

**Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Факультет безопасности информационных технологий

Дисциплина:
«Информатика»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

«Работа с ассемблером»

Выполнил:

Суханкулиев Мухаммет,
студент группы N3146

(подпись)

Проверил:

Безруков Вячеслав Алексеевич,
старший преподаватель по информатике

(отметка о выполнении)

(подпись)

Санкт-Петербург
2023 г.

1 СОЗДАТЬ АССЕМБЛЕРНЫЙ МОДУЛЬ ДЛЯ ПЕРЕВОДА ЧИСЛА В СТРОКУ

- Создаем пустой проект в Microsoft Visual Studio;
- В окне Обозреватель решений правой кнопкой мыши по проекту => Зависимости сборки => Настройки сборки... => Выбираем .masm(.targets, .props) => Жмём ОК;
- В этом же окне выбираем Исходные файлы, жмём Ctrl+Shift+A => Создаем файл с расширением .asm. Убеждаемся, что в Свойствах => Общие Тип элемента выбран Microsoft Macro Assembler;
- Так же создаем файл с расширением .c. В Свойствах Тип элемента – Компилятор C/C++;
- Для Запуска без отладки жмём Ctrl+F5.

1.1 Код .C

```
# include <stdio.h>

extern "C" char* _stdcall fun1(int);

int main() {
    int a = 1024;
    // Вывод результата функции `fun1`
    printf("%s \n", fun1(a));

    return 0;
}
```

1.2 Код .ASM

```
.586
.MODEL FLAT, STDCALL
PUBLIC fun1
_DATA SEGMENT
    strMas db 11 dup
    dec1 dd 10
_DATA ENDS
_TEXT SEGMENT
    ; Функция принимает на вход целое число и возвращает его строковое
    ; представление в десятичной системе счисления:
    fun1 PROC par1:DWORD
        ; Инициализация переменных
        lea EBX, StrMas
        push EBX
        mov ECX, 11
        ; Заполнение буфера пробелами
    metka1:
        mov BYTE PTR [EBX], ' '
        inc EBX
        Loop metka1
        ; Сохранение числа в EAX
        mov EAX, par1 ; -2147483648(число в положительной области не существует)
        ...+2147483647
        push EAX
        ; Проверка знака числа
        OR EAX,EAX
        jns metka2
        ; Если число отрицательное, делаем его положительным
        neg EAX
        ; Цикл, выполняющий деление числа на 10 и вывод цифр на экран
    metka2:
        ; Обнуляем старшую часть числа
        xor EDX,EDX
        div dec1 ; делитель 32 бита в регистре или ячейке памяти
        ; частное 32 бита в регистре EAX, остаток 32 бита в регистре EDX
        ; Добавляем к остатку код символа цифры
        add DL, '0'
        ; Сохраняем цифру в буфере
        dec EBX
        mov BYTE PTR [EBX],DL
        inc ECX
        ; Проверяем, не равно ли частное нулю
        or EAX,EAX
        jnz metka2
        pop EAX ; восстанавливаем EAX из стека и проверяем знак
        or EAX,EAX
        jns metka3
        ; Если число отрицательное, добавляем минус
        dec EBX
        mov BYTE PTR [EBX], '-'
        inc ECX
        ; Возврат адреса буфера в регистр EAX
    metka3:
        pop EBX
        mov EAX,EBX
        ; Возврат из функции
        ret
    fun1 ENDP
```

_TEXT ENDS
END