

Dane są liczby a, b i c. Oblicz y bez używania skoków warunkowych.

y = min(a, b, c)

```
mov eax,a
mov ebx,b
mov ecx,c

// eax = min(eax, ebx)
cmp eax,ebx
cmovg eax,ebx

// eax = min(eax, ecx)
cmp eax,ecx
cmovg eax,ecx

mov y,eax
```

Napisz przy użyciu instrukcji łańcuchowych program przesyłający 555 bajtów z tablicy tab 1 do tablicy tab 2

```
mov eax, 555
mov esi, tab1
mov edi, tab2
cld
rep movsb
```

Łańcuchowo skopiować po 16 bitów z tab1 do tab2

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

Zerowanie macierzy

```
XOR eax, eax;
MOV ecx, N1;
MOV esi, A;
```

```
P1: MOV edi, [esi + 4 * ecx - 4];
PUSH ecx;
```

```
MOV ecx, N1;
```

```
P2: MOV [edi + 4 * ecx - 4], eax;
DEC ecx;
JNZ p2;
POP ecx;
DEC ecx;
JNZ p1;
```

Max lub min a i b

```
mov eax, a;
cmp eax,b;
jnc Max;
mov eax,b;
Max:mov wynik, eax;
```

Suma kwadratów elementów tablicy

```
mov esi,tab;
mov ecx,n;
xor ebx, ebx;
```

Petla:

```
mov eax,[esi+ecx*4-4];
mul eax;
mov ebx,eax;
```

add suma,ebx;

loop Petla;

Sumowanie liczb o minimalnym i maksymalnym przedziale:

```
mov eax,0;
mov ecx,n;
mov ebx,min;
mov esi,tab;
mov edx, max;
```

Petla:

```
cmp [esi+ecx*4-4],ebx;
jnc skok1;
cmp edx,[esi+ecx*4-4];
jc skok1;
add eax,[esi+ecx*4-4];
```

skok1:

```
dec ecx;
jnz Petla;
mov suma,eax;
```

$a*b + a^2 + b^2$  gdzie a = 17, b = 35

```
mov esi, tab;
mov ecx,10;//na kartkowce bylo 1000
```

```
mov eax,a;
mov ebx,b;
xor edx,edx;
```

Petla:

```
cmp [esi+4*ecx-4],17;
jne e1;
inc eax;
jmp L1;
```

e1:

```
cmp [esi+4*ecx-4],35;
jne L1;
```

//bo to jest od razu skok...!

```
inc ebx;
jmp L1;
```

L1:

```
loop Petla;
```

//loop wyskakuje z petli automatycznie!!!

dlatego jest bezpieczniej go uzywac!

```
mov edx, eax;//edx=a
imul edx,ebx;//a*b
imul eax, eax;//a*a
imul ebx, ebx;//b*b
add edx,eax;// edx= a*b + a*a
add edx,ebx;// edx= a*b + a*a +
```

b\*b

```
mov wynik, edx;
```

Silnia (loop for)

```
mov ecx,N;  
mov eax,ecx;  
dec ecx;
```

petla:

```
mul ecx;  
loop petla;  
mov silnia,eax;
```

Silnia ( JNZ )

```
mov ecx,N;  
mov eax,ecx;//do eax przenieś N  
dec ecx;
```

petla:

```
mul ecx;  
dec ecx;  
jnz petla;  
mov silnia,eax;
```

Silnia (for)

```
mov eax,1;  
mov ecx,N;  
mov ebx, 1;
```

```
petla: cmp ebx, ecx;  
jg wyjscie;  
mul ebx;  
inc ebx;  
jmp petla;
```

wyjscie: mov silnia,eax;

Wyzerowanie przekatnej macierzy:

```
mov esi,tab;  
mov ecx,n;
```

```
skok2: mov edi,[esi+4*ecx-4];  
mov [edi+4*ecx-4],0;  
loop skok2;
```

Wyzerowanie POD przekątną:

```
mov esi,tab;  
mov ecx,n;
```

```
skok2: mov edi,[esi+4*ecx-4];  
push ecx;
```

```
skok1: mov [edi+4*ecx-4],0;  
dec ecx;  
jnz skok1;  
pop ecx;  
loop skok2;
```

Program obliczający  $x = \{$  (typu całkowitego a, b, x

1.  $a + 7$  dla  $i=1$ ;
2.  $a - b + 1$  dla  $i = 2$ ;
3.  $-a(b-3)$  dla  $i = 3$ ;
4.  $-b^3 + a/4$  dla pozostałych i

```
mov eax,a;  
mov ebx,b;  
mov ecx,i;  
cmp ecx,1;  
je skok1;  
cmp ecx,2;  
je skok2;  
cmp ecx,3;  
je skok3;  
jmp pozostale;
```

skok1:

```
add eax,7;  
jmp koniec;
```

skok2:

```
sub eax,ebx;  
add eax,1;  
jmp koniec;
```

skok3:

```
neg eax;  
sub ebx,3;  
imul eax,ebx;  
jmp koniec;
```

pozostale:

```
neg ebx;  
mov edi,ebx;  
imul ebx,ebx;  
imul ebx,ebx;  
div eax,4;  
imul eax,ebx;
```

koniec:

```
mov x,eax;
```

Dane są dwie liczby całkowite a i b

$y = \begin{cases} -ab & \text{dla } 2 < a < 5 \\ b - a & \text{dla pozostałych a} \end{cases}$

```
mov eax,a;  
mov ebx,b;  
cmp eax,2;  
jng pozostale;  
cmp eax,5;  
jnl pozostale;  
neg eax;  
imul eax,ebx;  
mov edx,eax;  
jmp koniec;
```

pozostale:

```
sub ebx,eax;  
mov edx,ebx;
```

koniec:

```
mov y,edx;
```

