+=NUM;

sum\_of\_cls[CLS]+=NUM\*PRI;

mp[NAME]=cnt;

if(CLS=="日用品")

ryp.push\_back(NAME);

cout<<"新增货物成功！\n\n";

}

}

if(opt=="del"){

cout<<"\n输入要删除的货品的名称\n";

cin>>NAME;

cls[CLS]++;

if(mp[NAME]){

cls[it[mp[NAME]].clas]--;

for(vector<string>::iterator iter=ryp.begin();iter!=ryp.end();iter++){

if(\*iter==NAME) ryp.erase(iter);

}

mp[NAME]=0;

cout<<"删除成功！\n";

}else{

cout<<"货品不存在！！！\n";

}

}

if(opt=="pr\_ryp"){

cout<<"打印日用品清单\n";

for(int i=0;i<ryp.size();i++){

int to=mp[ryp[i]];

if(!to) continue;

cout<<it[to].id<<" "<<it[to].name<<" "<<it[to].clas<<" "<<it[to].num<<" "<<it[to].price<<"\n";

}

}

if(opt=="cal"){

cout<<"\n计算每个类别商品总数和平均单价，以及总个数\n";

int res=0;

for(map<string,int>::iterator iter=cls.begin();iter!=cls.end();iter++){

string cls=(iter->first);

if(num\_of\_cls[cls]==0) continue;

cout<<cls<<",总数:"<<num\_of\_cls[cls]<<",平均单价:"<<(double)sum\_of\_cls[cls]/(double)num\_of\_cls[cls]<<"\n";

res+=num\_of\_cls[cls];

}

cout<<"商品总个数"<<res<<"\n";

cout<<"计算结束\n";

}

if(opt=="pr\_sort"){

sort(it+1,it+1+cnt,cmp2);

cout<<"\n按库存输出\n";

for(int i=1;i<=cnt;i++){

cout<<it[i].id<<" "<<it[i].name<<" "<<it[i].clas<<" "<<it[i].num<<" "<<it[i].price<<"\n";

}

sort(it+1,it+1+cnt,cmp1);

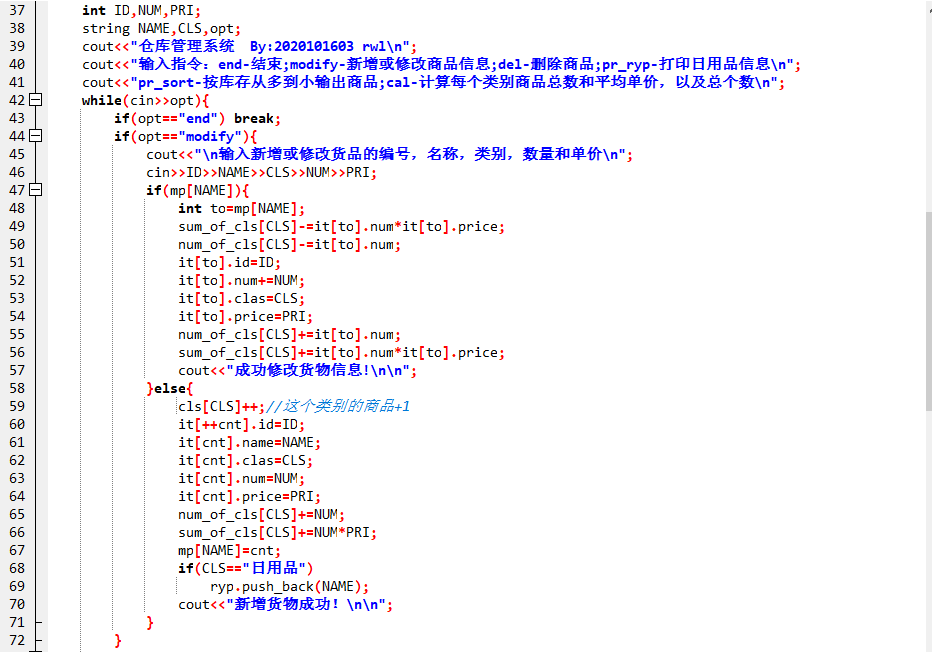
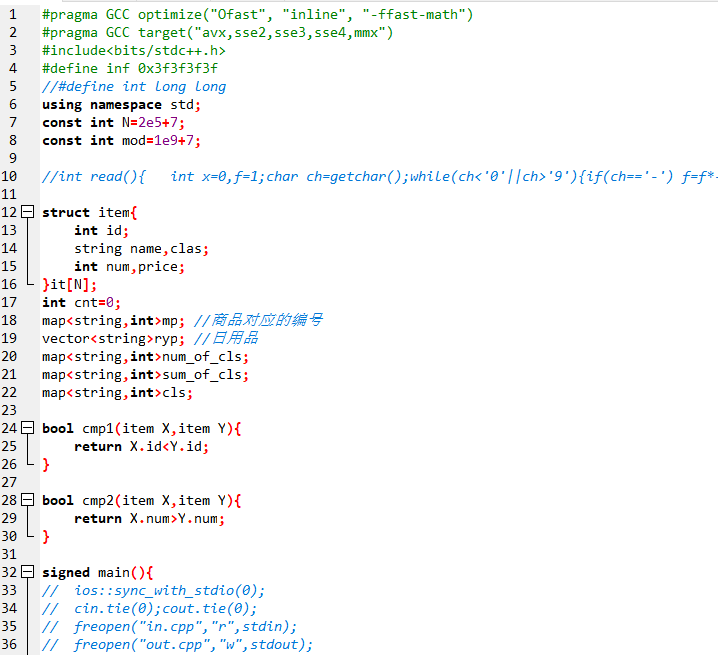
}

