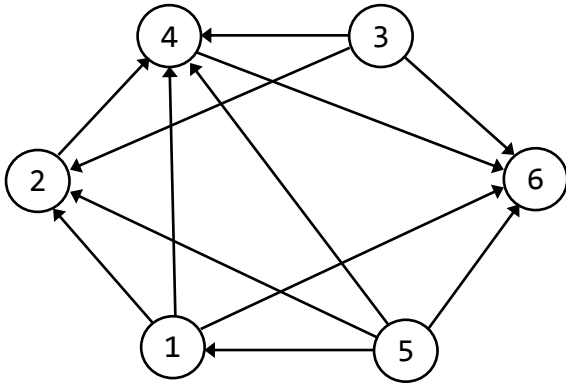


Задание №14. С помощью текстового файла опишите ребра следующего графа. Каждая строка файла должна содержать начальную вершину ребра, пробел и конечную вершину ребра. Используя содержимое файла, постройте матрицу ребер и выведите на экран начальные вершины графа (являются началом всех своих ребер) и конечные вершины графа (являются концами всех своих ребер):



```
=== Задание 14 =====
Ребра графа:
1-->2  1-->4  1-->6  2-->4
3-->2  3-->4  3-->6  4-->6
5-->1  5-->2  5-->4  5-->6
Начальные вершины графа: 3 5
Конечные вершины графа: 6
```

```
//s - динамическая строка
int nr=myinput(s, "Ребра графа:"); //файл --> s (nr-количество ребер)
R = new int[nr][2]; //матрица ребер
cout<<"Ребра графа:\n";
for(int i=0; i<nr; i++){
    R[i][0] = new int[1]; //p --> в начало строки s
    R[i][0] = atoi(s); //цикл по ребрам графа
    p = mystr(s, " "); //память для строки текущего ребра
    R[i][1] = atoi(s); //начальная вершина текущего ребра
    cout<<R[i][0]<<" --> "<<R[i][1]<<" "; //перемещаем p на конечную вершину
    if(i%4==3) cout<<endl; //конечная вершина текущего ребра
    p += mylen(s); //вывод ребра
} //endl после четырех ребер
//p -> в начало следующей строки

int nv = 6; //количество вершин графа
tmp = new int[nv]; //создаем динамический массив tmp
for(int i=0; i<nv; i++) tmp[i] = 0; //и обнуляем его элементы
for(int i=0; i<nr; i++){ //отмечаем в tmp единицами вершины,
    tmp[R[i][0]] = 1; //входящие в последний столбец R
}
cout<<"Начальные вершины графа: ";
for(int i=0; i<nv; i++){ //вершины, которые в tmp остались нулями,
    if(tmp[i]==0) cout<<i<<" "; //являются начальными
}
for(int i=0; i<nr; i++) tmp[i] = 0; //обнуляем tmp
for(int i=0; i<nr; i++){ //отмечаем в tmp единицами вершины,
    tmp[R[i][1]] = 1; //входящие в начальный столбец R
}
cout<<"\nКонечные вершины графа: ";
for(int i=0; i<nv; i++){ //вершины, которые в tmp остались нулями,
    if(tmp[i]==0) cout<<i<<" "; //являются конечными
}
delete R; //удаляем матрицу ребер
//и два динамических массива
```

Задание №15. Выполните двойную сортировку списка студентов: сначала по номеру группы, потом по номеру телефона:

```

=== Задание 15 =====
Итоговый список:
Павлющик Тимофей Николаевич      группа 1 25-216-03-96
Соловьева Людмила Владимировна   группа 1 25-387-50-74
Никитина Раиса Николаевна        группа 1 33-443-55-33
Батюшков Станислав Владимирович  группа 1 33-883-08-65
Мирончикова Зинаида Ивановна     группа 1 44-869-62-87
Дайнеко Евгений Игоревич         группа 2 25-214-01-58
. . . . .
Шумский Владимир Александрович   группа 4 44-828-38-67
Савич Сергей Александрович       группа 5 25-352-47-28
Турчинович Григорий Евгеньевич   группа 5 29-255-48-97
Куделик Леонид Викторович        группа 5 29-327-89-35
Богумило Евгений Иванович        группа 5 29-589-71-60
Суходольский Василий Сергеевич   группа 5 29-622-67-42
    
```

```

char* s = new char[10000]; //создаем и инициализируем исходный список
int ns = myinput(s, "731.txt"); //ns - длина исходного списка

char** stud = new char*[1000]; //массив строк для нового списка
int* numG = new int[1000]; //массив для номеров групп

char* p = s; //p -> в начало исходного списка
for(int i = 0; i < ns; i++){ //цикл по исходному списку
    stud[i] = new char[1000]; //память для stud[i]
    mycpy(stud[i], p); //текущую строку -> в stud[i]
    for(int j = 0; j < 10; j++){ //p -> на пятое слово
        p += mycspn(p, " "); p += 1;
    }
    numG[i] = atoi(p); //номер группы -> в numG[i]
    p += mylen(p); //p -> в начало следующей строки
}

mysrt(stud, ns, numG, (sizeof(char*))1000); //первая сортировка по группе
//чтобы выполнить вторую сортировку, надо знать
//адреса телефонов, размер групп и их начальные строки в stud

char** tel = new char*[1000]; //массив указателей на № телефонов
int* ng = new int[1000]; //массив для размеров групп
for(int i = 0; i < ns; i++){ //цикл по массиву студентов
    p = stud[i];
    for(int j = 0; j < 10; j++){ //p -> на слово №6 строки stud[i]
        p += mycspn(p, " "); p += 1;
    }
    tel[i] = p; //адрес № телефона -> в tel
    ng[i]++; //увеличиваем размер группы
}
    
```

```

int n = 0; //n - № начальной строки первой группы
for(int i = 0; i < groups; i++){ //цикл по группам
    //вторая сортировка текущей группы по телефону
    mysrt(&stud[i], groups[i], i, &stud[i]);
    n += groups[i]; //n <- № начальной строки следующей группы
}
cout<<"Итоговый список:\n";
for(int i = 0; i < groups; i++) { //цикл по массиву студентов
    p = mystr(stud[i], "phone")-1; //адрес конца ФИО студента
    int lenfio = p - stud[i]; //длина ФИО
    myout(stud[i], groups[i], groups[i]+1); //вывод ФИО
    cout<<setw(33-groups[i])<<' ' //вывод пробелов для выравнивания
        <<groups[i] //вывод № группы и телефона
}
//удаляем строки stud и динамические массивы задания
for(int i = 0; i < groups; i++) delete[] stud[i];
delete[] stud; delete[] groups; delete[] stud; delete[] groups;

```