

соглашения Stdcall

1. можно изменять `eax`, `edx`, `ecx` - остальные менять нельзя
2. стек чистит процедура
3. параметры принимает только через стек первый параметр ближе к началу памяти поэтому `push` его последний(справа налево)
4. результат возвращается `al`, `ax`, `eax`, `edx:eax`, `st0`
5. по наименованиям:
 - (a) внешние переменные (данные) начинаются с `_`
 - (b) метки с `_`
 - (c) процедуры начинаются с `_` заканчиваются `@N` - `N` число байтов параметров кратно 4

соглашения cdecl

1. можно изменять `eax`, `edx`, `ecx` - остальные менять нельзя
2. стек чистит тот кто вызвал
3. параметры принимает только через стек первый параметр ближе к началу памяти поэтому `push` его последний
4. результат возвращается `al`, `ax`, `eax`, `edx:eax`, `st0`
5. по наименованиям:
 - (a) Символ подчеркивания (`_`) префиксируется в имена, за исключением случаев, когда экспортируются `__cdecl` функции, использующие компоновку C. мы на ассемблере и паскале считайте всегда просто нижнее подчёркивание должно быть
 - (b) `.model, c` ; заставит ассемблер использовать `cdecl` для имён

соглашения fastcall

1. можно изменять `eax`, `edx`, `ecx` - остальные менять нельзя
2. стек чистит процедура
3. слева направо первые параметры что возможно засунет в `ecx`, `edx` другие параметры принимает через стек первый параметр ближе к началу памяти поэтому `push` его последний(справа налево)
4. результат возвращается `al`, `ax`, `eax`, `edx:eax`, `st0`
5. по наименованиям:
 - (a) все имена начинаются с `@`
 - (b) после имени для процедур указание как в `stdcall` `@N` - число байтов с учётом регистров

соглашения pascal

1. можно изменять `eax`, `edx`, `ecx` - остальные менять нельзя
2. стек чистит процедура
3. параметры принимает только через стек первый параметр ближе к концу сегмента поэтому `push` его первый(слева направо)
4. результат возвращается `al`, `ax`, `eax`, `edx:eax`, `st0`
5. по наименованиям:
 - (a) всё капсом

соглашения pascal (register) практическая инфа

1. читай pascal
2. стек чистит вызывающий
3. параметры принимает через регистры слева направо для регистров eax, edx, ecx стек тоже слева направо первый аргумент первым и pushится
4. результат возвращается al, ax, eax, edx:eax, st0
5. по наименованиям Pascal переименует при extern только cdecl лучше каждый раз самим писать extern name <имя>

практическая инфа fpc-masm

1. если будет задача на использовать функцию из fpc в masm то по умолчанию (если не указать в pascal у процедуры или функции - register/stdcall/cdecl/pascal) будет register соглашение
2. pascal сам переименует на венгерский манер все названия (я спросил это запоминать не надо если что укажут как назвать) там длинные имена
3. Если задача стоит наоборот в fpc использовать masm то следует указать название так как fpc не переименует (только cdecl переименует я проверял)
4. Masm же можно заставить работать на соглашениях stdcall cdecl (c) и pascal - он всё переименует при компиляции согласно стандарту и раскроет макросы: invoke согласно соглашению глобально

перепишите функцию с stdcall на cdecl, fastcall, pascal и подпишите как следует назвать функцию согласно соглашению если в .model не будет прописано соглашение

```
.data
    myr4n1 real4 8.5
; функция что будет рассмотрена
sum proc
    push ebp; создание стекового кадра
    mov ebp, esp
    sub esp, 4; место под результат сложения

    cvtsi2ss xmm0, dword ptr [ebp+12]
    addss xmm0, dword ptr [ebp+8]

    movss [esp], xmm0
    fld dword ptr [esp]

    mov esp, ebp
    pop ebp
    ret 8
sum endp

Start:
    push 7
    push myr4n1; 8.5

    call sum

    fstp z4; просто место в памяти
    outf z4; распечатка 15.5

    exit
end Start
```