**BÀI TẬP NHÓM**

**MÔN LẬP TRÌNH HỆ THỐNG**

Danh sách thành viên:

1. Bùi Thành Chung
2. Nguyễn Hoàng Linh
3. Bùi Thị yến
4. Nguyễn Chiến Thắng
5. Tống Đức Anh
6. Giáp Hồng Sơn
7. Đinh Hoài Nam
8. Bùi Biên Cương
9. Hoàng Thị Lan Anh
10. Nguyễn Hữu Thọ
11. Nguyễn Văn Hạnh
12. Trương Hà Thanh

***Đề bài***

Viết chương trình liên kết giữa ngôn ngữ C/C++ và Assembly với sự phân công công việc như sau:

* Ngôn ngữ C/C++:

- Nhập từ bàn phím mảng n số nguyên.

- Gọi hàm loại bỏ các phần tử chẵn ra khỏi mảng bên assembly viết

- hiện mảng kết quả ra màn hình.

* Ngôn ngữ Assembly

- Viết hàm loại bỏ các phần tử chẵn ra khỏi mảng do C/C++ truyền qua

1. **Tổng quan:**

- Công cụ sử dụng: Visual Studio 2015, MASM

- Link github:

1. **Project: gồm 2 file source code và 1 file exe để chạy demo**

*1. File EHOUAssignment.cpp****:*** gồm các hàm

***Hiển thị mảng***

void displayArray(const int\* x, const int n) {

cout << endl;

for (int i = 0; i < n; i++)

{

cout << x[i] << " ";

}

cout << endl;

}

***Nhập từ bàn phím mảng số nguyên:***

void inputArray(int\* &x, int& n) {

cout << "Enter size of the array: ";

cin >> n;

if (x != NULL) {

delete[] x;

}

x = new int[n];

for (int i = 0; i < n; i++)

{

cout << "Enter item " << "[" << i << "]: ";

cin >> x[i];

}

}

Và 1 hàm main là entry point của ứng dụng

int main() {

int\* x = NULL;

int n;

inputArray(x, n);

cout << "\nOriginal Array: \n";

displayArray(x, n);

int\* y = new int[n]; // the result will be saved to this array

int result = ExcludeEvenNumber\_(y, x, n);

cout << "\nAltered Array (After exclude even elements): \n";

displayArray(y, result);

system("pause");

delete[] x;

delete[] y;

return 0;

}

***2. File ExcludeEvenNumber.asm***:chứa mã nguồn assembly hàm thực hiện loại bỏ phần tử chẵn ra khỏi mảng

.model flat, c

.code

; extern "C" int ExcludeEvenNumber\_(int\* y, const int\* x, int n);

ExcludeEvenNumber\_ proc

; function prolog

push ebp

mov ebp, esp

sub esp, 4 ; allocate temp variable

push esi

push edi

; calculate sum, save to eax

xor eax, eax; sum = 0

; load parameters

mov esi, [ebp + 8]; esi = y

mov edx, [ebp + 12]; edx = x

mov ecx, [ebp + 16]; (counter)i = n

xor ebx, ebx; array offset(from pointer x)

xor eax, eax; return value from function: return = 0

calculate:

mov edi, [edx]; edi = x[i]

mov[esp - 4], edi; temp = edi

and edi, 1h ; check whether x[i] even or odd

jz next

mov edi, [esp - 4]; restore value from temp to edi

mov[esi + ebx], edi; y[j] = x[i]

add ebx, 4; j = j + 1

inc eax; increase number item in array y

next:

add edx, 4; i = i + 1

loop calculate

; function epilog

pop edi

pop esi

mov esp, ebp

pop ebp

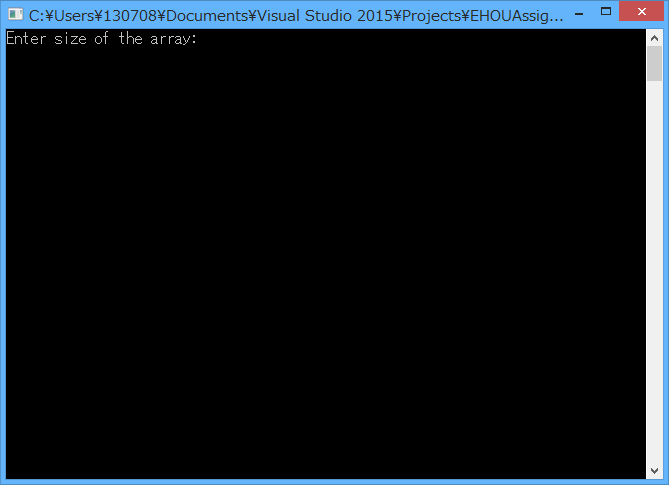
ret

ExcludeEvenNumber\_ endp

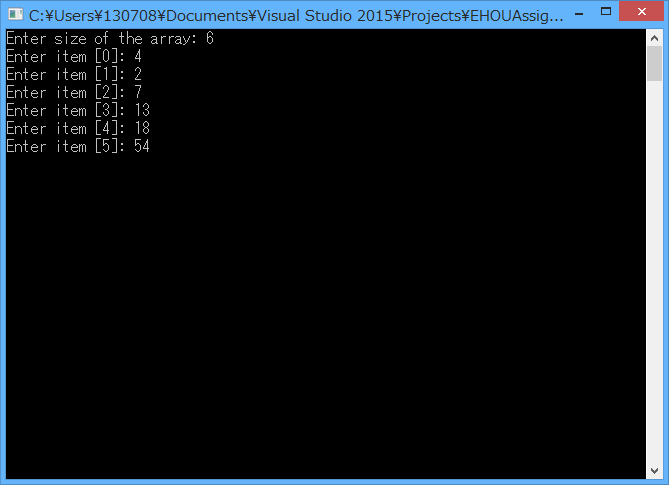
end

1. Demo

1. Khi bắt đầu chạy, chương trình yêu cầu nhập kích thước của mảng



2. Nhập kích thước mảng và nhập từng phần tử



3. Hiển thị kết quả

