

```

// W17_3.cpp : Defines the entry point for the console application.
//          Ponowne zgłoszenie wyjątku

#include "stdafx.h"
#include "windows.h"
#include <iostream>
using namespace std;

void try_alloc(size_t m_sz);
size_t get_maxmemorybuff();

size_t get_maxmemorybuff()
/*=====
Zwraca rozmiar maksymalnego buforu, który może być
alokowany w pamięci
=====*/
{
    size_t mem_size;      //rozmiar buforu
    size_t mem_decr = 1024*1024; // 1 MB
    MEMORYSTATUS stat;    //struktura MEMORYSTATUS - MSDN

    GlobalMemoryStatus(&stat); //Funkcja Win32 (SDK) - wypełnia pola struktury
                                //MEMORYSTATUS stat
    mem_size = stat.dwTotalPhys; //rozmiar pamięci fizycznej, B

    while(mem_size > mem_decr)
    {
        try
        {
            try_alloc(mem_size);
            return mem_size; //powodzenie - przerywamy petlie while
        }
        catch (bad_alloc)
        {
            //przechwytywanie ponownie zgłoszonego wyjątku w funkcji try_alloc
            mem_size -= mem_decr;
        }
    }

    return mem_size;
}

void try_alloc(size_t m_sz)
{
    double *buff = NULL;

    try
    {
        buff = new double [m_sz/sizeof(double)]; //parametr w [...] - ilość słów danego
        typu
        delete [] buff;
        /*
        buff = (double *)HeapAlloc(GetProcessHeap(), 0, m_sz);
        if(buff)
            HeapFree(GetProcessHeap(), 0, buff);
        else
        {
            bad_alloc a;
            throw a;
        }
        */
        buff = NULL;
    }
    catch (bad_alloc)
    {
        //nie udało się zaalokować bufor o podanym rozmiarze - powstał wyjątek
        throw; //generuje ponownie wyjątek tego samego typu
               //ten wyjątek będzie przechwycony blokiem catch
               //górnego poziomu
    }
}

int _tmain(int argc, _TCHAR* argv[])

```

```
{  
    size_t max_buffsize = get_maxmemorybuff();  
    cout << "największy rozmiar bufora, który można zaalokować w RAM: " << max_buffsize << "\n";  
    system("pause");  
    return 0;  
}
```