

1. W którym procesorze Intel można obliczyć adres instrukcji w postaci $CS*16+IP$

- a) 8086
- b) 80286
- c) 80386
- d) Pentium Pro
- e) wszystkie z powyższych**

2. Do odwołania się do parametrów aktualnych stosuje się rejestr:

- a) CS b) ES c) FS **d) SS** e) DS

3. Pośredni bazowy tryb adresowania występuje w instrukcji:

- a) inc ecx
- b) mov edi, offset cos
- c) mov zmienna, edx
- d) or edx, [ebx+tab] ?**
- e) mov [edi*8+tablica], edx;

4. Która z instrukcji jest poprawna:

- a) popf eax;
- b) fsubp st, st(1);
- c) fadd st, eax;
- d) movsx ebx, al;**
- e) fdivi st, zmienna;

5. Która z instrukcji zmienia flagę Z:

- a) ?
- b) popf**
- c) cwd //nie zmienia flag
- d) fadd //nie zmienia flag?
- e) jnc //niezmienia?

6. Po wykonaniu instrukcji:

```
mov ecx, 01020304h
bswap ecx;
and ecx 0FBF4FcFBh;
mov eax, 07654321h;
and eax, ecx;
rejestr będzie zawierał:
```

- a) 05765432h
- b) 0000000...h ?
- c) 000000001h**
- d) 0c3b2a190h
- e) 08765430h

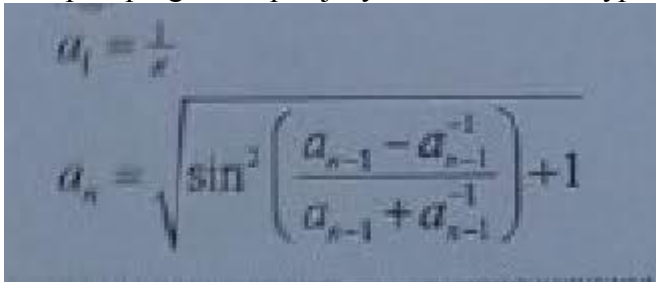
7. Po wykonaniu której instrukcji zawartość AL zmienia się z 0c9h na 0e4h? (nie jestem pewny tych wartości)

- a) or al, 0c4h;
- b) neg al
- c) rcl al, 1;

d) sar al,1;

e) mov 0c4h,al;

8. Napisz program zapisujący w n elementach typu double wektora A kolejne wyrazy ciągu:


$$a_1 = \frac{1}{x}$$
$$a_n = \sqrt{\sin^2 \left(\frac{a_{n-1} - a_{n-1}^{-1}}{a_{n-1} + a_{n-1}^{-1}} \right) + 1}$$

Rozwiązanie by spinacz

Kod:

```
mov edi, tab
mov ecx, n
dec ecx
xor eax, eax;
//obliczam pierwszy wyraz ciągu i wpisuje do tablicy

fldl    // 1
fldPI   // pi, 1
fdiv    // 1/pi
fst qword ptr[edi+8*eax];
inc eax ;

// obliczam kolejne wyrazy ciągu
petla:
fld st    // An-1 ;; An-1
fldl      // 1 ;; An-1 ;; An-1
fdivr     // 1/An-1 ;; An-1
fld st(1) // An-1 ;; 1/An-1 ;; An-1
fld st(1) // 1/An-1 ;; An-1 ;; 1/An-1 ;; An-1
faddp st(3), st // An-1 ;; 1/An-1 ;; An-1 + 1/An-1
fsubr     // An-1 - 1/An-1 ;; An-1 + 1/An-1
fdivr     // ( An-1 - 1/An-1 ) / ( An-1 + 1/An-1 )
fsin      // sin( ( An-1 - 1/An-1 ) / ( An-1 + 1/An-1 ) )
fmul st, st //sin^2
fldl      // 1, sin^2
fadd      // sin^2 +1
fsqrt     // wynik

fst qword ptr[ edi + 8*eax ] // zapisujemy obliczony wyraz ciągu do
// pamięci, i zostawiamy na szczycie stosu
inc eax;

loop petla
```

9.Podaj instrukcje mmx pakująca z nasyceniem słowa bez znaku do bajtów:

Kod:

PACKUSWB

10.Napisz przy użyciu instrukcji łańcuchowych program przesyłający 345 bajtów z tablicy tab1 do tablicy tab2.

