```
#include <iostream>
#include <cstdlib>
#include <cerrno>
#include <unistd.h>
#include <fcntl.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <time.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sstream>
#include <pwd.h>
#include <ctype.h>

using namespace std;

/**
@param argc the number of arguments
@param argv char array of arguments
function that prints that last specified lines or bytes of a file
*/
int main(int argc, char * argv[])
{
  bool f,c,n = false;
  int p;

  while((p = getopt(argc,argv,"fcn:")) != -1)
    {
      switch(p)
        {
        case 'f':
          f = true;
          break;

      case 'c':
        c = true;
        break;

      case 'n':
        n = true;
        break;

        default:
          return EXIT_FAILURE;
      }
    }

  char * filename;

  if(argc ==1)
    {
      int n =0;
      int numLines = 0;
      char buffer[100000];
      while((n = read(0, buffer, 1024)) > 0)
        {
          buffer[n] = '\0';
          write(STDOUT_FILENO,buffer,n);
          numLines ++;
          for(int i = 0; i < 1024; i++)
            {
              buffer[i] = '\0';
            }

        }

    }
  else if(argc == 2 && !f && !c && !n) // ./tail file
    {
      char * filename = argv[1];
      int fd = open(filename,O_RDONLY);

      if (fd != -1)
      {
        char buffer[1024];
        int n = 0;

        while ((n = read(fd, buffer, 1024)) > 0)
          {
            buffer[n] = '\0';
            write(STDOUT_FILENO, buffer, n);
          }
        close(fd);
      }
      else
      {
        cout << "tail: cannot open '" << filename << "' for reading: No such
file or directory" << endl;
        return EXIT_FAILURE;

      }

    }
  else if(argc == 2 && !c && !n && f) // ./tail -f
    {
      cout << "tail: warning: following standard input indefinitely is
ineffective" <<endl;
      int n =0;
      int numLines = 0;
      char buffer[100000];
      while((n = read(0, buffer, 1024)) > 0)
        {
          buffer[n] = '\0';
          write(STDOUT_FILENO,buffer,n);
          numLines ++;
          for(int i = 0; i < 1024; i++)
            {
              buffer[i] = '\0';
            }

        }

    }
  else if(argc == 2) // ./tail -c || ./tail -n
    {
      if(n)
      {
        cout << "tail: option requires an argument -- 'n'" <<endl;
        cout << "Try `tail --help' for more information." << endl;
        return EXIT_FAILURE;
      }
      if(c)
      {
        cout << "tail: option requires an argument -- 'c'" <<endl;
          cout << "Try `tail --help' for more information." << endl;
          return EXIT_FAILURE;
      }
    }
  else if(argc == 3 && f && (c || n)) // ./tail -f -c || ./tail -f -n
    {
      if(n)
        {
          cout << "tail: option requires an argument -- 'n'" <<endl;
          cout << "Try `tail --help' for more information." << endl;
          return EXIT_FAILURE;
        }
      if(c)
      {
          cout << "tail: option requires an argument -- 'c'" <<endl;
          cout << "Try `tail --help' for more information." << endl;
          return EXIT_FAILURE;
      }
    }
  else if(argc == 3 && !f && (c||n)) //.tail -c number || ./tail -n number
    {
      int num = atoi(argv[2]);
      if(num == 0 && argv[2] != "0")
        {
          cout << "tail: " << argv[2] <<": invalid number of lines" <<endl;
          return EXIT_FAILURE;
        }

      int n =0;
      char buffer[100000];
      while((n = read(0, buffer, 1024)) > 0)
      {
        buffer[n] = '\0';
        write(STDOUT_FILENO,buffer,n);
        for(int i = 0; i < 1024; i++)
          {
            buffer[i] = '\0';
        }

      }


    }
  else if(argc == 3 && f && !c && !n) //./tail -f file
    {
      filename = argv[2];
      int totalLines = 0;
      int fd = open(filename, O_RDONLY);
      if (fd != -1)
        {
          char buffer[100000];
          int n = 0;
          char newBuffer[100000];
          int curLine = 0;
        int startPoint = 0;
          while((n = read(fd, buffer, 2048)) > 0)
            {
              for(int i = 0; i < 2048; i++)
                {
                  if(buffer[i] == '\n')
                    {
                      totalLines ++;
                    }
            }
          }

        startPoint = totalLines - 10;
        int count = 0;


        for(int i = 0; i < 2048; i++)
          {
            char temp = buffer[i];

            if(curLine >= startPoint)
            {
              newBuffer[count] = temp;
              count ++;
            }
            if(temp == '\n')
            curLine ++;

          }
              write(STDOUT_FILENO,newBuffer,1024);
      }
      else
      {
        cout << "tail: cannot open `" << filename << "' for reading: No such
file or directory" << endl;
        return EXIT_FAILURE;
