```
#include <iostream>
#include <vector>
#include <cstdlib>
#include <cstdio>

using namespace std;
static int amount=0;
class derevo
{
    public:
    int val;
    derevo* parent;
    derevo* mybody;

    derevo(int myval)
    {
        val=myval;
        amount++;
        //cout<<"amount="<<amount<<endl;
    }
    void set_parent_adres(derevo* adress)
    {
        parent=adress;
    }
    void set_mybody_adres(derevo* body)
    {
        mybody=body;
    }
    derevo* add_child(int child_val)
    {
        derevo* myderevo=mybody;
        int rod=0;
        while(myderevo!=0)
        {
            if(myderevo->val==child_val)
            {
                rod=1;
            }
            myderevo=myderevo->parent;
        }
        //cout<<"rod="<<rod<<endl;
        if(rod==0)
        {
            derevo* new_child=new derevo(child_val);
            new_child->parent=mybody;
            new_child->mybody=new_child;
            //new_child.val=child_val;
            return new_child;
        }
        else
        {
            //derevo* new_child=new derevo(0);
            //new_child->parent=0;
            //new_child->mybody=0;
            derevo* new_child=0;
            return new_child;
        }
    }
    void print()
    {
        FILE *fp=fopen("my.txt","a");

        derevo* myderevo=mybody;
        while(myderevo!=0)
        {
            //cout<<myderevo->val<<" ";
            fprintf(fp,"%d ",myderevo->val);
            myderevo=myderevo->parent;
        }
        //cout<<endl;

        fprintf(fp,"\n");
        fclose(fp);
    }
};
void print_vec(vector <int> vec)
{
    for(int i=0;i<vec.size();i++)
    {
        cout<<vec[i]<<" ";
    }
    cout<<endl;
}
void print_ar(int ar[], int length)
{
    for(int i=0;i<length;i++)
    {
        cout<<ar[i]<<" ";
    }
    cout<<endl;
}
int poisk(int * ar, int length,int num)
{
    for(int i=0;i<length;i++)
    {
        if(ar[i]==num)
        {
            return 1;
        }
    }
    return 0;
```

```
}
int make_num(vector <int> vec)
{
    int num=0,r=1;
    for(int i=0;i<vec.size();i++)
    {
        num=num+r*vec[i];
        r*=10;
    }
    return num;
}
void olymp(int start, int finish, int sdvig, derevo* rost)
{
    //cout<<"sdvig="<<sdvig<<endl;
    if(amount>30000000)
    {
        return;
    }
    rost->print();
    //cout<<"start="<<start<<" finish="<<finish<<" sdvig="<<sdvig<<endl;
    derevo* second;
    if(finish!=start)
    {
        int zifry=finish, i, maxt, t, k, rod, even=0;
        vector <int> ar;
        while(zifry>0)
        {
            if((zifry%10)%2==0)
            {
                even++;
            }
            ar.push_back(zifry%10);
            zifry=(zifry)/10;
        }
        if(even==0)
        {
            return;
        }

        if(ar[0]%2==0)
        {
           // cout<<"ar[0]="<<ar[0]<<endl;

            //for(int r=0;r<sdvig;r++)
            //{
            //    cout<<" ";
            //}
            //cout<<finish/2<<endl;

            second = rost->add_child(finish/2);
            if(second==0)
            {
                //cout<<"stop"<<endl;
            }
            else
            {
                 olymp(start,finish/2, sdvig+1,second);
            }
        }
        int temp[ar.size()];
        for(int u=0;u<ar.size();u++)
        {
            temp[u]=u+1;
        }
        rod=1;

        while(rod==1)
        {
            rod=0;
            for(int u=ar.size()-1;u>=0;u--)
            {
                if(temp[u]<ar.size())
                {
                    maxt=-1;
                    for(int r=u+1;r<ar.size();r++)
                    {
                        if(maxt==-1 && temp[r]>temp[u])
                        {
                            maxt=r;
                        }
                        else if(temp[r]>temp[u] && temp[maxt]>temp[r])
                        {
                            maxt=r;
                        }
                    }
                    if(maxt!=-1)
                    {
                        rod=1;
                        t=temp[u];
                        temp[u]=temp[maxt];
                        temp[maxt]=t;

                        t=ar[temp[u]-1];
                        ar[temp[u]-1]=ar[temp[maxt]-1];
                        ar[temp[maxt]-1]=t;


                        k=0;
                        for(int r=u+1;r<=(ar.size()-1+u+1)/2;r++)
                        {
                            t=temp[u+1+k];
                            temp[u+1+k]=temp[ar.size()-1-k];
                            temp[ar.size()-1-k]=t;

                            t=ar[temp[u+1+k]-1];
                            ar[temp[u+1+k]-1]=ar[temp[ar.size()-1-k]-1];
                            ar[temp[ar.size()-1-k]-1]=t;

                            k++;
                        }

                        //print_ar(temp,ar.size());
                        //print_vec(ar);

                        //cout<<"lost"<<endl;
                        //cout<<"perestanovki"<<" Chain="<<chain<<endl;

                        if(ar[ar.size()-1]!=0)
                        {
                            //cout<<endl;
                            //for(int r=0;r<sdvig;r++)
                            //{
                            //    cout<<" ";
                            //}
                            //cout<<make_num(ar)<<endl;

                            second = rost->add_child(make_num(ar));
                            if(second==0)
                            {
                                //cout<<"stop"<<endl;
                            }
                            else
                            {
                                olymp(start,make_num(ar),sdvig+1,second);
                            }
                        }

                        break;
                    }
                }
            }
        }
    }
    else
    {
        rost->print();
    }
}

int main()
{
    FILE *fp=fopen("my.txt","w");
    fclose(fp);
    //olymp(1, 631, 0);
    int start=1;
    int sdvig=0;

    derevo first(218);
    first.set_parent_adres(0);
    first.set_mybody_adres(&first);

    olymp(start, 218, sdvig, &first);

    //derevo first(218);
    //first.set_parent_adres(0);
    //first.set_mybody_adres(&first);
    //derevo* second = first.add_child(128);
    //derevo* third = first.add_child(34);
    //derevo* fourth = second->add_child(218);
    //second->print();
    //fourth->print();
    //third->print();
    return 0;
}
```