Kod:

```
mov ecx, 345;
mov esi, tab1;
mov edi, tab2;
cld;
rep movsb;
```

11.Napisz program sprawdzający ile z 345 słów z tablicy tab1 jest różnych od 345. Wynik umieść w eax.

Rozwiązanie Marcina

Kod:

```
mov esi, tab1 ;
mov ecx, 345 ;
xor eax, eax ;
mov ebx, ecx ;
petla:
    cmp ebx, [esi+4*ecx-4] ;
    jne skok ;
    inc esi ;
skok:
    loop petla ;
    mov w, eax ;
```

moje rozwiązanie

Kod:

```
xor eax, eax;
mov ecx, 345;
mov ebx, 345;
mov ax, 345;
mov edi, tab1;
Label:
cld;
repne scasb;
```

```
inc eax;  
jnz Label;  
sub ebx, eax;  
mov eax, ebx;
```

12. Proszę napisać fragment programu w assemblerze realizujący to samo zadanie do fragment kodu w C++:

Kod:

```
int silnie_for(int N)  
{  
    int silnia=1;  
    for (int i=1; i<=N; i++)  
    {  
        silnia = silnia*i;  
    }  
    return silnia;  
}
```

Rozwiązanie Marcina

Kod:

```
xor eax, eax ;  
    mov ecx, N ;  
    inc eax ;  
petla:  
    mul ecx ;  
    dec ecx ;  
    jnz petla ;  
    mov wynik, eax ;
```

CD...

1. W którym procesorze Intel można obliczyć adres instrukcji w postaci $CS \cdot 16 + IP$
 - a) 8086
 - b) 80286
 - c) 80386
 - d) Pentium Pro
 - e) wszystkie z powyższych //według mnie odpowiedź poprawna

2.d

3.d

4.Która z instrukcji jest poprawna: //trzeba było skreślić nie

- a) popf eax; //nie jest poprawna
- b) fsubp st,st(1); //nie jest poprawna
- c) fadd st, eax; //nie jest poprawna
- d) movsx ebx,al; //jedynie ta jest poprawna
- e) fdivi st,zmienna; //nie jest poprawna

5.b

6.Po wykonaniu instrukcji:

```
mov ecx, 01020304h
```

```
bswap ecx;
```

```
and ecx 0FEF4FcFBh;
```

```
mov eax, 07654321h;
```

and eax, ecx //z tego co pamiętam to trzeba było zmienić na taką instrukcję
rejestr będzie zawierał:

- a) 05765432h
- b) 00000000...h
- c) 000000001h // według mnie ta jest poprawna
- d) 0c3b2a190h
- e) 08765430h

7.d

Przyporządkowanie rejestrów

esp, ebp: ss

eax, ebx, ecx, edx, edi, esi: ds.

eip: cs

Pamiętacie na zajęciach? było ds:[coś] odwoływaliśmy się tym do danych do argumentów na których pracowaliśmy w danym momencie nie przypominam sobie żeby pobierać cokolwiek ze stosu poza momentami w których konkretnie coś na niego wysłaliśmy wtedy się używało [esp] bądź [esp +offset] żeby odwołać się do miejsca na stosie innego ... ja tak myśle nie potrafię inaczej wyjaśnić z racji że hmm... nigdy nikt tego nie tłumaczył nam a poszukiwałem informacji na ten temat... jeśli się mylę poprawcie jak dla mnie łatwiejsze są laborki niż teoria z tego więc mogę się mylić

Z odpowiedzią nr 1 się zgadzam ...