You are asked to write a C++ program that simulates the loading and unloading a train. The train has wagons, and they are to be loaded/unloaded with various materials available. Materials are of different types and have different priorities for loading onto a pallet in a wagon (Figure 1). Each wagon has capacity of 2000kg for each type of materials. Both wagons and materials on each wagon need to be implemented as linked-list data structure.

Figure 1: Priority of material types on a wagon.

When the train arrives at a station, some materials are wanted to be loaded or unloaded.

Loading: Starting from the first wagon (If there is no wagon at all then one has to be added to the train), materials need to be loaded to the wagons which has room for that type of material. When a wagon runs out of room for material, the remaining amount must be loaded into the next available wagon. If there is no room in any of the wagons, a new wagon must be added to the train and the remaining amount loaded into the new wagon. If the train still does not have enough space for supplies despite the addition of a new wagon, another wagon must be added to the train.

Unloading: Starting from the last wagon, the requested amount of material must be unloaded from the wagons. For example: If 3000kg of coal is to be taken from the train and there is only 1500kg of coal in the last wagon, the remaining part must be taken from the previous wagon. After each unloading process, it should be checked whether the train has any empty wagon. If it does, they should be removed from the train.

When adding a material, it should be added to the material block of its type on the pallet, if any (Figure 1). Otherwise, a new node for that kind of material should be added to the material linked list of the wagon (Figure 2). For example: If there is 1000kg of Iron in a wagon, newly arrived Iron is added to the current amount until the iron limit of that wagon allows. If there is no Iron in that wagon, a new Iron node must be added to add the that of iron.
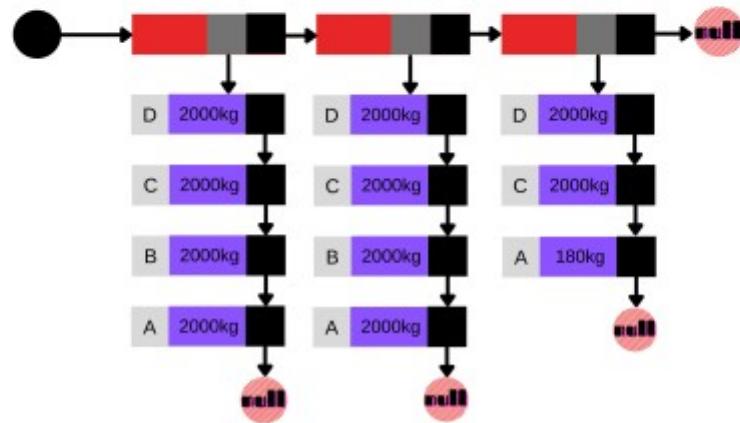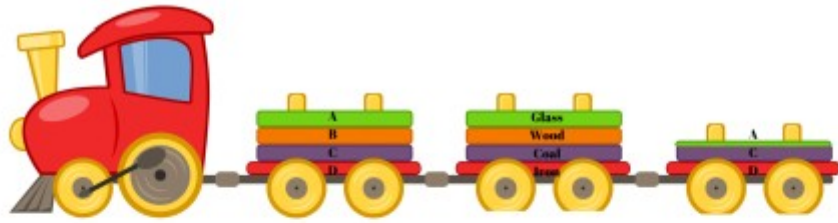
Figure 2: An example for wagon and material linked-lists schema.

Menu method():

```cpp
void menu(){

    Train newTrain;
    newTrain.create();

    int entry;
    char cargo;
    int weight;

    while(true){
        cout<<"Welcome to train station!"<<endl;
        cout<<"What would you like to do?"<<endl;
        cout<<"1 - Load Weight"<<endl<<"2 - Unload Weight"<<endl<<"3 - Print train info"<<endl<<"4 - Exit"<<endl;
        cin>>entry;

        if(entry == 1){
            cout<<"Which material you would like to load?"<<endl;
            cout<<"A - Glass"<<endl<<"B - Oak"<<endl<<"C - Coal"<<endl<<"D - Iron"<<endl;
            cin>>cargo;
            cout<<"Enter desired weight to be loaded"<<endl;
            cin>>weight;
            newTrain.addMaterial(toupper(cargo),weight);
            cout<<"Train cargo info"<<endl;
            newTrain.printWagon();
        }
        else if(entry == 2){
            cout<<"Which material you would like to unload?"<<endl;
            cout<<"A - Glass"<<endl<<"B - Oak"<<endl<<"C - Coal"<<endl<<"D - Iron"<<endl;
            cin>>cargo;
            cout<<"Enter desired weight to be unloaded"<<endl;
            cin>>weight;
            newTrain.deleteFromWagon(toupper(cargo),weight);
            cout<<"Train cargo info"<<endl;
            newTrain.printWagon();
        }
        else if(entry == 3){
            cout<<"Train cargo info"<<endl;
            newTrain.printWagon();
        }
        else if(entry == 4){
            break;
        }
        else{
            cout<<"Please enter one of the numbers stated above"<<endl;
        }
    }
};
```

As shown in the menu() method you are expected to fill 2 methods in the linkedList.cpp file, namely addMaterial() and deleteFromWagon(). You have to stay loyal to these method name, input arguments. Do not take input from user or print outputs using **cin** *and* **cout** *methods.*

As problem definition implies, you will need to implement addWagon() and removeWagon() methods. You are not restricted to these method names and you may add new methods as required according to your code design.

*void Train::**addMaterial**(char material ,int weight){ // Function for adding material to wagon*
}

*void Train::**deleteFromWagon**(char material, int weight){ //Delete function*
}

Complete and submit: Complete methods, Submit linkedList.cpp and linkedList.h files only.

Due time to submit: 10.10.2021 23:59 over ninova

Submission Rules
If explanations for this homework is not clear, you can ask your question on the message board for BLG 223E on NINOVA. Please check before writing your question whether your question is asked by someone else.

Barış Bilen – turkmenogluc@itu.edu.tr
Cumali Türkmenoğlu - bilenb20@itu.edu.tr

Make sure you write your name and number in all of the files of your project, in the following format: Make sure you write your name and number in all of the files of your project, in the following format:

/* @Author
Student Name: <student_name>
Student ID : <student_id>
Date: <date> */

- Only electronic submissions through Ninova will be accepted no later than deadline.
- You may discuss the problems at an abstract level with your classmates, but you should not share or copy code from your classmates or from the Internet. You should submit your own, individual homework.
- Academic dishonesty, including cheating, plagiarism, and direct copying, is unacceptable.
- Use comments wherever necessary in your code to explain what you did.
- Note that **YOUR CODE WILL BE CHECKED WITH THE PLAGIARISM TOOLS!**