}

cout<<"程序结束！！\n";

return 0;

}

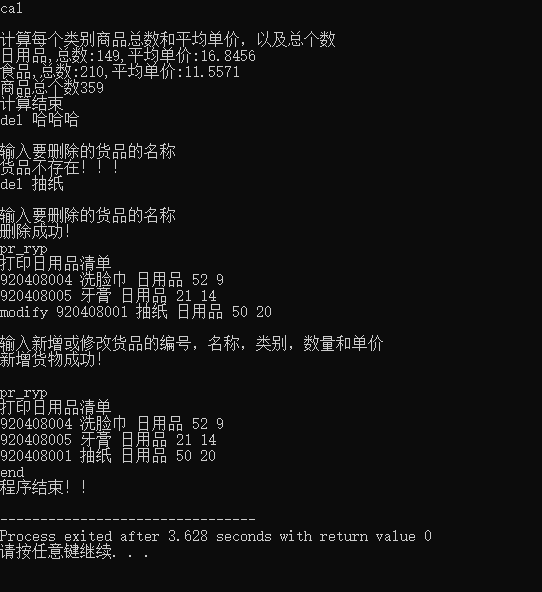


**输入数据**



**输出结果**





**题目二**

**代码**

#pragma GCC optimize("Ofast", "inline", "-ffast-math")

#pragma GCC target("avx,sse2,sse3,sse4,mmx")

#include<bits/stdc++.h>

#define inf 0x3f3f3f3f

//#define int long long

using namespace std;

const int N=2e5+7;

const int mod=1e9+7;

//int read(){ int x=0,f=1;char ch=getchar();while(ch<'0'||ch>'9'){if(ch=='-') f=f\*-1;ch=getchar();}while(ch>='0'&&ch<='9'){x=x\*10+ch-'0';ch=getchar();}return x\*f;}

int n,m,u,v,w,fa[N],ans=0;

int find(int x){

if(fa[x]==x) return x;

return fa[x]=find(fa[x]);

}

void unite(int x,int y){

fa[find(x)]=find(y);

}

struct E{

int from,to,dis,nxt;

}e[N];

int head[N],cnt=0;

void addedge(int u,int v,int w){

e[++cnt]=(E){u,v,w,head[u]};

head[u]=cnt;

}

bool cmp(E a,E b){

return a.dis<b.dis;

}

signed main(){

// ios::sync\_with\_stdio(0);

// cin.tie(0);cout.tie(0);

// freopen("in.cpp","r",stdin);

// freopen("out.cpp","w",stdout);

cout<<"铺设方案(MST) By2020101603 rwl\n";

cout<<"输入点数和边数\n";

cin>>n>>m;

for(int i=1;i<=n;i++) fa[i]=i;

cout<<"输出双向边的起点、终点和权值\n";

for(int i=1;i<=m;i++){

cin>>u>>v>>w;

addedge(u,v,w);

addedge(v,u,w);

}

sort(e+1,e+1+cnt,cmp);

for(int i=1;i<=cnt;i++){

u=find(e[i].from),v=find(e[i].to);

if(u==v) continue;

fa[u]=v;

ans+=e[i].dis;

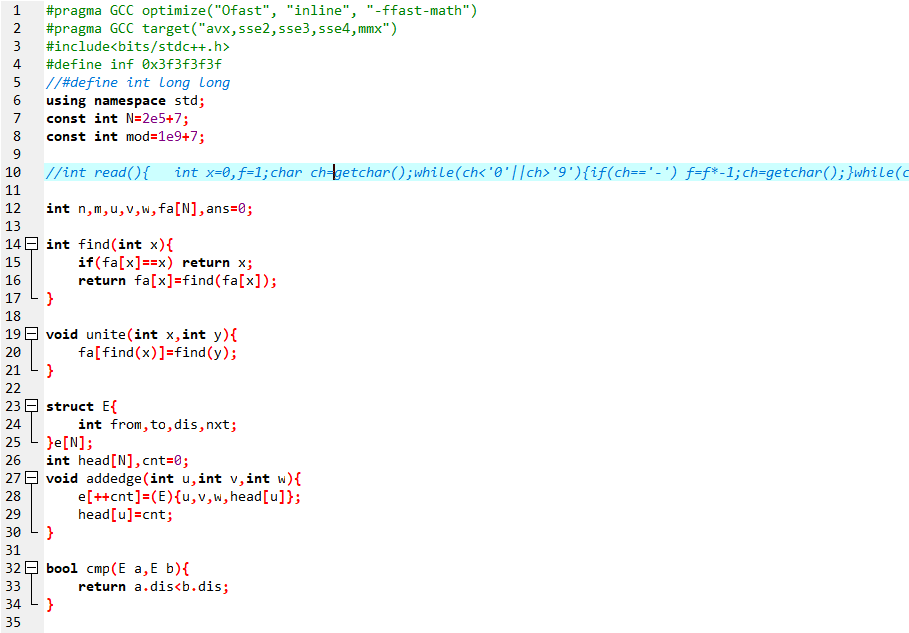
cout<<e[i].from<<"与"<<e[i].to<<"的道路相连\n";

}

cout<<"最少花费为:"<<ans<<"\n";

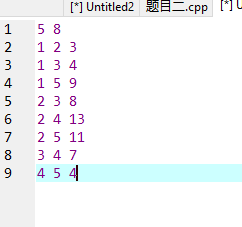
return 0;

}





**输入数据**



**输出结果**

