


```

; Funkcja wykonuje algorytm dyfuzji, poprzez badanie otoczenia każdego piksela, w oparciu o
; przekazany element strukturalny. Przypisuje wartość 0 (czarny piksel) do badanego piksela, jeżeli
; w obrębie przyłożonego do niego elementu strukturalnego w punkcie centralnym występuje przynajmniej
; jeden piksel o wartości 0 (czarny piksel)
;
; parametry wejściowe:
; rejestr rcx - src - wskaźnik na tablicę pikseli obrazu źródłowego
; rejestr rdx - dst - wskaźnik na tablicę pikseli obrazu docelowego
; rejestr r8 - imageWidth - szerokość obrazu - zakres parametru: liczba całkowita nieujemna
; rejestr r9 - imageHeight - wysokość obrazu - zakres parametru: liczba całkowita nieujemna
; rejestr stos - elemWidth - szerokość elementu strukturalnego - zakres parametru: liczba całkowita nieujemna
; rejestr stos - elemHeight - wysokość elementu strukturalnego - zakres parametru: liczba całkowita nieujemna
; rejestr stos - centrPntX - współrzędna x punktu centralnego - zakres parametru: liczba całkowita nieujemna mniejsza niż elemWidth
; rejestr stos - centrPntY - współrzędna y punktu centralnego - zakres parametru: liczba całkowita nieujemna mniejsza niż elemHeight
;
; procedura nie niszczy rejestrów
;
; parametry wyjściowe:
; void
;
; =====

```

dilatationAsm PROC

```

; kopiowanie argumentów do rejestrów, te rejestry są używane w całej procedurze, gdy potrzebne jest użycie danego argumentu
movd xmm0, rcx ;skopiuj 1. argument - src - do rejestru xmm0
movd xmm1, rdx ;skopiuj 2. argument - dst - do rejestru xmm1
mov rax, r8 ;skopiuj 3. argument - imageWidth - do rejestru rax
movd xmm2, eax ;skopiuj 3. argument - imageWidth - do rejestru xmm2
mov rax, r9 ;skopiuj 4. argument - imageHeight - do rejestru rax
movd xmm3, eax ;skopiuj 4. argument - imageHeight - do rejestru xmm3
mov eax, dword ptr[rsp+40] ;skopiuj 5. argument - elemWidth - ze stosu do rejestru rax
movd xmm4, eax ;skopiuj 5. argument - elemWidth - do rejestru xmm4
mov eax, dword ptr[rsp+48] ;skopiuj 6. argument - elemHeight - ze stosu do rejestru rax
movd xmm5, eax ;skopiuj 6. argument - elemHeight - do rejestru xmm5
mov eax, dword ptr[rsp+56] ;skopiuj 7. argument - centrPntX - ze stosu do rejestru rax
movd xmm6, eax ;skopiuj 7. argument - centrPntX - do rejestru xmm6
mov eax, dword ptr[rsp+64] ;skopiuj 8. argument - centrPntY - ze stosu do rejestru rax
movd xmm7, eax ;skopiuj 8. argument - centrPntY - do rejestru xmm7

; kopiowanie obrazu źródłowego do docelowego obszaru pamięci (pod wskaźnikiem dst)
; petla 1 - po wierszach obrazu
;for(int j = 0; j < imageHeight; j++)
mov ecx, 0 ;int j = 0
jmp checkIfEndOfPetla1 ;skok do sprawdzania warunków końca petli
petla1:
; petla 2 - po kolumnach obrazu
;for(int i = 0; i < imageWidth; i++)
mov ebx, 0 ;int i = 0
jmp checkIfEndOfPetla2 ;skok do sprawdzania warunków końca petli
petla2:
movd xmm8, ecx ;kopiuj j do rejestru xmm8
pmuludq xmm8, xmm2 ;j*imageWidth
movd xmm9, ebx ;kopiuj i do xmm9
paddq xmm9, xmm8 ;i + j*imageWidth
movd rsi, xmm9 ;rsi = i + j*imageWidth
movd rdi, xmm0 ;adres src do rdi

```



```
503     add ebx, 1                ;h++
504 checkIfEndOfPetla3:         ;sprawdzanie warunku końca pętli 3
505     movd eax, xmm3           ;kopiuj imageHeight do eax
506     cmp ebx, eax             ;porównaj h z imageHeight
507     jl petla3                ;skocz do początku pętli, jeżeli h < imageHeight;
508
509
510     ret
511 dilatationAsm ENDP
512
513 end
```