      }
      int p =0;
      int numLines = 0;
      char buffer[100000];
      while((p = read(0, buffer, 1024)) > 0)
        {
          buffer[p] = '\0';
          write(STDOUT_FILENO,buffer,p);
          numLines ++;
          for(int i = 0; i < 1024; i++)
            {
              buffer[i] = '\0';
            }

        }

    }
  else if(argc == 4 && f && c && n) //./tail -f -n -c
    {

    }
  else if(argc == 4 && !f && (c || n) && !(c&&n)) //./tail -n number file
    {
      filename = argv[3];
      int num = atoi(argv[2]);
      int totalLines = 0;
      int totalBytes = 0;
      if(num == 0 && argv[2] != "0")
      {
        cout << "tail: " << argv[2] <<": invalid number of lines" <<endl;
        return EXIT_FAILURE;
      }
      int fd = open(filename, O_RDONLY);
      if (fd != -1)
      {
        char buffer[100000];
        int n = 0;
        char newBuffer[100000];
        int curLine = 0;
        int curByte = 0;
        int startPoint = 0;
        while((n = read(fd, buffer, 2048)) > 0)
          {
            for(int i = 0; i < 2048; i++)
            {
              if(buffer[i] == '\n')
                {
                  totalLines ++;
                }
              if(buffer[i] != '\0')
                {
                  totalBytes ++;
                }
            }
            if(n)
             startPoint = totalLines - num;
            if(c)
            {
              startPoint = totalBytes - num;
            }

            int count = 0;


            for(int i = 0; i < 2048; i++)
            {
              char temp = buffer[i];
              if(n)
                {
                  if(curLine >= startPoint)
                  {
                    newBuffer[count] = temp;
                    count ++;
                  }
                  if(temp == '\n')
                  curLine ++;
                }
              if(c)
                {
                  curByte ++;
                  if(curByte > startPoint)
                  {
                    newBuffer[count] = temp;
                    count ++;

                  }
                }
            }
            write(STDOUT_FILENO,newBuffer,n);
            close(fd);
          }
        //cout <<"curLine: " <<  curLine << endl;
        //cout << "startPoint: "<<startPoint << endl;
        //cout << "totalLines: " << totalLines << endl;
        //cout << "num: " << num << endl;
        //cout << "total bytes: " << totalBytes << endl;
        //cout << "start point (total bytes - num): " <<startPoint << endl;
        //cout << "current byte: " <<curByte;
      }
      else
      {
        cout << "tail: cannot open" << filename << "for reading: No such file or
directory" << endl;
        return EXIT_FAILURE;
      }
    }
  else if(argc == 5 && f && (c || n))
    {
```

```
filename = argv[4];
int num = atoi(argv[3]);
if(num == 0 && argv[3] != "0")
  {
    cout << "tail: " << argv[3] <<": invalid number of lines" <<endl;
    return EXIT_FAILURE;
  }

int totalLines = 0;
int totalBytes = 0;
int fd = open(filename, O_RDONLY);
if (fd != -1)
  {
    char buffer[100000];
    int n = 0;
    char newBuffer[100000];
    int curLine = 0;
   int curByte = 0;
   int startPoint = 0;
    while((n = read(fd, buffer, 2048)) > 0)
      {
        for(int i = 0; i < 2048; i++)
          {
            if(buffer[i] == '\n')
              {
                totalLines ++;
              }
         if(buffer[i] != '\0')
           {
             totalBytes++;
           }
      }

      if(n)
      startPoint = totalLines - num;
      if(c)
      startPoint = totalLines - num;

      int count = 0;

      for(int i = 0; i < 2048; i++)
      {
        char temp = buffer[i];
        if(n)
      {
        if(curLine >= startPoint)
          {
            newBuffer[count] = temp;
            count ++;
          }
        if(temp == '\n')
          curLine ++;
      }
        if(c)
```

```
                {
                  curByte ++;
                  if(curByte > startPoint)
                  {
                    newBuffer[count] = temp;
                    count ++;

                  }
                }
            }

            write(STDOUT_FILENO,newBuffer,n);
          }
      }
      else
      {
        cout << "tail: cannot open `" << filename << "' for reading: No such
file or directory" << endl;
        return EXIT_FAILURE;
      }
      int p =0;
      int numLines = 0;
      char buffer[100000];
      while((p = read(0, buffer, 1024)) > 0)
        {
          buffer[p] = '\0';
          write(STDOUT_FILENO,buffer,p);
          numLines ++;
          for(int i = 0; i < 1024; i++)
            {
              buffer[i] = '\0';
            }

        }

    }
  else
    {
      cout << "Error, invalid input." <<endl;
    }
}
```