Oblicz (i zostaw w ST(0)) moduł zmiennej zespolonej

```
FLD z.re        ; załaduj część rzeczywistą na stos
FMUL st, st     ; pomnóż stos przez stos, czyli do
                  kwadratu, wynik w st
FLD z.im        ; załaduj część urojoną
FMUL st, st     ; podnieś ją do kwadratu
FADDP st(1), st ; dodaj wierzchołek stosu do 2 elementu
                  stosu, zdejmij wierzchołek stosu
FSQRT           ; wyciągnij pierwiastek z wierzchołka
                  stosu, wynik w wierzchołku czyli st(0)
```

Napisz przy użyciu instrukcji łańcuchowych program przysyłający 777 bajtów z tab1 do tab2

```
MOV ecx, 777    ; ustaw licznik (ecx) na 777
LEA edi, tab1   ; wczytaj adres tab1 do rejestru celu
LEA esi, tab2   ; wczytaj adres tab2 do rejestru źródła
REPNZ movsb     ; dopóki ecx większe od zera wykonuj
                  operację movsb (czyli przenieś jeden bit)
                  ze zmiennej wskazywanej  przez esi do
                  zmiennej wskazywanej przez edi
```

Napisz program zaliczający ilość powtórzeń wartości 21 w tablicy która zawierała 777 pozycji

```
LEA ebp, tab            ; załaduj adres zmiennej tab do
                          rejestru bazowego
MOV ecx, 777            ; ustaw licznij (ecx) na 777
MOV eax, 0              ; zeruje eax, to samo co
@petla:
CMP [ebp+ecx-1], 21     ; sprawdza czy element tablicy
                          tab o indexie ebp+ecx-1 jest
                          rowny 21
JNZ @nie_zliczaj        ; jeżeli nie, skaczemy do etykiety
                          o wymownej nazwie aby pominąć
                          instrukcję niżej
INC eax                 ; w eax zliczamy wystąpienia
                          liczby 21 zwiększamy więc eax o1
@nie_zliczaj:
DEC ecx                 ; zmniejsza licznik ecx
JNZ @petla              ; sprawdza czy ostatnia operacja
                          ustawiła flagę ZF (zera), jeżeli nie
                          to skacze na początek pętli i
                          powtarza czynności ( to samo co
                          LOOP @petla )
```

Napisz program liczący ln x. Wskaźnik do x znajduje się w eax, wyniki pozostaw w rejestrze ST(0)

```
FLD1
FLD single [EAX]
FYL2X
FLDL2E
FDIV
```

Napisz program umieszczający w eax zaokrągloną średnią z eax i edx

```
PUSH EAX
PUSH EDX

FILD dword [ESP]
FIADD dword [ESP+4]
FLD1
FLD1
FADD
FDIV

FRNDINT
FISTP dword [ESP]

POP EAX
ADD ESP, 4
```

Łańcuchowo skopiować po 16 bitów z tab1 do tab2

```
mov ecx, 100
mov esi, tab1
mov edi, tab2
cld
rep movsw
```

Łańcuchowo uzupełnić tablicę wartościami 0 dla 1000 elementów.

```
MOV ECX, 1000   ; ilość elementów
LEA EDI, [tab1] ; wskaźnik do początku tablicy
XOR EAX, EAX    ; mamy wyzerować 1000
                  elementów; zależnie od rodzaju
                  operacji może starczyć zerować
                  AX, lub AL
CLD             ; chcemy iterować "do przodu"
REP STOSD       ; weeeeeeee, słowa 32bitowe
```

Policzyć wyrażenie: d=b*b-4*a*c

```
FLD [b]         ; b
FMUL st, st     ; b^2
FLD1            ; 1 ; b^2
FADD st, st     ; 2 ; b^2
FADD st, st     ; 4 ; b^2
FLD [a]         ; a ; 4 ; b^2
FMUL            ; 4*a ; b^2
FLD [c]         ; c ; 4*a ; b^2
FMUL            ; 4*a*c ; b^2
FSUB            ; b^2 - 4*a*c
FSTP [d]        ; obiad podano do stołu
```

Minimum dwóch liczb rzeczywistych. x<=min(X,Y)

```
FLD [x]
FLD [y]

FCOMI st, st(1)
FCMOVNB st, st(1)

FSTP  [x]
FFREE st
FDECSTP
```

Policzyć przyprostokątną, generalnie: $\sqrt{(c^2-b^2)}$

```
FLD [c]
FMUL st, st
FLD [b]
FMUL st, st     ; b^2 ; c^2
FSUB            ; c^2 - b^2
FSQRT
FSTP [z]        ; albo i nie, zależnie co mieliśmy
                  zrobić z wynikiem
```