Fehlerbehandlung bei new bzw. malloc/calloc (newdemo.cpp)

```
#include <iostream>
#include <new.h>
#include <process.h>
#define N 1940000000UL   // N = 1940 MB

using namespace std;

//Fehlerbehandlungsroutine:
int MyNewHandler(size_t size) //size: angeforderter Speicherplatz
{
    clog<<"Dynamic memory requested: "
        <<(unsigned long)size
        <<" Bytes, allocation failed\n";
    return 0; //Rueckgabe  0: kein erneuter Versuch Allokierung
}             //Rueckgabe !0: Wiederholung der Allokierung

void main()
{
 _set_new_handler(MyNewHandler); // Zeiger MyNewHandler
 char *bigString = 0;
```

Fehlerbehandlung bei new bzw. malloc/calloc (newdemo.cpp)

```
try {
     bigString = new char[N];    // Fehler: Aufruf MyNewHandler
     for(size_t i=0; i<N-1; i++) // auffuellen
         bigString[i]='#';       // mit '#'
     bigString[N-1] = 0; // am Ende: '\0'
     cout<<"adr bigString = "<<hex<<(int *)(bigString)<<endl;
     cout<<"    bigString = "<<bigString<<endl;
     delete[] bigString; bigString = 0;
}
catch(...){ clog<<"caught exception 1\n"; }

// Retten des Zeigers auf bisherige Fehlerbehandlungsroutine und
_PNH old_handler = _set_new_handler(0);//Fehlerbeh. deaktivieren

try {
     bigString = new char[N]; delete [] bigString; bigString = 0;
}
catch(...){ cerr<<"caught exception 2\n"; }

_set_new_handler(old_handler);  // MyNewHandler wieder aktiviert
```

Fehlerbehandlung bei new bzw. malloc/calloc (newdemo.cpp)

```
 _set_new_mode(1); // auf Fehlerbehandlung fuer malloc umschalten

 char *s = (char *)malloc(N);
 if(!s) cout<<"malloc("<<N<<") nicht erfolgreich !\n";
 free(s); s=0;

 _set_new_mode(0); // auf Fehlerbehandlung fuer new umschalten
 cin.ignore();
}

/*
Dynamic memory requested: 1940000000 Bytes, allocation failed
caught exception 1
caught exception 2
Dynamic memory requested: 1940000000 Bytes, allocation failed
malloc(1940000000) nicht erfolgreich !
*/
```