

```

#include <string>
#include <iostream>
#include "Collection.h"
#include "Graphe.h"
using namespace std;

static bool e = false, g = false; // variables permettant de traiter les options
static int t = -1;                //

static string ficGraphe, ficLogs; // nom des fichiers

const char *EOPT = "-e";          // les options
const char *TOPT = "-t";          //
const char *GOPT = "-g";          //

const string EXTLOG = ".log";      // extension des fichiers
const string EXTDOT = ".dot";      //

int main(int argc, char* argv[])
{
    /*verification du nom de fichier de logs*/
    if (argc > 1) // si nom de fichiers sont spécifi  
    {
        ficLogs = string(argv[argc-1]);
        if (ficLogs.find(EXTLOG) != string::npos) // le dernier param  tre est
un fichier de log
        {
            //on est bon
        }
        else
        {
            cerr << "Nom du fichier de logs manquant ou extension
invalide" << endl;
            return 1;
        }
    } // fin du si fichiers
    else
    {
        cerr << "Nom du fichier de logs manquant" << endl;
        return 1;
    }

    /*traitement des options*/
    for (int i = 1; i < argc; i++) // parcours des arguments
    {
        if (strcmp(argv[i], EOPT) == 0) //test si c'est l'option e
        {
            e = true;
        }
        else if (i + 1 < argc && strcmp(argv[i], TOPT) == 0) //test si
option t et argument suivant
        {
            if (atoi(argv[i+1]) >= 0 && atoi(argv[i+1]) < 24)
            {
                t = atoi(argv[i+1]);
            }
        }
    }
}

```

```

        } // fin de l'option t
        else if (i + 1 < argc && strcmp(argv[i], GOPT) == 0) //test si
option g et argument suivant
        {
            ficGraphe = string(argv[i + 1]);
            if (ficGraphe.find(EXTDOT) != string::npos)
            {
                g = true;
            }
        } // fin de l'option g
    } // fin du parcours d'arguments

#ifdef MAP
    cout << "e = " << e << endl;
    cout << "t = " << t << endl;
    cout << "g = " << g << endl;
    cout << "nom graphe = " << ficGraphe << endl;
#endif

    /*execution des traitements demandés*/
    Collection collection(ficLogs);
    if (g) // si option demande de graphe
    {
        Graphe graphe(collection, e, t);
        graphe.GenereFichier(ficGraphe);
    }
    else
    {
        collection.Top10(e, t);
    }

    return 0;
}

```